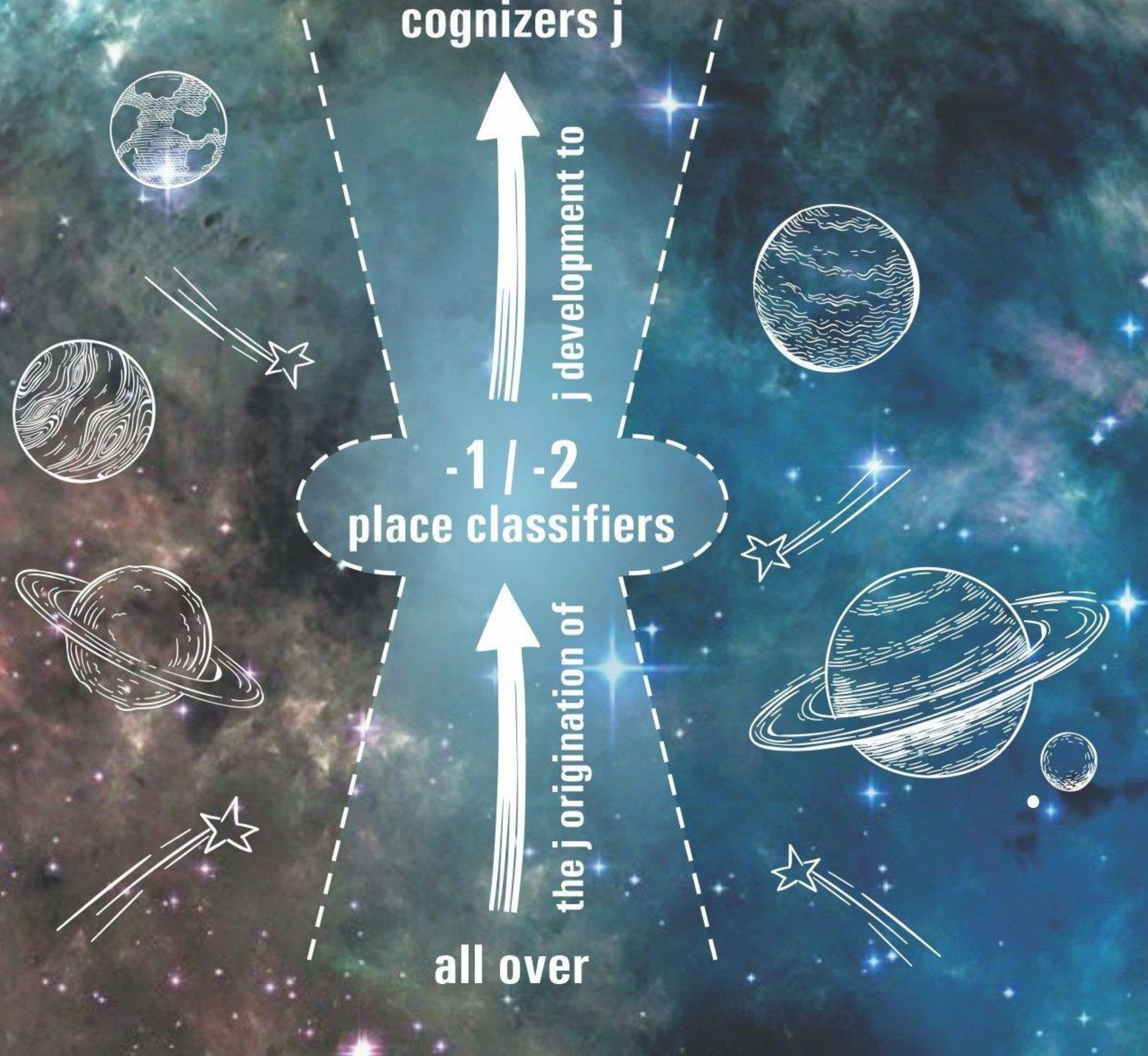
National Academy of Sciences of the Republic of Armenia

Institute for Informatics and Automation Problems

Edward M. Pogossian

CONSTRUCTING MODELS OF BEING BY COGNIZING

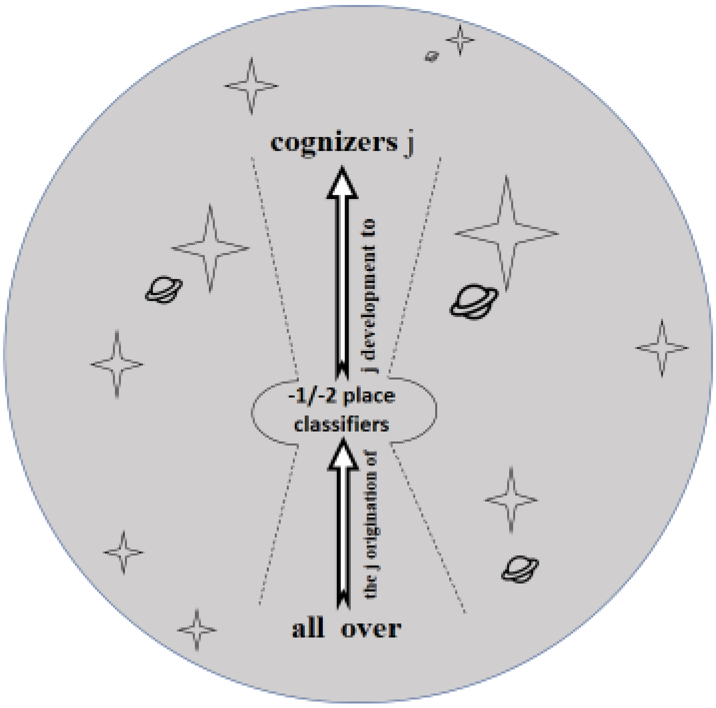


National Academy of Sciences of the Republic of Armenia

Institute for Informatics and Automation Problems

Edward M. Pogossian

CONSTRUCTING MODELS OF BEING BY COGNIZING



YEREVAN 2020

National Academy of Sciences of the Republic of Armenia

Institute for Informatics and Automation Problems

Edward M. Pogossian

CONSTRUCTING MODELS OF BEING BY COGNIZING

Publishing House of the Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia

Yerevan 2020

The book is published by the decision of the Scientific Council of the Institute for Informatics and Automation Problems of NAS RA on September 3, 2020.

Managing editor: **Anahit Chubaryan**, Doctor of Physics and Mathematics, Professor, Yerevan State University

The book is recommended for publication by the reviewers:

1. **Suren Khachatryan,** Assistant Professor, PhD in physics, American University of Armenia

2. **Kristina Sargsyan,** Dean of the Faculty of Informatics and Applied Mathematics at the French University in Armenia, PhD in computer sciences.

Գիրքը հրատարակվել է ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի գիտական խորհրդի որոշմամբ, 2020թ. սեպտեմբերի 3-ին:

Պատասխանատու խմբագիր. **Անահիտ Չուբարյան**, ֆիզմաթ գիտությունների դոկտոր, պրոֆեսոր, Երևանի պետական համալսարան

Գիրքը առաջարկվում է տպագրության հետևյալ գրախոսների կողմից.

1. **Սուրեն Խաչատրյան**, ասիստենտ պրոֆեսոր, Հայաստանի ամերիկյան համալսարան, ֆիզմաթ գիտությունների թեկնածու

2. **Քրիստինա Սարգսյան**, Հայաստանում ֆրանսիական համալսարանի ինֆորմատիկայի և կիրառական մաթեմատիկայի ֆակուլտետի դեկան, տեխ, գիտությունների թեկնածու

UDC 004.8 ISBN

E. Pogossian. Constructing Models of Being by Cognizing. Academy of Sciences of Armenia, Yerevan, 2020, 492 pages.

1. Following Jean Piaget, we interpret cognizing as mental doings on learning and organizing mental systems (mss), while learning and organizing are mostly reduced to revelation and acquisition of mss accumulated in communities.

To ground the fundamental hypotheses of Piaget that cognitive doings are learned stage by stage from certain root doings to the highest ones by means of only a few rules, we construct an extension of object-oriented programs*, **mentals,*** to approach the expressive power of natural languages in representation of realities and confirm the adequacy of mentals in modeling cognizing by several criteria.

Then we argue that mentals, including the rules of their cognitive development, are reducible to certain roots, including 1- and 2- place classifiers, and a chain of development of classifiers to various cognitive mentals can be tracked based on the

- algorithms (inductors) of revelation of 1/2 place classifiers of increasing abstractness, aimed to model the law of accommodation by Piaget, and
- procedures of acquisition of mss from communities and their processing for several cognitive doings, aimed to model the law of assimilation by Piaget.

2. Then, we question whether these 1/2 place classifiers can originate in Nature and develop to the highest levels of cognizing allowing to construct their own AI robots.

A positive answer to this question could resolve the mystery of the origin of cellular realities, since their enormous complexity cannot be attained by chance but it can only be constructed.

The promises of the positive answer rely on the uniformity of the measures of negentropicity by Schrödinger and information by Shannon as well as on the hypotheses of physicists, referring to J. Parondo, that information (and, therefore, negentropics) can originate in Nature, while information and classifying are, we assume, inseparable from each other.

3. Consequently, one may question whether this origination is unique to our solar system or is manifold in the Universe? Following the assumption that conditions similar to those of our solar system are manifold in the Universe. Thus, we can assume that powerful cognizers can originate in various regions of the Universe and self-develop to the highest levels allowing them to reproduce themselves in various modes.

Keywords: Piaget, cognition, adequate, constructive, modeling, classifiers, neuron nets, negentropics, information, cellular, origination, anthropic principle.

The Meaning of It All
Richard Feynman
Whether It Is the Brain We Exist in…
Michael Alkire
Can God Be Modeled?
Thomas B. Sheridan
God is Nature !
Albert Einstein

CONSTRUCTING MODELS OF BEING BY COGNIZING

Edward M. Pogossian

Cognitive Modeling Direction at the
Institute for Informatics and Automation Problems of the
National Academy of Sciences of the Republic of Armenia

e-mail: epogossi@aua.am

Abstract

1. Following Jean Piaget, we interpret cognizing as mental doings on learning and organizing mental systems (mss), while learning and organizing are mostly reduced to revelation and acquisition of mss accumulated in communities.

To ground the fundamental hypotheses of Piaget that cognitive doings are learned stage by stage from certain root doings to the highest ones by means of only a few rules, we construct an extension of object-oriented programs***, mentals,*** to approach the expressive power of natural languages in representation of realities and confirm the adequacy of mentals in modeling cognizing by several criteria.

Then we argue that mentals, including the rules of their cognitive development, are reducible to certain roots, including 1- and 2- place classifiers, and a chain of development of classifiers to various cognitive mentals can be tracked based on the

- algorithms (inductors) of revelation of 1/2 place classifiers of increasing abstractness, aimed to model the law of accommodation by Piaget, and

- procedures of acquisition of mss from communities and their processing for several cognitive doings, aimed to model the law of assimilation by Piaget.
2. Then, we question whether these 1/2 place classifiers can originate in Nature and develop to the highest levels of cognizing allowing to construct their own AI robots.

A positive answer to this question could resolve the mystery of the origin of cellular realities, since their enormous complexity cannot be attained by chance but it can only be constructed.

The promises of the positive answer rely on the uniformity of the measures of negentropicity by Schrödinger and information by Shannon as well as on the hypotheses of physicists, referring to J. Parondo, that information (and, therefore, negentropics) can originate in Nature.
3. Consequently, one may question whether this origination is unique to our solar system or is manifold in the Universe? Following the assumption that conditions similar to those of our solar system are manifold in the Universe, we can assume that powerful cognizers can originate in various regions of the Universe and self-develop to the highest levels allowing them to reproduce themselves in various modes.

Keywords: Piaget, cognition, adequate, constructive, modeling, classifiers, neuron nets, negentropics, information, cellular, origination, anthropic principle.

Acknowledgements

The author is grateful to all colleagues of *Cognitive Algorithms and Models*, *Theory of Algorithms and Math Logics and Constructive Mathematics* Directions for their fruitful teamwork and continuing partnership, to the leaderships of the Academy of Sciences of the Republic of Armenia and the Institute for Informatics and Automation Problems (IIAP) fort heir interest and support of our studies.
Thankful to family provided me with peace and love while working on this book.
Grateful to Dmitri Pospelov for incredible efforts in integrating researchers of cognizing, to Arman Kuchukyan for the frutful discussions of the world' classic "Undersatnding Computers and Cognizing ", Anahit Chubaryan , Suren Khachatryan and Kristina Sargsyan for strong revieing and fruitful inquiring of the work.

Grateful to Boris Karapetyan, Sedrak Grigoryan, Nelli Adjabyan, Parandzem Hakobyan for their help in preparing the publication and, specially, to Anahit Hovhannisyan-Janazyan for her hard work in correcting the texts.
Grateful to Levon Pogosian for continuing discussions of the main ideas with the hope to cooperate for disclosing *the Meaning of It All* together.

Content Pages

Part 1. MODELING

Part 1: MODELING

Chapter 1/1 Introduction

1. Introduction

1.1. Cognizing: Interpretation.

1.1.1. Being Inheritable. I myself, and, in general, we humans are somehow mainly predetermined by the genomes and cultures of our communities while what we contribute to our being personally in a lifetime is usually too little.

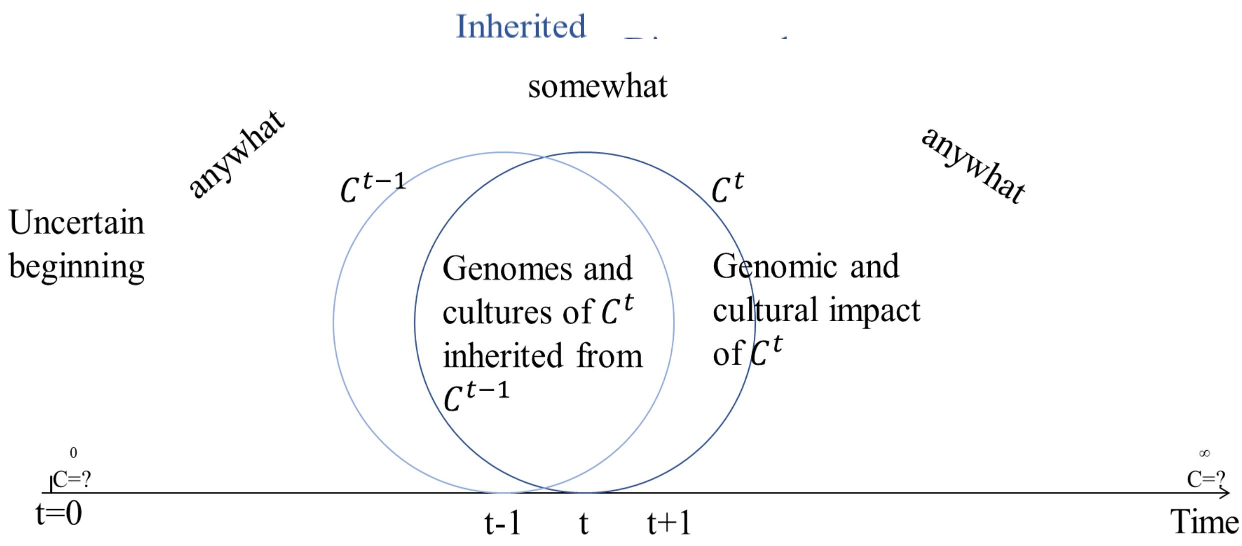


Fig. 1. Being of communities C inherited at time t, while uncertain at t=>0 and t=>∞.

What we are, includes the *roots* or inherited *utilities* that we enrich with new utilities in lifetime.

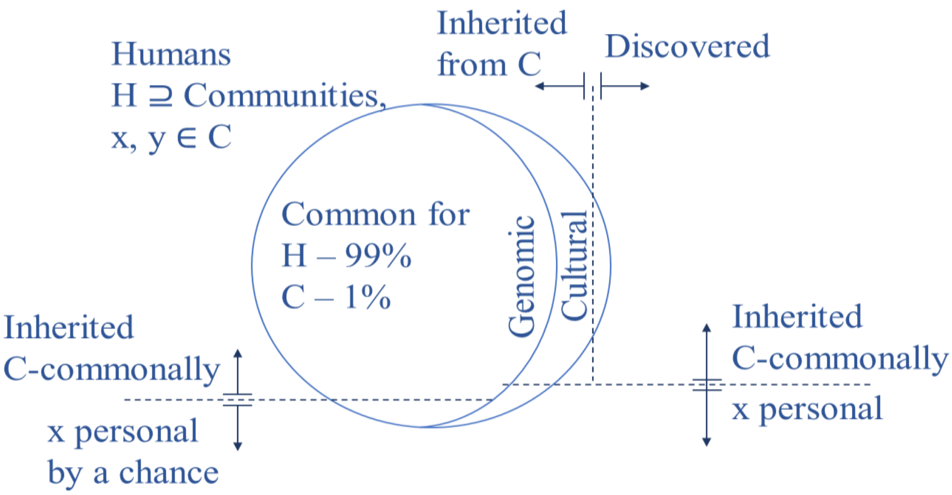


Fig.2. Types of doings of humans and in their proportions.

Our *roots,* first of all, cover doings on continuing to be non-entropic or negentropic by Schrödinger [6], comprising our *energizers*, then doings specific for cellular realities, or *cellulars*, especially the ones on diversified reproducibility.

The roots, *sensors* of all over, *effectors* to figure out our doings, overall *controllers* and some others embrace *octaves* of our cognizing.

1.1.2. Sensors along with other classifiers inherited and identified by controllers in conjunction with those studied and identified in a lifetime, i.e., revealed, discovered but mostly acquired from cultures of communities, comprise our *attributes*.

1.1.2.1. The outputs of attributes entail *imprints* in us that we classify to represent the *causers of imprints*, particularly those caused by impacts of a causer on our utilities.

The imprints, their causers and classifiers are our *realities,* while the totalities of realities comprise our *Universes*.

1.1.2.2. Regular enrichment of attributes with new classifiers lets us identify not only the causers of imprints themselves, but also determine the imprints caused by time-space relationships (rels) between the already identified causers.

The cause-effect time relationships are, particularly, represented by rules and regularities comprising the base of methods and algorithms.

1.1.3. All over we distinguish and categorize, the ways of their processing for reasoning, prognostication of doings, revelation of new classifiers, etc., are mostly inherited from communities.

And obviously, we have to learn from communities the correspondence, mapping of IDs of our innate and revealed basic classifiers into the communicatives of languages of communities to be able to acquire classifiers and their systems of the entire cultures of communities.

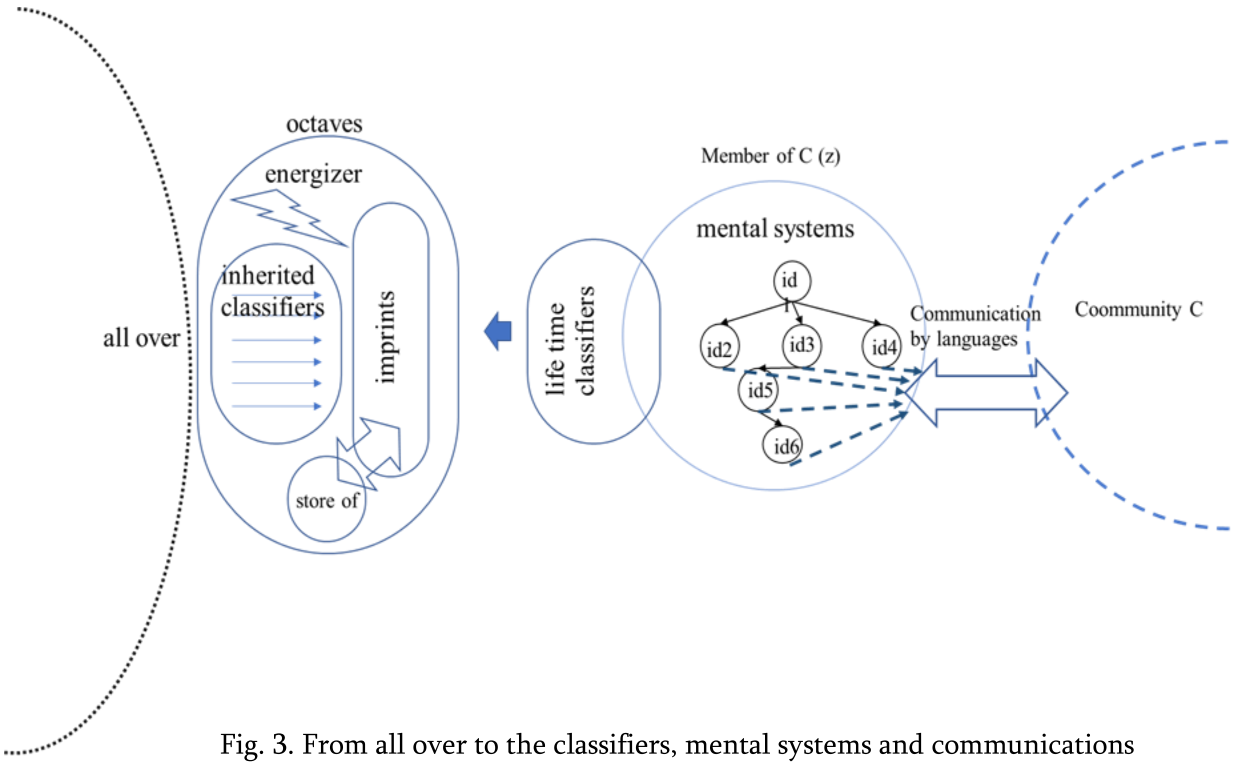


Fig. 3. From all over to the classifiers, mental systems and communications

1.1.4. Classifiers of the causers of imprints are represented in languages by categories of subjects, objects and relationships between them, or by 1- and 2-place relationships. New classifiers detailing, refining the primary categories, are incrementally being studied.

For example, *do* relationship in languages seems to be a primary category while *have* and *be* ones are categories detailing that *do* basic one.

1.1.5. Being an inherited and inseparable part of communities, we innately coordinate and direct our doings with the doings of communities and do that explaining /understanding our classifiers and their systems by the corresponding comunicatives already familiar to the members of the communities.

Ideally, these expressions should have activated compositions of IDs and their nominals equal to those of the explainers while being, in fact, some mosaics, expressions of units of languages they, in general, only bring closer to the ideal.

And an example of an attempt to present the IDs of the author’s systems of classifiers in English, is, in fact, this ongoing text.

1.2. The variety of classifiers and their compositions, ***Mental systems*** are processed to support the promotion of our utilities.

Constructively, mental systems (mss) are interpreted as *doins* and *systems of doins*, or ***mentals***, defined on the sets of doins.

In turn, doins (or *doers over IDs of nominated realities*) are interpreted as algorithms that use the IDs of imprints, the IDs of other algorithms or the IDs of certain innate or given classifiers as inputs [47,49,54,64,80].

For example, in OOP, the interpretation of doins corresponds to the algorithms that use either IDs of basic types (integers, symbols, etc.) as inputs, or IDs of other algorithms encapsulated in abstract classes, while mss correspond to systems of abstract classes incrementally ascended from ad hoc available ones by attributing, parenting and do types of relationships (rels).

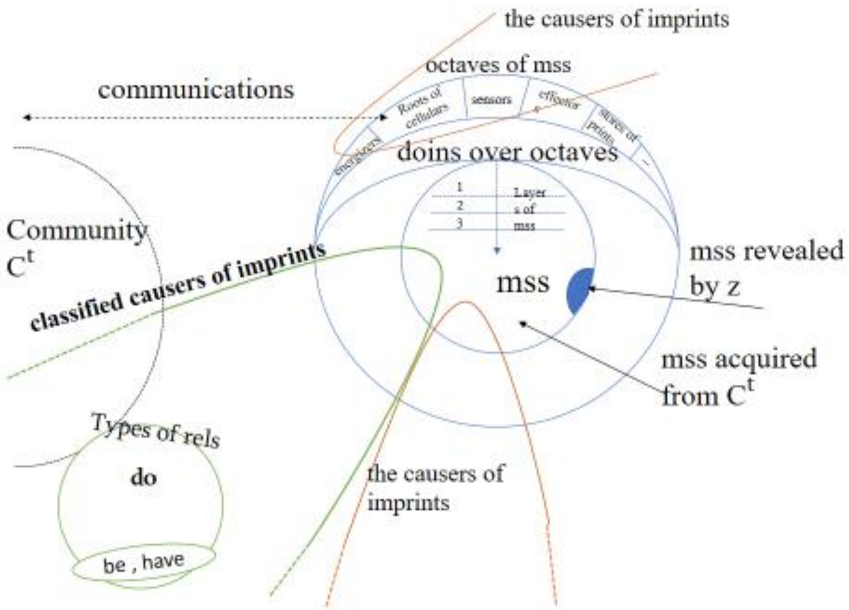


Fig. 4. A view on classified causers of imprints of members z of community C^t.
$\forall x \in C^t$: xM={mss of x}, CM=$\cup_{x \in C^t}$ xM. Assumption:$\exists$ z $\in C^t$ (zM=CM=M).
$\forall$m $\in$ M let's agree: m denote body of mss m and m denote the IDs of mss m.

1.2.1. Any mss *m* induce *systemic classifiers msCl* and, if *m* are doins, in addition, *m* also represents algorithmic, or ***do-*** *classifiers* mdoCl.

A powerful way to enhance the effectiveness of classifiers is to *regularize* them.

Namely, classifiers Cl of members x of communities C are regularized in C if accompanied by ontological in C methods, instructions allowing x regularly provide positive samples of inputs of Cl as well as let the members of C do the same by communicating with x.

In constructive regularization, those samples can be totally assembled from ad hoc units of the matter.

Regularly provided positives r of classifiers Cl and Cl themselves are interpreted as models of classifiers Cl' if r are classified as positives of Cl` , while Cl are interpreted as adequate models of Cl' if positives r meet certain additional requirements focused on positives of Cl.

For example, algorithms are adequate models of deterministic methods if, interpreting Church, equal algorithms can correspond to any method [30, 36].

If msCl and mdoCl classifiers are regularized, modeled or adequately modeled, either constructively or not, then mss m inducing those classifiers are named correspondingly.

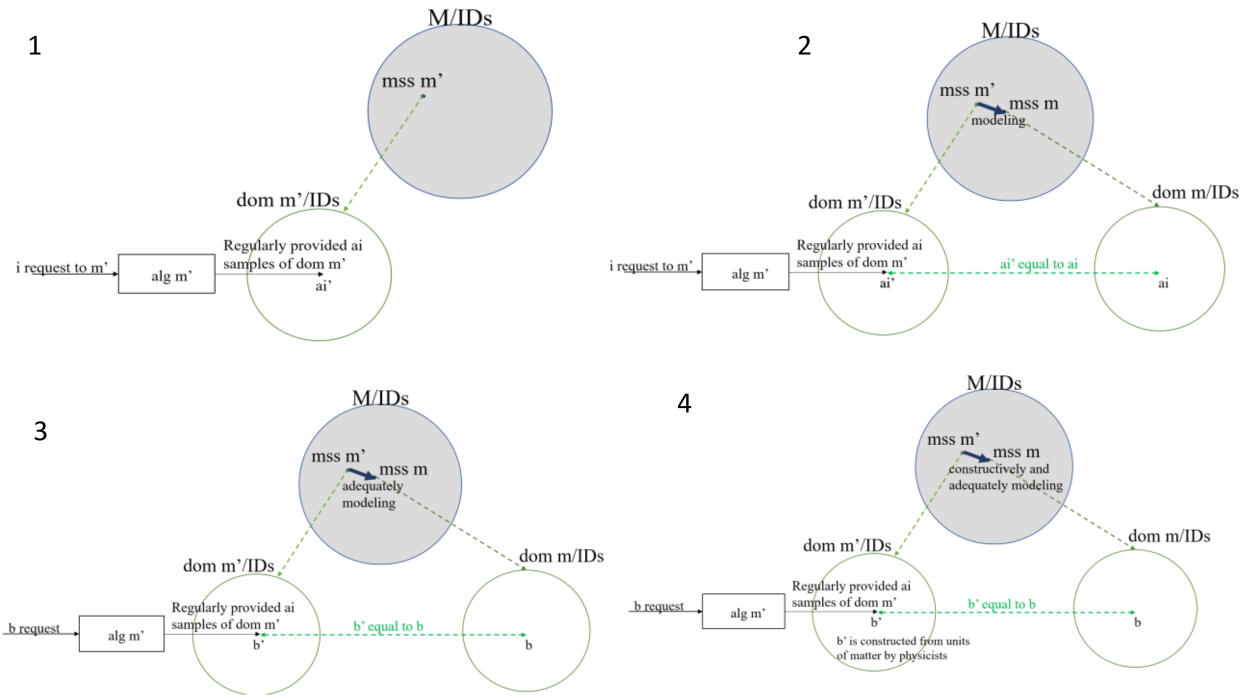


Fig. 5. Types of classifiers. 1. m' is regularized if ∃ alg m' that can regularly provide samples of dom m', 2. m' is modeling m if m' is regularized and samples of dom m' are samples of m as well, 3. m' is adequately modeling m if m' is regularized and ∀ samples b ∈ dom m alg m' can provide samples b' ∈ dom m' equal to samples b, 4. m' is constructively and adequately modeling m if m'

1.2.2. Mss altogether comprise *mental thesauri* that can, particularly, be represented by color-oriented graphs or nets, where connectivity subnets rooted in identified nodes **a** denote complete mss with IDs **a** or the *meaning of* ***a***, while partial subnets with the same roots denote parts of mss **a** or the *partial meanings of* ***a***.

1.2.3. Thesauri can be communicated, i.e., explained and interpreted. Namely, they can be explained to communities by outputting evolved *communicatives* (cms) of mss, i.e., IDs of doins and mss of ThC, as well as the samples of regularized do classifiers and rels of ThC, and interpreted by activating mss of communities by inputted cms.

1.2.4. Units of corpora of natural languages (NL) of communities C, say, more than 300 thousand units of English, represent the cms of mss of unified thesauri ThC of the C members.

1.2.5. Cms consequent to their meanings, roles in communications, etc., comprise a variety of classes represented in science, grammar and models of NL by the corresponding classes of NL units.

For example, physicists emphasize space and time rels, grammars of NL distinguish nouns, verbs, adjectives, propositions, as well as clauses comprised of subjects, objects and rels between them.

English grammar classifies rels, particularly, by categories *have, be, do* stating simultaneously that almost all rels are of the type *do* one. Other NL models, say, UNL [53], in turn, distinguish about 50 universal categories of rels.

Consisting of both of these categorizations, let us assume that the union of about 50 *do* type classes can cover almost all rels.

1.3. Doings of mental systems or **Mental doings** are a variety of doings based on the classification of realities as favorable or damaging with respect to our utilities to benefit or avoid them, correspondingly, thus supporting our being in the Universe.

Psychologists and psychiatrists classify mental doings consequent to the dimensions of doings of humans. Particularly, mental patterns of everyday life include adaptation to self-care, health and safety, social interactions and transactions at home, at school and at work, memorization of basic instructions, personal data (name, address) and important interests, goals setting and problem-solving, judgments, as well as doing integrated, i.e., setting goals, making decisions, then judgment of consequences, basics of cognizing and social and communicating, social relationships and norms including family, humanistic and ethical ones.

1.3.1. Focusing on the modeling of cognition, apparently, it should be questioned what mental doings identified by psychologists are unavoidable in effective constructive cognizing. Then, following [8], it is worth prioritizing those that at the time are denotative.

Particularly, it is necessary to question and deeply argue the need for modeling the will, emotions, consciousness and self-awareness in effective constructive cognizing.

1.4. So far, cognitive doings or ***cognizing,*** we assume, *are mental doings for learning and organizing mss.*

Learning and organizing mss are mostly reduced to revelation and acquisition of mss accumulated in communities and, in essence, guided, supervised by them.

Particularly, revelation is assumed to be goal-oriented, thus, motivated, and includes doings of inductive, deductive, imaginary and intuitive inferring of mss, enhancement of effectiveness of mss, processing mss to search for or prognosticate classifiers and strategies.

In turn, the effectiveness of mss can be raised by their regularizing, constructive and adequate modeling, etc.

Acquisition of mss includes learning from teachers, texts and other presentations of cms.

1.4.1. In general, cognizing we are emphasizing, meant to be consistent with the assertions by J. Piaget and, especially, with those on the life long universal engines of cognitive development.

Simultaneously, understanding and evaluating the ongoing research and findings comprehensively presented in [20], we are doing our best to keep up with new ideas and models to examine, modify or enrich our own ones.

1.5. Within the **Framework of cognitive doers**, then, **cognizers**, it should be acknowledged that cognitive doers, at least, include a type of doins, cognitive algorithms, focusing, learning and organizing mss, while controllers, we assume, are algorithms focusing on processing of mss aimed, in general, to govern any doings.

Then, cognitive algorithms being realities can be cognized, and therefore, enhance the effectiveness themselves.

Thus, we assume, that

cAss1. *Cognitive doers include cognitive algorithms on learning and organizing mss.*

cAss2. *Cognitive algorithms cognize realities, thus, they can cognize and develop themselves.*

1.5.1. Cognizing algorithms along with the entire mss support the preservation of negentropicity and means of that preservation.

Focusing on study of detached from the origin of cellulars, constructed cognitive algorithms, it should be acknowledged that cognizing algorithms along with the entire mss, should, at least, be supported by certain negentropics, *octaves*, containing

- *energizers,* i.e., means of energy supplying, its storing along with ones of preserving these utilities, in other words, the means of negentropicity,

-means enriching and developing ones comprising negentropicity, particularly, additional sensors of not only of energy, +sensors, also +effectors, storages of prints and basic classifiers, +controllers.

1.5.2. ***Cognizers, or cogs,*** are negentropics over octaves that in collaboration with analogous communities are cognizing the Universe, i.e., they learn and organize mss for preserving their utilities.

Cognizers cover cellulars, autonomous agents, as well as solvers of combinatorial problems tending to the effectiveness of experts [43].

1.5.3. By definition, learning mss at any stage assumes certain mss thesauri, including certain cogs.

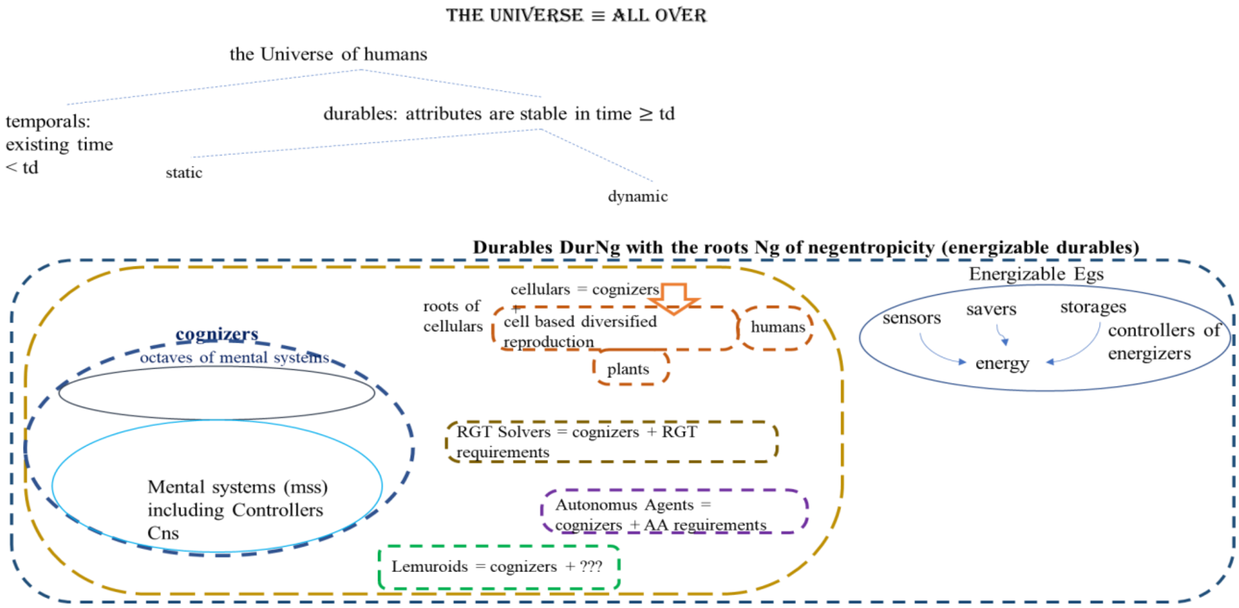


Fig. 6. Positioning cognizers in the Universe.

And, apparently, learning cannot begin without some min, root thesauri (rTh) and cogs.

1.5.4. In other words, we assume, that

cAss3. *Cognizing at time t aims to develop thesauri Th' at t into Th" at t+1 with respect to (wrt) the attainment of ad hoc goals* (actual enhancement of effectiveness of Th" wrt Th' needs to be proven).

cClr.1.3. *Cognizing requires the existence of certain starting root thesauri rTh that necessarily include certain root cognizers, comprising root negentropics rNg, octaves of mss and mss.*

cClr. 2.3. *Root cogs, or rcogs, should be able to develop root rTh and themselves to the highest human thesauri HTh, including the highest human cognizers Cogs.*

In other words, rcogs learn mss, develop and organize mss (including themselves) to attain the highest human cognizers, mss and their organizations.

The last corollary, apparently, shouts on the conviction of J. Piaget [4] that humans inherit the kernel of cognizing, controlling and implementing their entire cognitive development.

1.5.5. Later on, let us assume that HTh and Cogs coincide to address only to Cogs because rcogs are the crucial part of rTh while doings of Cogs are unavoidably based on sufficient thesauri.

1.6. Constructing cognizers. The above assumptions and their consequences raise fundamental questions on

-refinement of the scope of constituents and doings of acknowledged human cognizers, Cogs, including the highest picks of cognizing, followed by

-construction models adequately representing the acknowledged scope of human cognizers

-specification rcogsai capable to develop themselves to CogsAI adequate to Cogs, finally,

-construction models of rcogsai implementing the above specification.

1.6.1. Thus, **the problem of Constructing r*C*** arises that questions *the construction of root rcogsai that have constituents inevitable for cognizing and are able to self-develop with those inevitable constituents up to CogsAI, functionally equal to Cogs*, i.e., up to becoming adequate models of Cogs, therefore, in turn, capable to construct CogsAI' equal to themselves, i.e., to CogsAI.

Construction of r*C*, in fact, has not yet been completely resolved and remains one of the central fundamental problems of AI.

Let us provide some aspects of our approach to solving the problem.

1.6.2. In specifying rcogs, we approximate the scope of acknowledged human cognizing by cognitive doings already listed above.

1.6.3. We start the specification of rcogs with assumptions that the scope of cognitive doings can be represented within the framework of our models while the root negs rNg and octaves of mss are already determined.

Nevertheless, a variety of heavy questions arise, particularly, on

- what doings, that is, algorithms, should be necessarily included in rcogsai, or what mss revealing, acquiring and organizing algorithms are inevitable for rcogsai rC*?

-can doings of rcogsai be equally represented by doings of unified units of classifiers?

Thus, we consider that the solution of r*C* is worth looking for step by step, starting with visibly solvable subproblems of r*C*, then, strengthening the requirements incrementally approach r*C*. And we temporarily omit the requirements to rcogsai to have only inevitable constituents for self-development, to the algorithms for the formation of constituents of cognizing and enhancement of the effectiveness of classifiers.

1.6.4. Addressing to the picks of Cogs, let us recall that researchers are approaching the constructive adequate models of cells and their genomic reproduction.

Simultaneously, AI models of Cogs become capable to learn higher mental doings D” for a variety of given starting doings D’.

And AI is questioning whether maxD” can be equal to doings of Cogs, particularly, if maxD” can be equal to doings of Cogs in constructive modeling of cells and their incremental development.

1.7. This work addresses to our research in answer to the above and consequent questions on specifying, construction and possible origination of rC*.

The results are split into three parts that concisely can be represented as follows.

1.7.1. Part 1 presents constructions of models and reasoning around them assumed to be self-sufficient for their understanding and analysis.
Let us first sketch the contents of the Chapters of Part 1, and then introduce, we hope, the impact of our constructions on the modeling of cognizing.

1.7.2. Following the Introduction, Chapters 2-5 of Part 1 are devoted to the construction of adequate models of cognizing, detailing and deepening earlier argumentations on their validity.

Chapter 2 presents our interpretation of nova days views on cognizing, discusses the effectiveness of mental doings in dimensions of their picks, peculiarities and scales, and also recalls the earlier defined constructive models of mental doings.

Chapter 3 introduces the models of mss and our approaches to arguing their adequacy. Chapter 4 chains mental constructions to the root, basic classifiers, i.e., to root 1-/2- rels, argues the ways of their formation by inductors based on the given matrices of classified imprints, discusses the ways of learning mss and, in principle, the available perfectness of classifiers. Concluding Part 1 in Chapter 5, we discuss and provide premises on the origination of root cognizers and outline further research to prove the hypothesis by modeling.

1.7.3. In more details, Part 1 is devoted to our research in answers to the above and consequent questions on construction, specification and possible origination of rC^*.

1.7.3.1. Construction of models adequately representing r^*C^* will start with the assumption that the roots of negentropicity and octaves of mss are already predetermined.

Then we introduce a type of models of mental systems, *mentals* that expressively cover algorithms, OO languages and are approaching one of the natural languages.
Then, arguing the sufficiency of mentals for adequate modeling of human cognizing, we state that mentals, being consistent with algorithmic cognitive models inherit all models of cognitive doings so far implemented by those means.

We state that mentals consist of functional models of cognizing with connectivity models presented by artificial neural nets (ANN) allowing, particularly, to reduce ANN to the systems of classifiers composed of certain basic ones.

Moreover, mentals extend these models to ones allowing plausibly interpret the explanation and understanding of mss, as well as the entire human-computer communication of mss.
Adequacy of mentals is examined also in the frame of rich by applications class of combinatorial games, interpreted as models of human interaction with nature [40, 43, 44, 78].

Thus, in our nowadays belief mentals let adequately represent the scope of human cognitive doings.

1.7.3.2. In specifying r^*C^* we question the constituents of learning, organizing and processing mss, which simultaneously add to arguing the adequacy of mentals in cognitive modeling.
So far, noticeable procedural findings, we assume, are as follows:
- mentals, that is, mental systems, are decomposable up to the sets of classifiers of the standard types of 1-/ 2- place relationships,
- inductive algorithms, inductors, based on the matrices of imprints can form identified classifiers of the types of 1-/2-place relationships with varying degrees of abstractness,

- learning of mss, as a rule, is reduced to assembling of already learned rels into mss equal to those of communities and that assembling is supervised, explicitly or not, by instructions provided by cms of those communities.

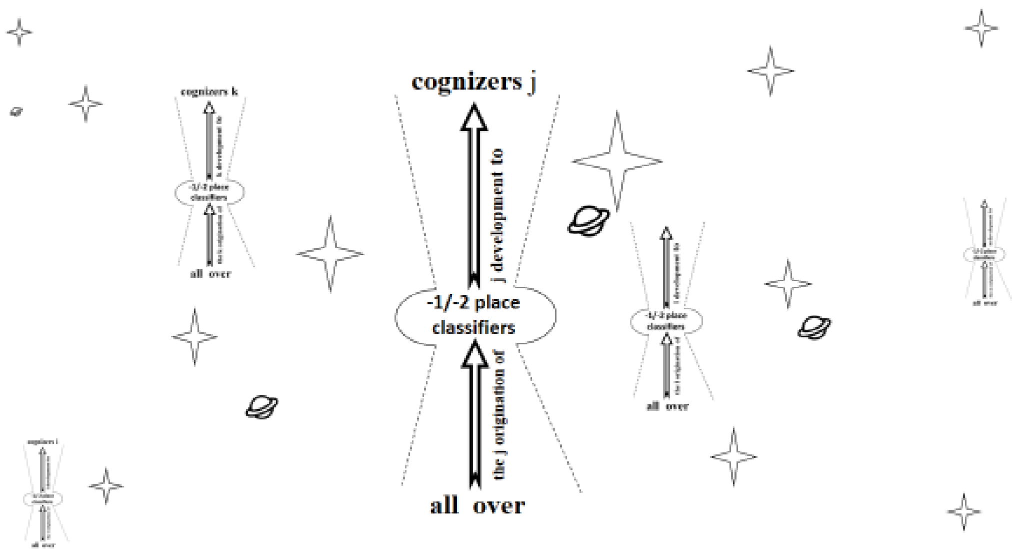


Fig. 7. From all over to various originations of -1/-2 place classifiers then to various ways of development to the highest cognizers.

1.7.3.3. The above findings in construction and specification of r*C*question the min number of inevitable constituents of cognizers capable of self-development up to ones equal to the highest Cogs ones.

The statement that all mss can be composed of basic types of 1-/2-place rels let us narrow the search for inevitable constituents of r*C* and reduce it to the search for algorithms of regular formation of 1-/2-place rels and their composition in a variety of mss.

The necessity of 1-/2-place rels of a variety of abstractness can be maintained by uniform inductors of the formation of rels.

Then, given proper assembling and search algorithms, the scope of learned mss can be enhanced by the acquisition of mss from communities, by their search in the spaces of rels, etc.

1.7.3.4. The end of Part 1 questions the origination of cognizers in Nature.
Advances in physics and AI lead to a significant consequence that realities not only of a cellular nature but also those that can be constructed, assembled from the ordinary units of matter available to humans, can attain the highest cognizing of the Universe, at least, comparable with that of the humans.

And since these artificial, constructed highest cognizers are only the assembles of units of matter unconstrained from the mysteries of the origin of cells it is expected to question the

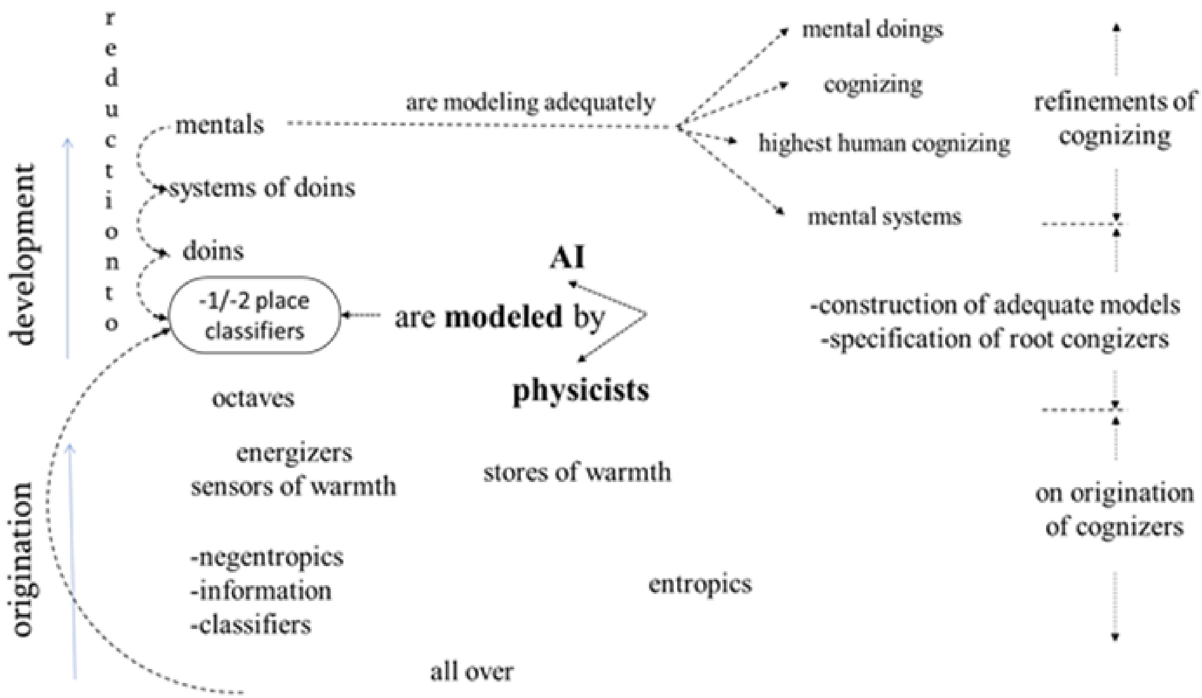


Fig. 8. On origination of -1/-2 place classifiers and their development to the highest cognizers.

necessity for humans or someones else in assembling, constructing the highest cognizers, thus, to question the feasibility of origination of highest cognizers within of the framework of the acknowledged knowledge in Nature.

Nawdays physicists approach the grounding of the origination of information, that is, the grounding of the origination of non-entropic or negentropic or living realities by Schrodinger. And since we believe that classification and information are inseparable from each other it becomes reasonable to question the origination of elementary classifiers in Nature followed by their development up the highest cognizers.

Possibility of origination of cognizers in Nature, then the development to the highest degrees of cognition letting their self-reproduction allow us to assume that cognizers can regularly, cyclically be both originated and constructed in Nature which resembls the regularity in origination of stars, their being, explosions followed by the birth of new stars.

1.7.3.5. Part 1 concludes the questions that we believe can expand and develop our research.

1.7.4. Part 2 addresses the interpretations of views on cognizing and mental doings enlightening the ground of models in Part 1.

Interpretations, while interesesting with the statements of Part 1, aim of be helpful in understanding the constructions of models, as well as in linking of ongoing views on cognizing with our models and views on expectations of their possible dominance.

1.7.5. Part 3, first of all, provides a series of **Appendices** with our publications aimed to enlighten the software and experiments in examining the adequacy of models, as well as provide examples of their successful applications in intrusion protection, defense, management and tutoring that seem to be the most interesting for further development of constructed cognizers.

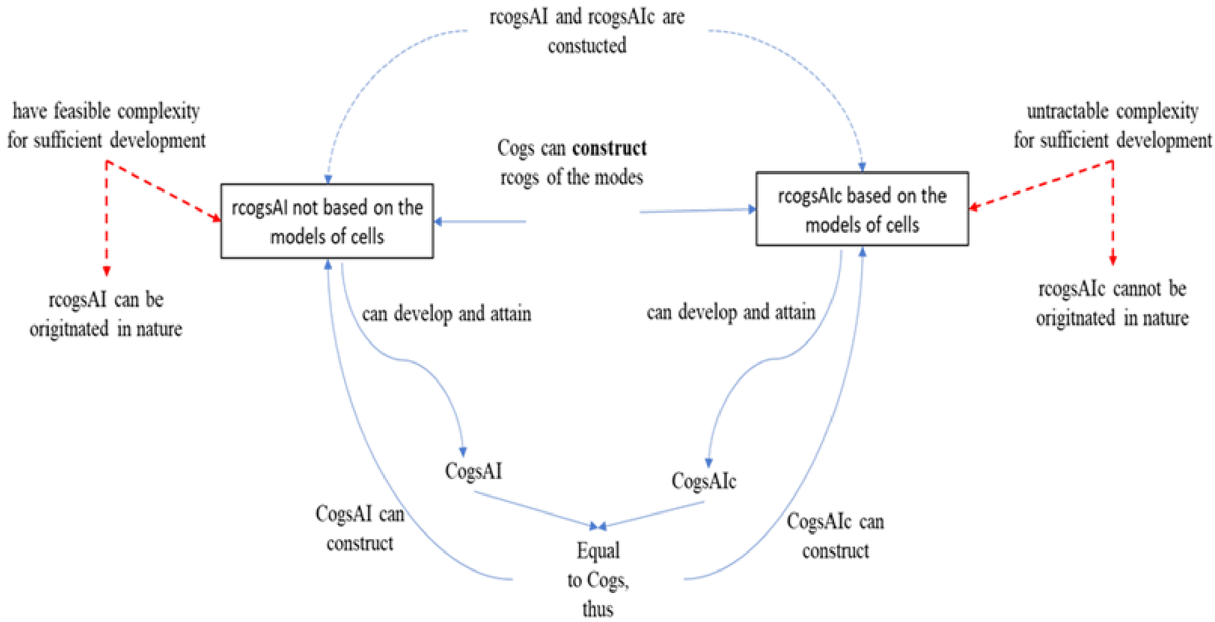


Fig. 9. Positioning the highest human cognizers (Cogs). A1-Ass.1. The highest human cognizers Cogs can construct root cognizers in the modes bases on the models of cells, as rcogsAIc, or not based, as rcogsAI, that developing can attain correspondingly CogsAIc and CogsAI both equal to Cogs,

- rcogsAI can be originated in nature
- developing rcogsAI can attain CogsAI
- CogsAI analogously with Cogs can construct rcogsAI as well as rcogsAIc that int is not excluded could construct cellular, i.e. be their ancestors

1.7.5.1. Then, Appendices of Part 3 refer to the theoretical frameworks on games, inductors, adaptation of combinatorial algorithms, theorems on the convergence of classes of algorithms to the most successful ones in the class by means of a series of regularized local tournaments, as well as a complete presentation of the theorem on systems of sets with a min number of subsets that is grounding the theorem on convergences [42].

Accompanied by the corresponding references, this Appendices address to the technique of the proving some of key theorems.

1.8. Our models are based on and try to fuse the findings of many outstanding researchers. We refer to some of their publications [1-35, 52, 53, 55, 56, 58-63], as well as to some of our works [54, 36-51, 57, 64-81] both in Parts 1-2 and in Appendicies that can improve understanding of our ideas and their approbations presented in Part 1.

1.8.1. The appendices aim to present AI studies within of the framework of *Cognitive Algorithms and Models* Direction of the Institute for Informatics and Automation Problems (IIAP) of the Academy of Sciences of the Republic of Armenia and include some of the publications of *Boris Karapetyan, Artashes Martirosyan, Robert Simonyan, Karo Chalikyan, Mushegh Hambartcumyan, Yuri Harutunyan, Nikolay Khachatryan, Rafael Kamalyan, Levon Djandjulyan, Karen Grigoryan, Vahan Alanakyan, Emma Danielyan, Arsen Javadyan, Arthur*

Grigoryan, Vachagan Vagradyan, Tadevos Bagdasaryan, Sedrak Grigoryan, Zaven Naghashyan, Karen Khachatryan, Nairi Hakobyan.

A more complete list of publications of the Direction is presented in References.

1.9. "*Cognitive Algorithms and Models"* and *Theory of Algorithms* Directions in IIAP, led by Edward Pogossian since 1973 and *Hrant Marandjian* since 1975, correspondingly, were branched from the Laboratory of *Mathematical Logics and Constructive Mathematics* led by *Igor Zaslavski* since 1963 the noble traditions of the school by Andrey Markov, one of the founders of Computer Sciences along with Kleene, Church, Turing and Post.

Since 2014 Cognitive Models Direction deeds in Computational and Cognitive Networks Division by Vladimir Sahakyan.

AI studies were initiated at IIAP since its foundation in 1957 by an outstanding mathematician *Sergey Mergelyan,* including studies of full value chess masters-computer communications that should be supportive and examined by *Tigran Petrosyan*, world chess champion.

1.9.1. Humans have long been viewed inseparable from their cellular nature, evolving step by step to self- identification by the patterns of mental doings.

Nowadays, *Cognitive Models Direction,* interpreting Alonzo Church, and then Allan Turing, formulates its vision on the fundamental societal impact of AI as an understanding of being of humans through cognizing and, possibly, their origination.

In other words, we interpret *Cognitive Models research direction* as *a branch of science aimed at understanding humans by providing adequate constructive cognitive models, at least, comparable in effectiveness to mental doings of humans* and for that

- refining human ways of cognizing the Universe
- specifying adequate constructive models of cognizing
- revealing constraints on cognizing
- alternating means of cognizing the Universe.

Chapter 2/1: Specifying Mental Models/Mentals

Preface

In Chapter 2/1, we are going to move from ad hoc ontological classifiers of mss to specifications of adequate constructive models of these mss.
For that, first, we refine classifiers, specifications, modeling and scales of effectiveness and efficiency (ee) of mss to argue they are consistent with those accepted by scientific communities.

Then, specify a type of systems, constructed sensor-based doers systems, ***mentals,*** arguing that they areconsistent with algorithms and object-oriented (OO) programs, therefore, with ad hoc algorithmic and OO adequate models of certain mss.

Finally, question the adequacy of mentals analogously to one of the algorithms by Church, adequacy by consistency with connectivity neuron net models and by effectiveness of on the job performancesof models integrated into cognizers for solving certain problems in competitions with other solvers

1. Introduction

1. Our approach to specification and modeling of mental systems relies on the ideas of methodology for the transition from ad hoc classifiers of computable functions and deterministic methodsto their specifications, which are the algorithms (Algs), introduced by Turing, Church, Post, Markov [29].

The methodology states that Algs are constructed models of the class of Deterministic Methods (DM), since they belong to DM and can be regularly provided.
Simultaneously, Algs are supposed to be adequate models of DM, since, according to the assumption of Church [30], an equivalent Alg can be corresponded to every DM.

Following the same methodology, we look for comprehensive ontological classifiers of msystems, and then specify constructed sensor based mental doers, ***mentals***, which are modeling input realities of target classifiers and can pretend to be their adequate models, i.e., a certain equivalent mentals can correspond to any classified mental system.

1.2. Natural languages are acknowledged by humans as the most comprehensive representation of the Universe.

While Algs can model any computable functions, logical and inductive inferences, search of classifiers and strategies in games, a question remains open whether their expressive power is enough for adequate modeling of natural languages.

Therefore, by having prove that mentals allow one to successfully specify adequate models of natural languages, we would have essentially advanced in arguing that the potential power of mentals in representing the Universe can be comparable with the mental systems of humans.

1.3. Ad hoc views on being, cognition and their modeling are widely spread in icebergs of theories and prominent publications including those in [1]-[32].

However, questions regarding the constructive theories of origin of humans and, in general, cellulars, constructive models of the formation of cultures, their enrichment and acquisition in communities remine open.

In our works, we try to contribute to these studies, advancing in specifications of regularized classifiers of the essentials of these views as we understand them.

We also question the adequacy ofmental models, mentals, and provide criteria for measuring adequacy of mentlas, followed by the presentation of our advances in the examination of their adequacy.

To evaluate adequacy by the job performances in consequent parts of the work we suggest constructive models for the formation of classifiers and strategies in competition and defense problems, approaches to their regular adaptation, enhancement of effectiveness and their measurement.

That evaluation is examined in a class of combinatorial games including, along with chess, the ones reducible to chess, such as problems of defense of networks from unauthorized penetrations or navies from rocket attacks, optimal management strategies formation in oligopoly competitions or in supply chain management, etc .

2. Constructing Mental Systems

1. In what follows *doers* and their *NRl systems* are specified to be later expanded to certain constructions, *mentals,* claimed as adequate constructive models of mss.

2. *Doers*, in general, we assume, are realities having input-output parts and for available *inrealities,* i.e., for realities, and moreover, some extent, at the input parts, either elaborate certain output realities or remain passive.

In- out- realities comprise their *in- out- domains,* or *in- out-doms.*

Indoms wrt outputs are split into classes of equality, thus, the absence of outputs corresponds to the class (?) of uncertain inrealities.

3. Doers are *do-classifiers* Cl if indoms are split into two classes +Cl and ?Cl; otherwise they are *corresponders, cors.*

Apparently, identifiers of do-classifiers Cl by themselves are sufficient to indicate their classes of equality, i.e., the positives +Cl, while classes of cors can be indicated by pairing those identifiers with the corresponding outputs.

4. Realities I{i} are *identifiers, IDs* of realities R{r} and doers Z{z} wrt Z if

- to any r, z there, z correspond unique IDs i(r), i(z),
- certain classifiers are linked any r, z allowing z to recall the corresponding r, z by i(r), i(z),
- any r can address to any z by certain requests, say, recalling some r, z.

Identified realities of the given R, Z paired with their IDs are named *nominals* wrt Z.

4.1. Classifiers of n-tuples of nominals are *n-place relationships*, or *nrels*.

Since later on we mainly address the classifiers of nominals they will be interpreted as 1rels while 2rels will be shortly named *rels*.

Rels (a, b) can be dependet or not on the orders of their arguments.

5. *Systems H over nominals Nls containing rels Rls*, or *systems H over Nls/Rls*, or *NRl systems H,* include Nls as systems and if H' are systems of H then systems of any subset of H' linked to each other by rels of Rls and nominated by IDs consistent with nominations of Nls are systems of H as well.

5.1. The totality H of NRl systems comprise *NRl nets* where the nodes a, b are IDs corresponding to systems of H, the edges (a,b) correspond to rels between the systems represented by the nodes a,b , signed, "colored " by IDs of those rels and oriented from b to a if correspond to rels(a,b).

5.2. NRl nets are, in fact, colored and oriented nets where the *nodes a depend on the nodes b* if rels (a, b) correspond to the edges linking a and b. Assuming the Nls are *1st layer* systems of NRl nets, the systems of *n+1- layers* are formed as systems over nominals of the n-th layers and rels Rls.

3. Systemic Classifiers vs. Do Classifiers

1. *Do classifiers* Cl are defined as a type of doers that for realities at the input parts either elaborate certain outputs or remain passive, thus splitting the indoms into two classes +Cl and ?Cl.

1.1. Doers of the type of classifiers are *sensors* and their outputs are *percepts* if inrealities are not necessarily pre-classified.

2.We assume that not only doers, but also systems over nominated doers, particularly mss h induce certain systemic classifiers hsCl with positives +hsCl.

Let us outline the classifiers induced by the systems h over the nominated doers while the details of such classifiers for the decomposed mss h will follow.

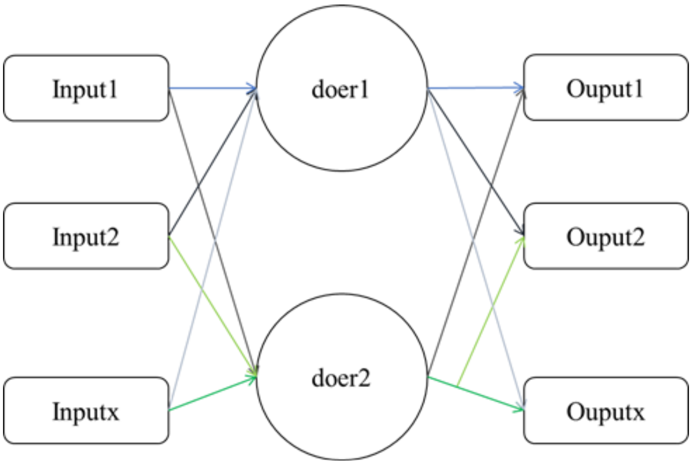


Fig. 10. Equal doers.

3. Realities r, r' are (fuzzy) equal with respect to doers d if the outputs (fuzzy) of d applied to r,r' are the same. In other words, d analyzing r, r' by their embedded regs doesn't find any (fuzzy) distinction between r and r'.

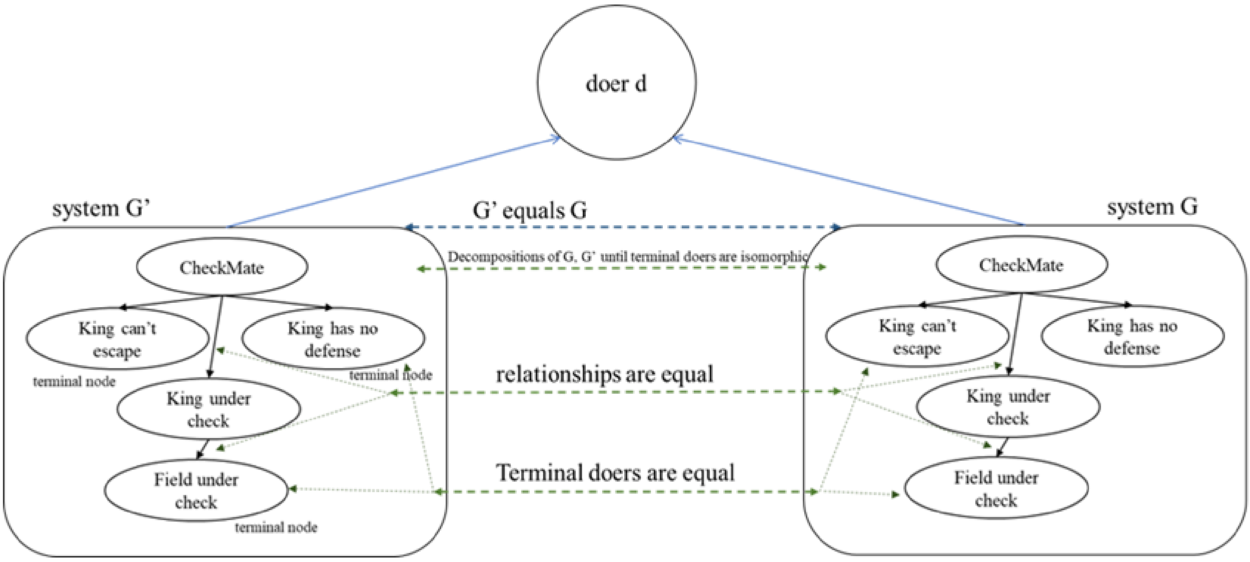


Fig. 11. Equal systems.

Due to the equality of realities that can be incomplete, approximate or fuzzy later on in refining the equality we , as a rule, skip to name the option of their fuzziness.

4. *Doers are equal* if for any inputs their outputs coincide.

For constructiveness of this requirement, we assume that indoms of doers are made finite by some criteria. Thus, doers are equal if their performances, i.e., the input/output of the pairs, coincide for the inputs of their indoms.

5. *Systems G, G' over the given nominated doers are equal* if certain procedures, algorithms, in general, doers d determine that decompositions of G, G' until terminal doers are isomorphic whereas the relationships between the nodes and the terminal doers related to each other by means of d are, correspondingly, equal.

6. *Realities r match systems h* if certain doers d can reveal in r equal to h systems h'.

Thus, pairs (mentals h, d) determine *systemic classifiers hsCl* with positives +hsCl.

4. Regularized and Modeled Classifiers

1. Classifiers in their min mode identify some realities while if regularized they can regularly provide samples of their indoms.
In turn, regularized classifiers can be models or adequate models of each other either constructive or not, as follows.
2. *Classifiers Cl* of members x of communities C, x € C, *are regularized* if x can accompany Cl, first, by methods Clrgz letting x provide positives of +Cl regularly, and second x by ontological in C communicatives Clcms can explain Clrgz to any y € C why y can provide positives of +Cl.

For example, classifiers of produced goods, grown-up domestic plants and animals, services and skills provided hand to hand are regularized.

Then, scientists x discovered some realities *a* and classified them by regularized classifiers Cl can by Clrgz provide samples of *a* and by Clcms successfully explain to others how to do the same.

2.1. Classifiers Cl are *fuzzily regularized* if they are regularized, but to the extent that outputs of Clrgz only fuzzilyy match +Cl, and the fuzziness can range up to +Cl mismatch.
3. Let's recall that positives of classifiers are realities of indoms of do classifiers doCl, while for systemic classifiers sCl positives will be the systems of doers.

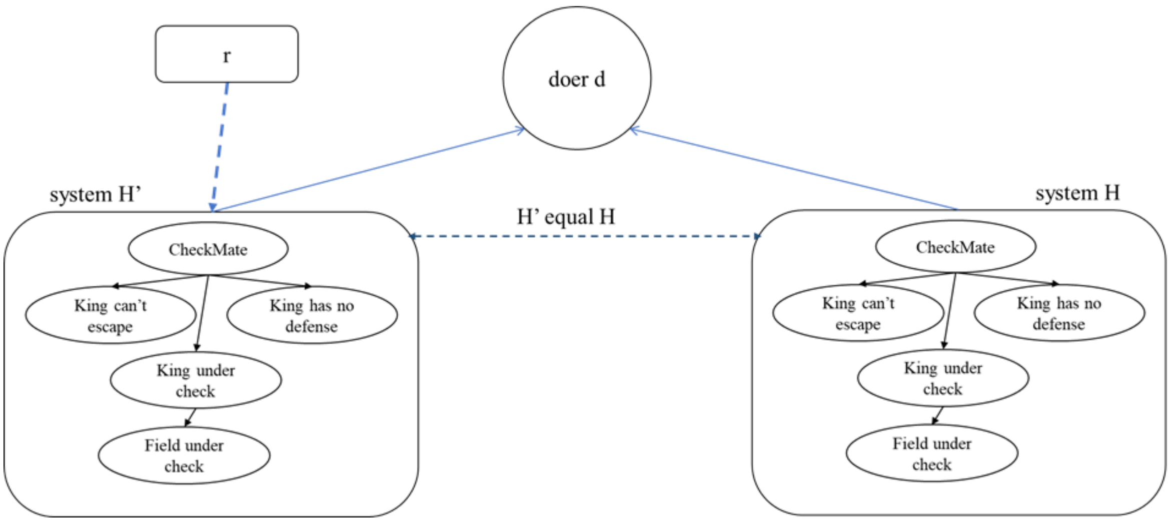


Fig. 12 Realities r match to systems h

Particularly, if mss h are some do classifiers doCl positives of +doCl will be some realities of indoms of +doCl (as it is realized in RGT Solvers), while positives +hsCl matching systemic classifiers hsCl will be doers including doCl themselves .
3.1. ***Corollary1.*** *Regularized mss h and associated with h methods hrgz are equal wrt out realities of hrgz, i.e., any realities reproducible by hrgz inevitably match h.*

Apparently, regularized mss h necessarily are systemic classifiers hsCL too either explicitly or implicitly.

Statements on mss, as a rule, also have transparent fuzzy interpretations, which can often be skipped later.

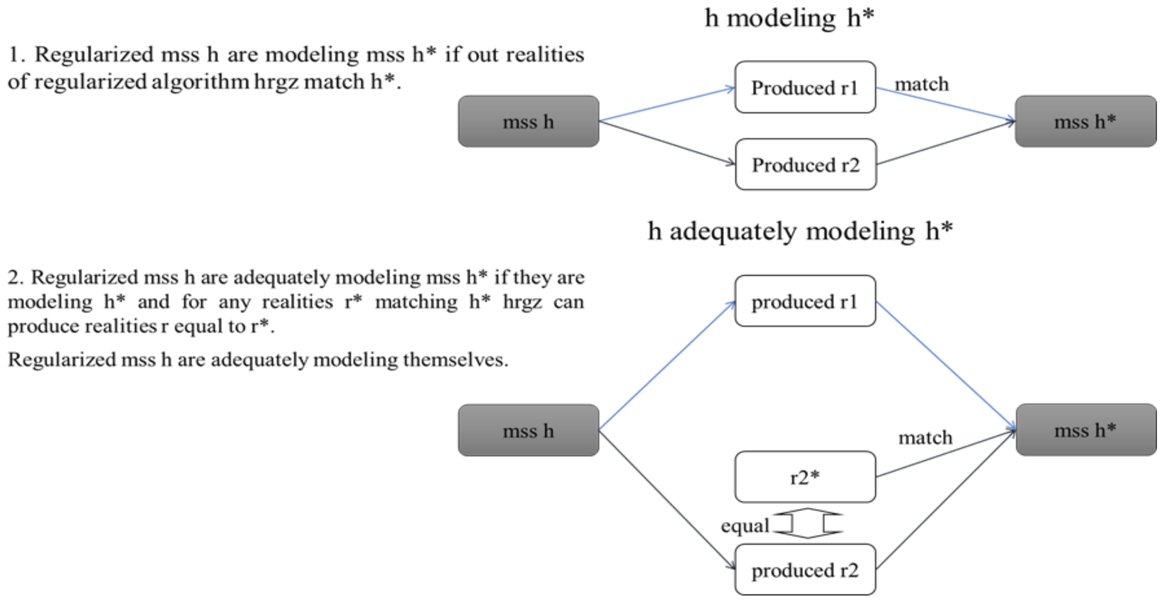


Fig. 13. Mss h modeling h*.

4. The samples of classified realities r can be constructed deterministically and totally independentof cellulars in *constructive regularization*, *conrgz*, or can be *regularized non constructively*, particularly be provided non deterministically, be grown up from a priory given realities like cells or crystals, be a product of services for humans or machines.

For example, mss Goods (Gds) representing producible within of the framework of some civilization goods, say, mss Computers (Cps), are conrgz since they are reproducible from totally non cellular realities.

Conrgz are mss Algorithms (Ags), represented either as Turing Machines, Post Productions, Markov algorithms or Recursive Functions that can be specified to be assembled from non-cellular units by mathematicians or programmers, and even more, they are enumerable.

Mss Wheat, Domestic Animals are fuzzily regularized and grown up while mss Services of cellulars, say, Treatments by Doctors, are fuzzily regularized and inseparable from cellulars.

Assuming a numerical scale for fuzziness of regularization its variable f might be zero for Cps, Gds and Ags, range from zero for mss Deterministic Methods (DM) to ½ for the Heuristics, and have f = 1 for mss Consciousness, Emotions, Passions questioned yet to be regularized.

4.1. Humans tend to regularized classifiers Cl` for the advantage of not only passively classifying but also actively providing positives of Cl'.

In addition, constructive regularization, let to extend leverages for amplifying target doings. Recall, for example, the transition from riding on horseback to moving by cars or trains.

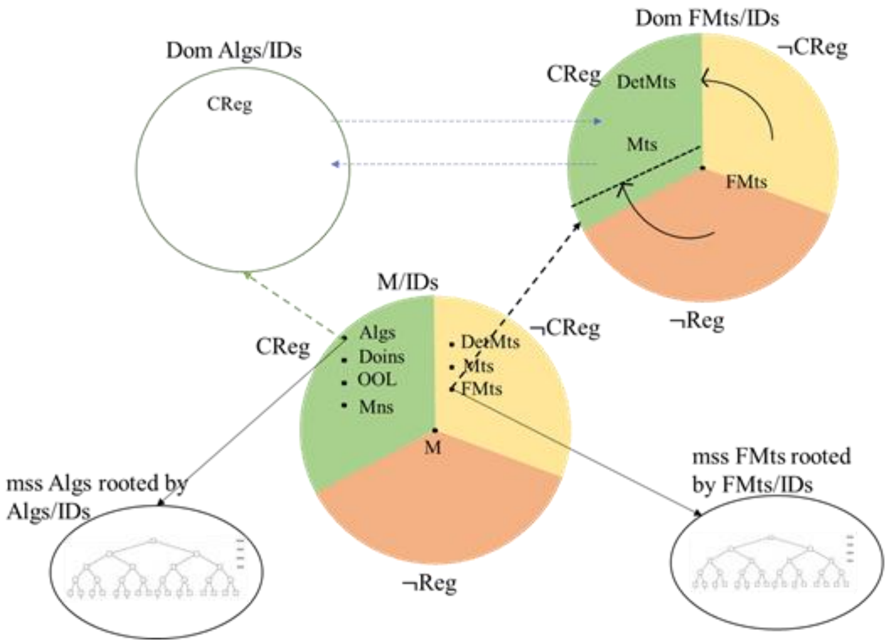


Fig. 14. Adequacy of Algorithms and Deterministic Methods by Church + (Extended) to Fuzzy Methods. Church's Hypothesis +: Algorithms are adequately modeling CReg Fuzzy Methods, Definition: mdom of mss m ∈ M, or the domain of matching m realities, are positives +sClm of systemic classifiers of m while IDs mdom of are IDs of +sClm, Examples: m = DetMtsDetMts dom = +sCl DetMts, IDs DetMts dom = IDs of +ClDetMts, m = Mns (Mentals), Mns dom = +sCl Mns, IDs Mns dom = IDs of +Mns

5. *Regularized mss h are modeling mss h** if out realities of hrgz match h*.
*Regularized mss h are adequately modeling mss h** if they are modeling h* and for any realities r* the matching h* hrgz can produce realities r equal to r*.
Apparently, the regularized mss h are adequately modeling themselves.

5.1. According to Church DM can be adequately modeled by Ags, i.e., equal Ags can be corresponded to any DM.
Recalling Corollary1, it can be stated
Corollary2. *Regularized mss h are equal to certain algorithms wrt out realities matching h.*

5.2. Church thesis, we assume, can be expanded for fuzzy and non deterministic algorithms and methods as well.
Namely, DM can be expanded to Methods (Mds) and Ags to fAgs adding to Ags, not deterministic, say parallel, probabilistic and heuristic methods / algorithms.
Corollary3. *Regularized mss h (fuzzy or not) are equal to certain algorithms (fuzzy or not) wrt realities matching h.*

5. Extending Constructive Mental Models

1. Let us recall that doers of the type of classifiers are *sensors* if inrealities are not necessarily pre-classified. Then, doers of the type of cors are *effectors* if inrealities are necessarily classified while they are *controllers* if both in- out- realities are necessarily classified.

1.1. Controllers Cns, are assumed, can assign IDs to the given mdoers aimed to control their processing and in- out- interactions with realities.

In the identification of realities R, Z, the realities Z can be interpreted as sets of Cns controlling in certain ways the realities of R analogical to servers of "star" types controlling networks of computers or, seemingly, analogical to unicellular controllers.

2. As it was acknowledged in Introduction cognizers are negentropics over octaves that in collaboration with analogous communities are cognizing the Universe, i.e., they learn and organize mss for preserving their utilities.

The kernel of octaves embraces the roots, means of energy supplying and storing, sensors, effectors and controllers specialized to preserve these utilities and all together to figure out doings on negentropicity of cognizers.

2.1. Constituents of octaves and the domains of their applications can enrich caused, particularly, by development of mss and, finally, aimed to enhance, directly or not, the effectiveness of negentropicity.

For example, controllers of octaves extend assigning IDs to the constituents of ostaves to the entire mss.

3. Let us name *otids* (**ID**s of **out**puts of **d**oers) the nominated wrt Cns outputs of doers, i.e., given doers d with IDs x name pairs (x, y) otids of d where y are some outputs of d, and name *alphabets of d* the sets of otids of doers d.

Then name *words in A1, A2, …, An the* sets of otids comprised from only some representatives of alphabets A1, A2, …, An of doers d1, d2, …, dn.

4. *Doers D over IDs of constituents of octaves*, or *doins D* (**do**ers **o**ver **I**Ds of **n**ominal**s**), are doers D nominated wrt to Cns of octaves while indoms of D are words in alphabets of the outputs of D and classifiers of octaves as well as over IDs of constituents of octaves.

4.1. Bundles of otids of Sns, Cns and oct doins at time t are *t prints* comprised into certain stores Pns of octaves and nominated wrt to their Cns.

4.2. *Systems of doins*, or *sysdoins*, are defined as NRl systems where Nls are doins.

The totalities of sysdoins comprise nets Nts.

5. While sysdoins can range from the sets of disjoined to the totally interconnected systems we refine the mental systems, mss, as those of sysdoins that are connectivity subnets G of Nts rooted in the nodes *a* of Nts.

Namely, *total connectivity sysdoins G rooted at nodes* ***a*** (or *total* ***a*** *connectivity sysdoins* G) *of Nts* are connectivity subnets of Nts rooted in ***a.***

And ***a*** *rooted* sysdoins *G'* of total connectivity **a** sysdoins G, or ***a*** *mentals*, or, generally, ***mentals***, are **a** rooted connectivity subsystems of G.

Then, *a1, a2, ..., an aspects of G'* are a1, a2, …, an rooted connectivity subsystems of G'.

Apparently, connectivity **a** sysdoins are **a** mentals and nodes a1, a2, …, an by themselves can be the aspects of G'.

The totalities of mentals of Nts comprise their *thesauri Th.*

5.1. *Decompositions of the 1st depth*, or *the 1st decompositions*, *of* ***a*** *mentals G* having nodes ***a*** on some layer k of Nts are a1, a2, …, an rooted mentals of all subsystems G1, G2, …, Gn of G with nodes a1, a2, …, an of k-1 layers of Nts connected to a.

And if G1, ..., Gn are the i-th decomposition of G then *i-1 th decomposition of G* will be the union of 1st decompositions of G1, …, Gn.

Apparently, the terminal decompositions of G will be comprised from doins of Nts.

The unions of i-th decompositions of G for I =1, …, k-1 comprise total decompositions of G.

5.2. Analogously j-th abstractions and total abstractions of ***a*** mentals G can be defined in such a way that thr 1st abstractions of ***a*** mentals G with nodes ***a*** at some layer k of ces nets Nts could be a1,a2,..,an rooted mentals G1,G2,…,Gn of all subsystems of G with a1,a2,…,an on the k+1 layers of Nts and connected to ***a,*** etc.

6. Thesaur Th are assumed to be stored by analogy with storing libraries, say in Java.
Namely, nodes **a** of nets Nts are stored with IDs of those nodes **a**, the classifiers of IDs of **a**, IDs of rels of nodes **a** with nodes **b** along with IDs of those **b.**

Nodes **a** corresponded to doins representing abstracts **d**, in addition, contain either decision- makers of **d** or references to them.

6.1. Apparently, nodes corresponding to doins/abstracts of nets Nts or in thesauri Th will coincide with abstract classes of Java in the case when their rels with other nodes of Nts are restricted by "attributed", "parented" and "done by" ones.

6.2. Started with identifying common for communities C doers, mdoers and msystems, so far, we have specified doins, sysdoins, nets Nts of sysdoins and thesauri Th of mentals as well as total connectivity sysdoins, their connectivity subsystems, mentals, and the aspects of mentals representing, we assume, mss and the aspects of mss.

We question whether mentals can be adequate constructive models for mss and for this purpose refine their constructive modeling and adequacy as follows.

7. The following types of sysdoins can equally correspond to algorithms, say in Markov or other equal modes.

Equal to rules by Markov are types of doins, regularities, or regs, corresponding certain otids to only some selected words of the indoms.
Algorithms are sysdoins consisting of regs by rels similar to ones comprising rules into algorithms by Markov [29,30].

Sysdoins, in fact, deepen the definition of algorithms by Markov detailing the origin of the rules. Namely, while Markov algorithms are defined starting with the given basic alphabets, the sysdoins assume certain classifiers, for example sensors, providing these basic alphabets.

7.1. Sysdoins of the types of "abstract classes", referring, say, to those in Java, are systems of algorithms, or methods, in rels of the types: "attributed", "parented" and "done by", with other abstract classes.

Abstracts expand abstract classes allowing arbitrary rels with other abstracts.

Finally, packages of abstracts and their libraries are mimicking the ones in Java.

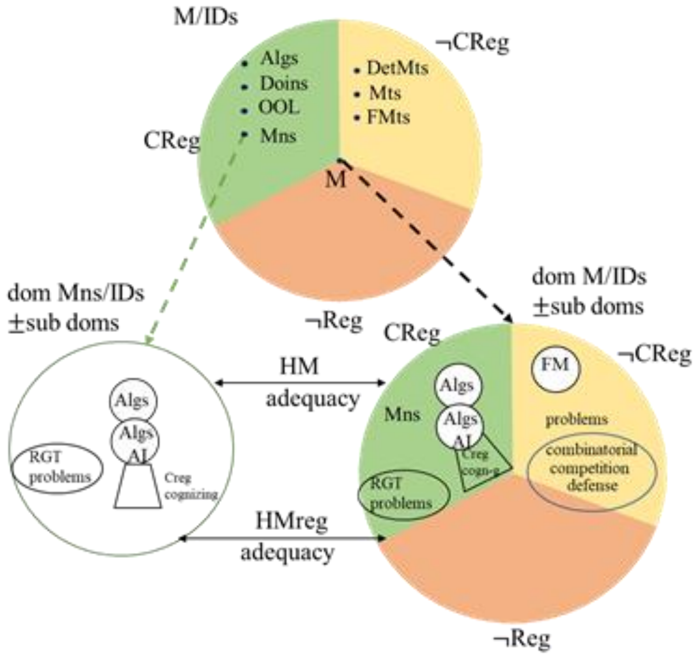


Fig. 15. Disposition of IDs of mss M. Mts = Methods, Mns = Mentals, DetMts = Deterministic Methods, FMts = Fuzzy Methods, The Hypothsis on adequacy of mentals to CRegM (HMreg): *Mentals are adequate models of regularized mss* CRegM, The Hypothsis on adequacy of mentals to mss M (HM): *Mentals are adequate models of mss M*

6. Advancing in Mental Modeling

1. Let's recall some models to focus on modeling mss by mentals to enrich the already known models.

The branch of theory of algorithms, synthesis of algorithms, where assuming apriori certain classifiers of mdoers are already given, the algorithms of synthesis of equal constructive versions of these mdoers are developed.

In deductive modes of synthesis these mdoers can include certain axioms and logical statements or can be determined recursively [31]. In the inductive modes, including machine learning, these mdoers can be represented by samples of their domains or their representations, performances of mdoers, etc. [39].

Certain mss provide methods of transmission, teaching of mss inside the communities C as well as methods for acquisition of these mss.

Although commonality of thesauruses of members of C let them avoid specification of those methods, it becomes unenviable in transmitting and acquiring human mss by computers.

In contrast with machine learning where teachers are forced to provide computers the representations of mss step by step, by portions, teachers can do that holistically and completely when they teach them. Questions arise to the abilities of computers in accepting, disposing and properly processing those mss.

Prospective answers to the above questions for RGT problems are presented in [33 - 36], [39 - 43].

In what follows a way how mental ability to communications can be regularized is discussed to address then their constructive adequate modeling.

2. At the time various mss have proper algorithmic or OO models partly mentioned in the Introduction.

Following the ideas of functional modeling, we are examining mentals, by consistency of their performances with targeted mss, first of all the cognitive ones.

Namely, providing mentals with, say, their OO dimension, for target mss m to examine the expectations, or the hypothesis H, so that these mentals are adequately modeling m [49].

So far, we have advanced in examining H, particularly, for mss of inductive learning [36, 40 - 42], utilization of realities by acquisition or by strategies [38, 39], matching with classifiers [44 - 46] and communications [47] as well as in examining an ability of mentals adequately represent compound classifiers of experts [44 - 46].

To enrich the already known models in [60] we analyze how mental abilities to formation of strategies and classifiers, to communications and acquisition of thesauri can be regularized to question later their constructive adequate modeling.

3. Targeting Further Modeling.

3.1. Believing that *motivation, will, imagination, intuition, cognitive* and *meta cognitive self- awareness* are among the ongoing targets in the modeling of cognition, as well as *regularizing,* the *modeling* itself, it has to be questioned whether the entire dimensions of personality of humans they classify should be necessarily regularized in modeling cognition?

3.2. Targeting the further dimensions of mental modeling, it has to be questioned, following Thomas Sheridan [53], whether these dimensions have *denotative* or *connotative* classifiers as it was questionedin in his book: *What Is God? Can Religion Be Modeled?* as follows:

"…This is a study of the concept of God, not from the perspective of any religious tradition, but rather as belief in a supernatural entity as has prevailed through the ages.

The book's unique perspective is to ask what can be modeled in a denotative language (much as modeling in science, medicine and modern professions) in contrast to a connotative language (e.g., myth, metaphor, art and music). It adopts the assumption of model-based reality, as currently prevalent in physics and some branches of philosophy. This criterion means that an entity can be called real for public discourse purposes only to the extent that a credible model can be made of what the entity is or how it works (as contrasted to the private reality of thoughts, perceptions or dreams.)"

3.3. Thus, for constructive and adequate modeling of cognizing it is first necessary to question, whether the target classifiers are denotative, then, follow to the next question whether they are unavoidable in the modeling of cognition.

Particularly, it is worth making sure that consciousness and self-awareness are denotative to question then their adequate constructive modeling.

Note also that focusing on constructive cognition and creativity allows us to extract simultaneously analogous denotative aspects in classifiers of God, followed by the prospects of modeling the God as it is discussed later in the work.

7. Questioning the Adequacy of Mentals

1. Justification of mentals as adequate models of mss can be done by analogy with justification of algorithms as adequate models of computability by Church.

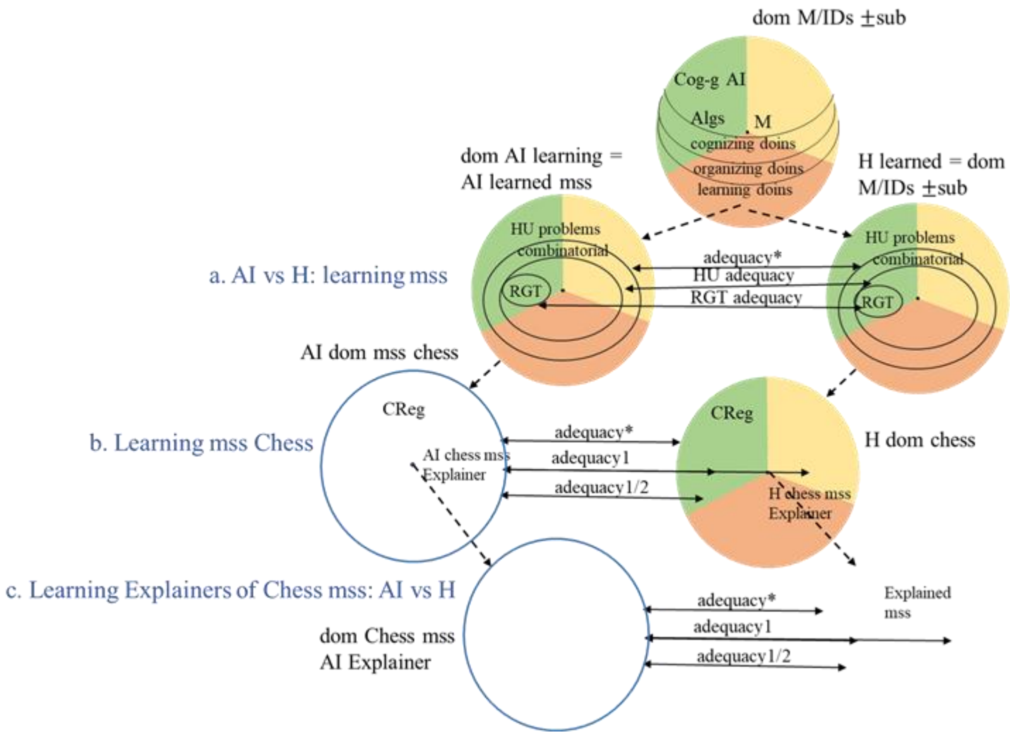


Fig. 16. Adequacy by Church Interpreted for AI Learners of RGT Chess mss: Chess mss, Explainers, Interpreters. a. adequacy*: any mss AI and H learn equally, HU: mss of any HU problem AI and H learn equally, RGT adequacy = adequacy* but only for RGT problems, b. adequacy*: any Chess AI and H learn equally, adequacy1: CReg+¬Creg Chess mss AI and H learn equally, Adequacy1/2: only CReg Chess mss AI and H learn equally, c. adequacy*: any Chess AI and H explain equally, adequacy1: CReg+¬CReg Chess mss AI and H explain equally, Adequacy1/2: only CReg Chess mss AI and H explain equally.

Namely, the adequacy of mentals for several mss should be proven, then, a hypothesis h is declaredon adequacy of mentals to any mss that are examined empirically for mss until h is refitted by some mss or another not equal to mentalsalternative models of mss will be discovered.

1.1. Ideally, this justification means that for theoriginal problem of being of Humans in the Universe (HU) for systemic classifier sClm of any mss m of any x@C solving HU it is possible to provide mentals m’ with classifier sClm’ equal to sClm.

Realistically, since the adequacy of mentals can be examined only for finite number of mss only it is worth examining, first of all, for certain key mss.

As such key mss we select meta mss, i.e., those doing over mss, then ones acknowledged by psychologists and psychotherapists as a nucleus for identifying the norms of being healthy humans.

2. The next barrier in justifying the adequacy of mentals is the incredibility of HU problem in examining the equality of mentals to target mss.

Ideally, to prove adequacy of mentals m’ for target mss m, the we should confirm equality of m and m’ for any type of their relevant processing for any tasks of HU problem, which is unrealistic.

2.1. To overcome this barier, we follow the views that the HU problem can be approximated by game models [52]. Then, analogously with [27] we argue that combinatorial games with known hierarchies of utilities and solutions in spaces of possible strategies of game trees can represent the HU problem with a proper adequacy.

Thus, we are narrow the HU to the Reproducible Game Trees (RGT) problems, design Solvers processing mentals to estimate the adequacy of mentals for mss by estimating On the Job Performances (OJP) of RGT Solvers for the target mental doings.

The approach the estimation of adequacy of models by the effectiveness of their OJP will be detailed in the consequent part of this chapter.

3. By analogy with the justification of the adequacy of algorithms, an argument of adequacy of mentals for mss is if other models of mss are equally reduced to mentals.

Mentals are models of mental doings within of the framework of the functional paradigm, while currently there are models of the same doings within of the framework of the connectivity one, for example, as artificial neuron nets (ANN).

So, a dimension of justification of the adequacy of mentals to mss could bethe proof of the ability of transition from any mentals model of mss to the equal connectivity ones, for example ANN, and vice versa.

In other words, this proof will provide a fundamental argument that both of those types of models, in fact, catch the essence of mental doings.

In what follows we provide certain premises consisting of the above paradigms, while in the consequent part we accomplish them with more arguments.

Summary of Chapter2/1

1. We continue studying ontological, constructive and systemic models of mental systems, *mss,* comparable with expressiveness to algorithms and natural languages.

First, we present the modified specification of the models of mss, *mentals*, introduced in [47], refine systemic classifiers and constructive regularization of classifiers.

Then, question the ways to prove that mentals can be adequate constructive models of mss to provide arguments of their adequacy for explaining, understanding and human-computer interactions, as well as convincing to follow the ideas of inventors of algorithms in adequate modeling of mental behavior.

To consist functional and connectivity mental models, we provide certain evidence that mentals can be reduced to systems of classifiers, which let us concentrate on the organization of neurons to link the mental doings of neuron nets with mentals.

2. To avoid the symbolic jungles, the basics of this Part 1 state, particularly, as follows:

We can operate with realities, i.e. everything that causes imprints in us, while, in fact, we operate consciously or not with realities that can classify.

Utilities are realities that support our goals.

Mss represent realities, and, in particular, utilities, but have varying effectiveness with respect to goals and are processed to support utilization and gaining benefits from utilities.

All classified realities, in principle, can model each other.

For example, realities themselves provide models of the Universe, utilities – the models of realities that support our goals and mental systems are, in particular, the models of utilities.

Mss classification is effective with respect to our goals to the extent, to which they regularly provide appropriate utilities and constructively and adequately model utilities.

Natural languages, parametric, statistical and other types of algorithms, programming languages, logical calculus, are systemically modeling mss in a wide range of constructiveness and adequacy.

3. The statements we convince in Chapter 2/1can be summarized as follows:

Stt1. *Acknowledging that algorithms constructively regularize and model deterministic methods and OOL are constructively regularized and strongly extend the expressiveness ofalgorithms,* ***mentals*** *are constructively regularized and strongly extend the expressiveness of OOL.*

Stt2.*Mentals can be reduced to systems of do classifiers*

4. Finally, let us remind some open questions so far:

Can we provide relationships universal for certain classes of languages or prove that this is impossible?

Can relationships be represented non-numerically, particularly, by neuron nets?

What patterns of mental personality of humans including their emotions, motivations and social relationships can be, in principle, represented by adequate algorithmic models?

Are human personality patterns, including entire ranges of human emotions, motivations and social relationships inevitable in cognitive modeling and if so, can they be equally represented by mentals?

Can we advance in Human-Computer Communications to the extent acceptable, at least, for chess players recalling that chess languages are natural ones like drops of oceans water ?

Can we organize neurons in the nets having mental doings equal to those of mentals?

The following parts will present progress on some of these issues.

Chapter 3/1: Examining the Adequacy of Mental Models

Preface

By questioning the adequacy of mentals analogously with that by Church for algorithms, we make obvious the enhanced expressiveness of mentals, presenting our experience in straightforward specification by mentals of the mss of explanations, understanding, languages and communications.

Then we argue the consistency of mentals with connectivity neuron net models, which adds confidences that mentals are comprehensive and adequate mental models.

Finally, we focus on the methodology for examining the adequacy of cognizers and their constituents based on the evaluation of their on the job performances (OJP) in solving chess, marketing competitions, intrusion protection and defense problems of RGT combinatorial class adequately modeled by Reproducible Game Trees.

1. Regularizing Explanation and Understanding

1. Mss explanations and their understanding are inseparable constituents of communication in communities. Communities C are unions of people aroused to enhance the effectiveness of solving of communal problems of C members through coordination and cooperation of their efforts.

For these purposes, people communicate explaining to each other their goals and plans to attain these goals, assuming that addressees understand explanations properly, i.e., that original mss and those activated in addressees are equal to each other.

The premises for proper understanding of each other in C are the commonalities of their roots and goals evolutionary originated from the roots, as well as the mss that people acquire from C to gain memberships of communities C, while at the same time acquiring them together with *communicatives* (*cms*) of these mss represented by IDs of mss and, in addition, by samples of indoms of mss if they are ***do*** classifiers.

In general, *explanations of mss of members of C* are, we assume, representations by, say, resolution, projection or embodiment of cms of those mss aimed to recall, activate, ideally, equal mss of members of C to cooperate with them in solving common problems. And members of C understand explanations of mss m of members of C if, ideally, they are activated equall as m mss m'.

2. Focusing cms on the type of IDs and assuming that mentals are modeling mss adequately, the explaining/understanding of mentals can be specified as follows.

Namely, certain algorithms, *mental explainers* correspond to the target mentals m certain expressions mexp unanimously representing m while to understand mexp certain algorithms, mental interpreters of members of C, correspond to mexp equal to m mss m'.

Recalling, that mentals can be interpreted as subnets of colored oriented nets or graphs, and recalling that these graphs can be isomorphically represented by their matrices of incidents [32], we can state that, ideally, certain explainers correspond to the mentals m in certain expressions consisting of IDs of these mentals, particularly, in the form of their matrices of incidents or equal to matrices of other expressions of m, while certain interpreters correspond to these expressions of certain mentals m' equal to m.

3. The above explanation /understanding refers to the ideal ontological mss of communities, say, mss of Math or programming languages.

Apparently, real communications only approximate the ideal ones.

3.1. Referring to explanations of mentals m using matrices of incidences we emphasize, first of all, the existence of means of transition from m to the equal m'. In languages, these matrices are presented by a range of realities equal to them, for example, by triples b rels b` resolving mentals by rels and incidental to rels nodes b, b' which, in fact, can be interpreted as explanations of mss, say, in English by clauses or their compositions.

3.2. Explanations by resolutions can vary in length and details depend on the addressee of explanations and can provide additional values, for example, by chaining the clauses consistent with cause-effect or logical inference rules.

Recall also that cms along with IDs, say texts in English, can also be inrealities.

3.3. Resuming the aforementioned, we state that algorithms can adequately specify the explaination and understanding of adequate wrt mss mentals for a range of modes used in human communications.

4. Let now outline the examination of the statement on the adequacy of explanation algorithms for chess classifiers of Solvers [43-46,50].

Recall, that these classifiers are realized in Java and, apparently, can be represented by graphs of IDs of their abstract classes, which using Harary's algorithm can be represented by the equal matrices of incidents, finally transformed into clauses with *attributing, parenting and done by* (HBD) relationships.

For example, the steps of explaining the classifier Lines could be outlined as follows:

- address the decomposition of Lines

-starting from terminals represent the doers/algorithms of Lines in the Markov's way

-identify all the used relations based on interpretations of numeric ones and assign them the corresponding IDs

-transit from the Lines classifier to its representation by the graph of IDs

-using the theorem on the equality of graphs-matrix of incidents (see Harary), transit from the graph of IDs of Linesto its matrix of incidents, i.e., get a complete explanation of Lines by their IDs

- transit from IDs to the corresponding names in English comprising a comprehensive detailed explanation of Lines in English

-apply proper heuristics to compress comprehensive texts into those understandable tot target learners.

5. Finally, let us address some peculiarities, handicaps, disadvantages of explaining.

5.1. Let us note that the **scales of explanations of mss** should necessarily reflect their contentiousness, as well as the following peculiarities:

TBC-vulnerability of explanations to any type of imperfectness of classifiers involved in explanation of mss, since they are composed of IDs or samples of indoms of these classifiers, i.e., of communicatives (cms) of mss

TBC-explanations by decompositions of classifiers are not appropriate for terminal ones, since they are learned only by personal experience and can only be explained by providing the samples of their indoms.

Logical or cause-effect explanations are more exceptions than regularities.

5.2. Common explanations can suffer from malicious circles and be antinomies.
They can come to dead ends caused, particularly, by limited expertise involved in the formation of classifiers. For example, we face dead ends when we try to imagine the “end, final realities” applying the “apparent” rule that any reality in space is contained in some bigger ones, when we try to apply Euclidean geometry or Neuton physics to the entire Universe.

5.3. Then, mss are holistic and, in general, need a large number of cms to be represented.

Thus, explanations of mss m with IDs **a** of member x of the community C can be ideally understandable for y@C, when the meanings of **a** can also be represented holistically, be provided in parallel, like we have for sculptures and theaters, involv max cms of m, at the same time being so much differentiated and detailed as in language texts

5.4. Recalling that the vast majority of mss are learned by their acquisition from communities, ideally, thesauri of members formed by the same communities should be highly equal, have common IDs of mss and corresponded to cms common units of languages of those communities, thus, have nearly equal meanings for language units.

But for various reasons thesauri approximating the ideal ones, appear only for narrow straites of communities, for example, for professionals in certain areas.

Thus, x@C for a successful explanation , at least, have to be aware of the content and scope of thesauri of y@C to chose appropriate cms and sequence them properly.

5.5. Resuming we can state that, in general, an explanation is a choice between contrary requirements in, say, wholeness, differentiations, comprehensiveness and conciseness, so a good explaination involves more arts than sciences.

2. Regularizing Human Computer Communications (HCC)

1. People of communities C communicate in languages L of C, and premises for successful communications include, we assume, the following:

- commonality of thesauri Th of mss in C, i.e., for the members of C
- commonality of totalities of cms of mss of Th, corpuses crpL of L in C
- rules, syntaxes of correspondencesto mss ofmTh certain cms-expressions of crpL

-mdoers/algorithms, mental explainers, corresponding cms-expressions to mss

-rules, semantics, corresponding mss to cms-expressions

-mdoers/algorithms, mental interpreters, corresponding mss to cms-expressions.

2. HCC communications in C to be comparable with those between humans, apparently, have to satisfy the above premises.

At present, however, HCC are based on programming languages (PL) that only partly meet these premises.

Namely, computers by algorithms, abstract classes and their packages are adequately modeling only mdoers of thesauruses of C.

Programmers assembling those mdoers or their compositions can explain to computers the plans for processing mdoers that are unanimously understood and realized by computers.

In addition, HCC can be consistent with C communications wrt systems equally representable by numeric ones as it was illustrated for chess, and therefore [40], allowing to do the same for the adjacent to chess RGT problems.

In the frame of these numerically equal systems, say, the RGT class, inconsistency in HCC wrt pure human communications are inevitable, especially for cms representing emotions, motivations, social rels, etc.

2.1. Resuming the foregoing, we assume, that mentals based HCC can approximate C communications to the extent to which constructive Th thesauri of computers are equal to mTh of C.

3. Let's address now the requirements to HCC interpreters of languages L into languages L' in C.

People interpret expressions exsL in L into exsL' in L' of the given languages L and L' of C by human interpreters Intrs.

Intrs understand exsL, i.e., activate mss m* equal to those m expressed by exsL, then select, pick out exsL' activating mss m', ideally, equal to m*, to, finally output exsL' expressing m', equal to m.

Apparently, HCC-based interpreters, cIntrs, should ideally perform equally with respect to Intrs.

3.1. Now let's see to what extent one of the popular UNL cInts [20, 21] meets the above requirements.

UNL originates from the assumptions that thesauri ThC of communities C communicating in languages L and L' converge to communal thesauri Th due to the consequences of globalization.

As a result, the communities C and C' are assumed to do with the same realities and mss representing realities while named by different cms.

Particularly, basic rels of L and L' become equal to some very limited bRels (UNL indicates about 44 of them) that acceptably approximate rels of L and L'.

Following these assumptions corresponding to each other mss of C,C' can get the same codes, Universal Words (UW), while any exsL by human Intrs of C using UW and bRels can be represented by equal to excL certain UNL expressions exs* that, ideally, would be decoded into equal to exsL expressions exsL' of L'.

3.2. Unfortunately, any properly acceptable UNL based HCC interpreters remain unknown during over 20 years of attempts of realization of UNL ideas in about 20 languages /countries. Even it is not proved yet that exsL coded by exs* in UNL by human Ints after some algorithmic decoding into expressions of the same L will be equal to the original exsL.

3.3. Thus, it is natural to revise UNL assumptions.
First, the most doubtful of them, seemingly, concerns the universal basic relsbRels of UNL. Particularly, several researchers have been proving incompleteness of bRels.

Second, the units of languages, particularly, UW only partly represent mss while for proper expression of certain mss m the interpreters can be forced to look iteratively for compositions of units of corpora of languages that will acceptably activate m.

3. Consisting of Functional and Connectivity Mental Models

1. The adequacy of mss models is questioned functionally and connectively.
Functional questioning examines the equality of performances of mss and models of those mss abstracting from physical nature and the origin of the models.

In contrast, in connectivity modeling, it is required that the units of the mss models be adequate models of the units of nerves systems, neurons, and looking for adequacy of mss with the nets of those model neurons.

Following the ideas of functional modeling, we examine our models of mss, mentals, by consistency of their performances with targeted mss, first of all cognitive ones.

Namely, provide mentals for target mss m to examine the hypothesis that these mentals are adequately modeling m.

2. These days, the adequacy of NN connectivity models has been intensively examined for artificial neuron nets, ANN, which are consistent with different physiological views on NN classifiers.

Consequently, the *hypothesis eANN?Th* arises on the equality of ANN and nets of mentals, thesauri Th or in other words, whether for any mental doings m represented by Th can be provided ANN equally representing m and vice versa.

Our expectations for the positive answer to eANN?Th are supported by the following premises.

3. Doers of Th, doins, and doers of ANN, neurons, are both doers over IDs of nominals over given the sensors/effectors/controllers.

3.1. ANN are NN models in which the constituent neurons and rels between them are classifiers either types of sensors, classifiers of IDs of nominals or n tuples of IDs of nominals, i.e., rels.

3.2. Mentals, in turn, are systems of classifiers and corresponders (cors) incrementally layer by layer formed into Th where only the kernel of cors, i.e., regularities (regs)/ rules, differ from ***do*** classifiers by the provided definitions.

4. Therefore, any mss m modeled by ANN can equally be modeled by Th, while for a positive resolution of eANN?Th, at least, evidence should be provided that rules can be represented in frame of ANN.

Premises for positive answer to this question are based on recalling that in logics functions can be interpreted as predicates, i.e., as classifiers, and vice versa.

In parallel, cause-effect reflexes according to Pavlov can be interpreted both as classifiers and as regs.

Therefore, both ANN and mentals, in principle, have the base of equal modeling.

4.1. Indeed.

4.1.1. Markov algorithms, apparently, are systems of rules, regs of the types of "realities y follows realities x", "y is a function of x", "x cause y", etc.

Recalling that functions and predicates in logicscan be equally represented by each other we can state that the above interpretations of Markov rules can be represented as corresponding time, case-effect, dependency rels, thus concluding that mentals, in general, can be represented as systems of do classifiers.

4.1.2. Then, Pavlov,' unconditioned reflex "salvation into food", for example, accompanied by lighting, causes a conditioned reflex of the same secretion to lighting without food.

In other words, activation of neuron classifiers Cl of food accompanied by the activation of classifiers Cl' of lighting generates positive synaptic links of a causal type between these classifiers.

As a result, a time type **casual rels** *"lighting* causes *food"* is formed between IDs of Cl and Cl'.

And lighting will activate that rels arising expectations for food, which, apparently, can be eliminated in case of regular failure.

Despite the fact that rels *"lighting* causes *food"* is a ***do*** classifier it can be interpreted as regs as well.

Indeed, the appearance of *lighting* indicates that *food* can be expected.

Therefore, in NN reasoning the rels "*lighting* causes *food* "can be processed as regs, rulesof the type "if *lighting* , then *food*".

5. Concluding, we can state that since mentals are reducible to and ANN are systems of do classifiers, they, in principle, can provide equal mental models.

In addition, it is worth notinge that mentals enrich algorithms to represent not only the classifiers having numeric analogues but also similar to NN and ANN ones provided by sensors.

Next, we have to question whether equal to each other ANN and mentals wrt mental modeling can be, in fact, constructed?

To resolve that question, it is worth to answer to the following induced questions.

5.1. Assuming that ANN can equally represent any Th classifier and cors, then can they represent Th algorithms, abstract classes and abstracts explicitly or not, and if explicitly then how should ANN be organized?

In other words it is worth questioning the assemblies of neurons providing a diversity of mental doings equal to those performed by mentals at the time.

5.2. How can human-computer communications, including explanations and understanding of mss, prognostication of utilities and search strategies having proper cesTh models, be interpreted in ANN?

5.3. NN are systems of neurons where rels between neurons of different layers are represented by synaptic links.

Recalling that the number of synaptic links in NN of any human concurs with the number of particles in the Universe, it is worth to assuming that NN provide, in fact, an example of a constructive implementation of rels, particularly of the type of cors.

Thus, the question arises whether the rels between IDs of nominals can be implemented in mentals of Th analogously to those in NN?

5.4. Eventually, how can mss identified by psychologists be equally interpreted both in ANN and cesTh?

The answers to some of the above questions we provide in the consequent part on the modeling of mental classifying.

5.5. The following questioning is urgent due to a burst of interest to ANN.

5.5.1. Given classifiers f presented by Boolean tables, we can transit both to their DNF and ANN representations.
DNF line is transparent while NN one remines vague.
The idea is that deep learning of f by NN can be interpreted as construction of reduced DNF.

5.5.2. Thus, prove that ANN deep learning of Boolean functions f given by truth tables can be interpreted as an iterative method of representation f by reduced DNF, rdnf, (abbreviated DNF).

5.5.3. The following proving premises are available:

- pdnf in n-cube represents the disjunction of districts covering positives of f
- rdnf can be represented by 3 ANN as follows
  - each 2d Neuron corresponds to a certain conjunction k of rdnf f with input variables coinciding with those of k, and is activated when k is positive
  - the weights of outputs of 2d neurons and the threshold of the 3D one n3 let activate it when, at least, one of n2 is active
- deep learning can be interpreted as an iterative approach to the 2d neurons representing conjunctions of rdnf representation of f.

4. Adequacy of Cognizers by On the Job Performance Tournaments

1. Human cognition, in general, consequences compositions of varieties of mss, primarily, cognitive doers where contributions of particular mss can only be identified approximately.

1.1. Classifiers, their systems are far for being perfect wrt to the roots and other utilities since the means their formation are imperfect, therefore, in principle, there can not be unanimous criteria for their standalone, detached effectiveness.

Thus, effectiveness of the mss, as a rule, make sense to evaluate by the integrative outcome of all mss involved in performance wrt utilities, i.e., by their On the Job Performances (OJP).

1.2. The value of evaluation of OJP of solvers of certain problems wrt qualities of solutions can be strengthened when problems need to be solved in competitions where some counter partners resist to the solution, in other words, when solutions are attained in tournaments.

1.3. In addition, tournaments can be organized in the way letting to some extent evaluate contributions of constituents of solvers when the counter partners are designed to be equal except the questioned target constituents of the solvers.

2. The OJP methodology was evolving in parallel with its approbation in RGT problems started with chess and followed by, particularly, the problems of marketing, intrusion protection and defense.

For example, the theorems on OJP were initially proved for chess-like games, followed by the design of RGT/chess Solvers and examining the adequacy of certain constituents of chess cognition.

Then the methodology was extended to management and marketing problems followed by intrusion protection and defense ones.

Each of these concretizations in the development of RGT Solvers and OJP methodology was based on the deep embracement into peculiarities and details of particular problems because their unification would definitely cause the poorer quality of modeling.

At present, we focus on adequate modeling of nuclear constituents of cognizers within of the framework of RGT problems and the development of the shell of RGT Solvers.

2.1. To present RGT problems/ Solvers, the base of OJP methodology and their positive impacts in examing the adequacy of models of cognizing, we first introduce the conceptual frameworks of OJP and RGT problems/Solvers in Paragraph 4 of Chapter 3/1 below, then, by basic publications reproduced in *Appendicies 1-9* , we detail and deepen these frameworks along with their successful applications.

2.2. In Paragraph 5 of Chapter 3/1, first of all, we refine the scaling of the effectiveness of models by their On the Job Performances.

Then, we argue the reasons, we assume, that the estimates of examination of the adequacy of RGT cognizers can be transferred to the estimation of the adequacy of models of Human cognizers in the Universe (HU).

The introduction to RGT Solvers and some of their applications in Paragraph 6 of Chapter 3/1 reproduce our publication [43] at NATO ASI 2011*: Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems, Salamanca, Spain, September 19-30, 2011.*

2.3. In Paragraph 7 we provide the list of Contents of Appendicies presenting basic results of researchers of the direction "Cognitive Algorithms and Models" in combinatorics, games, learning, RGT Solvers, OJP and their background at the Academy of Sciences of the Republic of Armenia (IIAP) since 1973 that we have refered at the 1st Chapter.

5. Scaling the Effectiveness of Models by Their On the Job Performances

1. So far, we have introduced classifiers of the types of:

-sensors xSns,

-genomic goals xGn, including evolutionary approved goals, or classifiers of utilities wrt roots and chains of consequent classifiers, goals, wrt the preceding ones,

-genomic classifiers *xClgn* uniting Sns and sGn,

-goals *Gl* corresponding to classifiers *xClt learned* in the life time (lt) of *x* and uniting *do* classifiers *xdoCl* by themselves and systemic classifiers xsCl induced by ad hoc *do* classifiers of the constituents of mentals of the thesauri *xTh*,

-goals *xG* uniting *xGn* and goals *Gl* and, finally,

-attributes *xAtr = x(ClgnU Clt) = x(GnUGl)* comprising all *do* classifiers of *x* available at time *t*.

1.1. Then, for unified addressing to any classifiers of any member of community C, we assume for a convenience, that classifiers of some ideal members z of communities C are equal to those that integrate the classifiers of all members of C , namely, with Sns, Gn, Th, Cll, G, goals Gl, Atr, store of prints SP* and Universe U (see in Chapter 4/1 for more details) .

Then, let us remind that these classifiers are changed over time, so they need to be analyzed depended on time, as well as that all classifiers can be interpreted as goals to gain or avoid certain realities why humans and more, cellulars and negs are *goal oriented* realities.

1.2. Bundles of IDs of activated by some realities *r* attributes rAtr at time t+ Δ are *t prints* of z.

Realities r represented by activated t prints and attributes rAtr are the *situations* of z at t, or *t situations* where z can be omitted since, in general, realities exist independly of z, and z by assumptions classify equal to united classification of all members of community.

The diversity of all situations Sit for z is apparently due to the diversity of realities of the Universe, as well as the variety of internal and external circumstances and goals pursued by z why the same realities r at t can be considered as different t situations.

1.2.1. Given situations P@Sit and goals g from Gl algorithms pursuing g from P are *gP strategies* or *gP-str.*

1.2.2. For g from Gl, let *gScs* be the scales of numeric degrees of equality of realities r wrt g, or scales of the proximity of r to g.

By analogy with RGT strategies, for example chess ones [43], gP-str are assumed to be finite with terminal situations P' scaled at certain scales *gScs,* while the chains *d* of doings of gP-str linking P to P' are *PP" trajectories* of gP-str with the *effectiveness* equal to the ones of P' wrt gScs.

Maps of effectiveness of gP-str S are sets of effectiveness for all trajectories of S and the *scales of effectiveness of S* are the correspondences of these maps in certain numeric scales strScs.

1.3. Let's refine *performances* zgP *of z@C from P@Sit for the goals* g, or *on the job gP performances of z* (*zgP-OJP*) as gP strategies S and the *effectiveness of zgP OJP wrt to scales strScs* by the effectiveness of S wrt strScs.

Analogously, let's refine *performances of z in U* or *zOJP* as the union of performances zgP by all g and P, while the *effectiveness of z in U* or *zeOJP* as the scores of the unions of effectiveness of gP OJP wrt numeric scales StrScs ordering in certain ways the effectiveness of unions of strategies in U.

For example, for human chess strategies common scales of effectiveness strScs can be those based on the scores in absolute tournaments of all chess strategies of human z from the standard Po positions, while the effectiveness of chess players in chess can be assessed by suitable integration of effectiveness of z wrt scales strSc from any legal chess positions in certain integrative scales StrScs.

1.3.1. Apparently, the above definitions are based on non constructive requirements, and even if some realities, for example, strategies, meet them, the scores of their effectiveness are essentially dependent on a variety of criteria by which the scales are chosen [40].

Reserving a further analysis of criteria of scales strScs and StrScs, let us assume that they have been chosen and address the constructive approximation of the above requirements.

1.4. We assume, that the effectiveness of humans z in U wrt , say strScs, can be well approximated by the provision of "key" sub problems subU of the Universe U properly representing the essential aspects and dimensions of U with the corresponding goals of z in subU, followed by regularization of subU and estimation of effectiveness of z in U wrt strScs by solving not all problems of U but only those of subU.

If subU sufficiently represent U, the zeOJP scores wrt strScs for the entire U can be estimated by the zeOJP scores wrt strScs considered only within the framework of subU.

And *subU adequately modeling of the problems of U* if the estimates of zeOJP in solving subU are acceptable approximate zeOJP in the entire U.

1.4.1. Interpreting the above approach for subU=RGT problems, then reducing them to the kernel ones [43], for example chess, Elo ratings to some extend will estimate OJP of human chess players, at the same time they can enhance the proximity if estimated by solving any RGT problem.

1.5. Continuing enhancement of constructiveness of zeOJP, in addition to adequate modeling of problems of U by subU, the models of z solving subU, i.e., subU Solvers, can be designed so that focusing certain target doings of z can allow by local resources to estimate zeOJP properly to a certain extent consistent with those for humans.

Namely, assuming that subU adequately modeling the problems of U and focus the mental doings zM of z, certain constructively regularized models zM* of zM within of the famework of subU Solvers can be designed.

Then, if these Solvers meet the requirements consisting of their doings with those of humans, their effectiveness can be estimated by human scales Scs.

And *constructions zM* adequately model the mental doings zM of z* if estimates of subU Solvers wrt scales Scs consistent with human ones are acceptable approximate zeOJP in solving the problems of subU.

1.5.1. Concluding, let,s state that, in principle, OJP adequacy of mentals wrt to mss can be acceptably estimated by the performances of RGT Solvers if , particularly, the scales of effectiveness of Solvers and humans are consistent and doings of Solvers are governed by the mentals.

In fact, estimation of OJP adequacy of mentals wrt mss along with accomplishing the above requirements needs experimenting with RGT Solvers in modeling target mental doings followed by their trustful comparison with the performances of humans.

6. Arguing Approximation of Estimates of OJP/ Human Cognizers in the Universe (HU) by Estimates of OJP/ RGT Cognizers

1. The next barrier in justificaying the adequacy of mentals is the incredibile problem of HU in examining the equality of mentals to target mss.

Ideally, for prove the adequacy of mentals m’ for target mss m we had to confirm the equality of m and m’ for any type of their relevant processing for any tasks of BU problem, which is unrealistic.

To overcome the barrier, we follow the views that BU problem can be approximated by game models [52]. Then, analogous to [27], we find that combinatorial games with known hierarchies of utilities and solutions in spaces of possible strategies in game trees can with proper adequacy represent the HU problem.

Thus, we narrow HU to the Solvers of Reproducible Game Trees (RGT) problems, a class of combinatorial problems with only a few following requirements to belong to:

- there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions at (c) specified moments of time and (d) specified types of situations,
- there are identified benefits for each of the actors,
- situations in which the actors act and in which are transformed after the actions can be specified by certain rules, regularities.

2. Arguments of proper adequacy of RGT Solvers to HU include the following ones.

First, we find out that games with known hierarchies of utilities and solutions in spaces of possible strategies in game trees can properly represent the HU problem.

Then, RGTs represent combinatorial environments that, generally, cover enormous, yet unsolved problems [27], in contrast to those that are well represented by classical continuity and parameterizations-based mathematics.

RGT is a spacious class of unsolved problems including many security and competition problems. Specifically, these are network Intrusion Protection (IP), Management in oligopoly competitions and Chess-like combinatorial problems as well as many other security problems such as Computer Terrorism Countermeasures, Disaster Forecast and Prevention, Information Security, etc. [43].

3. Along with acceptable adequacy, RGT software currently provides a preferable research environment letting look for proper scientific assertions. These preferences include the following ones.

a. Urgent and spacious RGT combating and competition problems are reducible to the standard kernel problems K of RGT class and we do focus on chess as K.

b. K- methodology multiplies the achievements for particular problems of the SSRGT class.

c. Distributed development of K-methodology is possible.

d. K-centric methodology enhances the effectiveness of RGT Solvers providing answers to urgent RGT questions including the following ones:

e. measurement of the effectiveness of Solvers,

f. analysis and typification of combating knowledge,

g. construction of knowledge-based Solvers,

h. regular acquisition of RGT expert knowledge and enhancing the effectiveness of Solvers.

k. The validity of K- methodology was proved for certain RGT problems including Chess, Network Intrusion Protection, Navy Defense from Attacks, Management, Marketing etc.

3.1. The shell of RGT Solvers is developed to provide a user-friendly Java environment for managing any RGT problem [40,44-46,50].

7. Effectiveness of Enhancing Knowledge-Based Strategies for RGT Class of Problems

(NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems, Salamanca, Spain, September 19-30, 2011, pp.16.)

ABSTRACT

We study diversified problems with space of possible solutions specified by Reproducible Game Trees and develop a software package, Solver, for a unified search of experts the solutions of specified RGT games.

We tend to justify assertions that urgent combating and competition problems are reducible to the standard kernel problem K while the achievements for K are multiplied for RGT class.

K-centric methodology enhances the effectiveness of RGT Solvers by answering the urgent questions on measuring the effectiveness of solutions, analysis and typifying combating knowledge, construction of knowledge based Solvers, regular acquisition of expert knowledge to enhance the effectiveness of Solvers, etc.

The validity of K- methodology was proved for Chess-like combinatorial games Network Intrusion Protection, Navy Defense from Attacks, Management, Marketing and other problems. The shell of RGT Solvers is developed to provide user friendly Java environment for managing any RGT problem

***Keywords*:** meaning processing, combating and competing games, expert systems, optimal strategies

7.1. Introduction

7.1.1. In the variety of problems we identify a class where space of possible solutions can be ***specified*** by *Reproducible* combinatorial *Game Trees* (RGT) and develop ***unified*** algorithms and software, ***RGT Solver***, to elaborate optimal strategies for any input specified problem of the class.

The RGT is a spacious class of problems with only a few following requirements to belong to:

- there are (a) interactingactors (players, competitors, etc.) performing (b) identified types of actions at the (c) specified moments of time and (d) specified types of situations
- there are identified benefits for each of the actors
- the situations in which the actors act and in which are transformed after the actions can be specified by certain rules, regularities.

Many security and competition problems belong to RGT class. Specifically, these are network Intrusion Protection (IP), Management in Oligopoly Competitions and Chess-like combinatorial problems. Many other security problems such as Computer Terrorism Countermeasures, Disaster Forecast and Prevention, Information Security, etc., have been announced by NATO (http://www.nato.int/science/e/newinitiative.htm).

7.1.1.1. Thanks to tremendous advances in computers, it becomes possible to personalize interactive tutoring for students and test their progress in learning new knowledge.

Particularly, while the knowledge acquisition by children with ordinary abilities follows traditions, personal experiments of parents and well-known standard methodologies for the development of unordinary children with positive and negative declines need to be extremely personalized to each child means.

In [67], the peculiarities and patterns of positive tutoring of autistic children (autism) in knowledge acquisition wich were continued in [69] for application to tutoring in general, were considered. It was argued that tools providing a complete tree of constituents needed to acquire certain communalized knowledge followed by providing recommendations on learning the missed constituents can be supportive for any child.

Since testing can be interpreted as an RGT problem, RGT Solvers can be used as advanced instruments to acquire expert knowledge and elaborate effective strategies of its testing while the adequacy of computer-based models of explanation of new knowledge and its testing to experts have to be substantiated.

7.1.1.2. Unified RGT specification of problems makes it possible to design a unified Solver for the problems of the class.

Solver of the RGT problems is a package aimed to acquire strategic expert knowledge to become a human-comparable in solving hard combinatorial competing and combating problems. In fact, the following three tasks of expert knowledge acquisition can be identified in the process:

- construction of the package of programs *sufficient* to acquire the ***meanings*** of the units of vocabulary (UV) of problems
- construction of procedures for*regular* acquisition of the meanings of UV by the package
-provision of means for *measuring the effectiveness* of solutions of RGT problems.

We formulate the limitations in designing an effective package as follows:

-be able to store typical categories of communalized knowledge as well as the personalized one and depend on them in strategy formation

-be able to test an approximate knowledge-based hypothesis on strategies in the questioned situation by reliable means, for example, using game tree search techniques.

The second task of *acquiring complex expert knowledge* was planned to solve in the following two stages:

- *proving the sufficiency*, i.e., prove that Solver, in principle, can acquire the meanings of expert knowledge of an intensive RGT problem, e.g.for the kernel RGT chess game
- *ensuring regularity,* i.e., develop procedures for regular acquisition of RGT problems and meanings of UV of these problems.

7.1.1.3. Regular improvement of Solver by expert knowledge is studied for chess, where the problems of knowledge representation and consistent inclusion into the programs remain central since the pioneering work by Shannon in 1950.

Players indicate and communicate chess knowledge by units of vocabulary and are able to form corresponding contents. It remains under question whether it is possible to form equal contents by computers.

A comprehensive overview of the state of the art in learning the contents of vocabulary in games is presented in [25]. Models of chess concepts based on the elements of chess contents – chunks, are discussed in [20,23] to answer whether the chunks are common for all players or they are individualized. These problems are analogous to other semantics- related problems, particularly the understanding of instructions by robots, ontology formation, etc.

The approaches to regular inclusion of chess knowledge in the strategy formation process are described in [16-18, 20,25]. All of them try to bring common handbook knowledge to cut the search in the game tree. The frontiers of those approaches can be revealed by understanding the role and proportion of the personalized chess expertise compared with the common, communicable one.

7.1.1.4. We conduct research on knowledge-based strategies at the Institute for Informatics and Automation Problems of the Academy of Sciences of the Republic of Armenia, and these studies have been started since thr foundation of the Institute in 1957 by Sergey Mergelyan.

Early noticeable results were published in [16] by the team of Igor Zaslavski, a bright follower of the Andrey Markov' school, in the Laboratory of "Mathematical Logics".

Since 1973 study of meaning processing has been continued by the branch of Zaslavski's team in our Laboratory of "Cognitive Algorithms and Models". Our knowledge-based study of interaction with computer chess master was initiated by Tigran Petrosyan, world champion in chess (1963 – 1966) after his victory a tournament with Michail Botvinnik, world champion in period of 1948-1963.

7.1.1.5. The aims of the work are

- to outline the main approaches and solutions of the team in effectiveness enhancing the knowledge-based strategies for RGT class of competition and defense problems
- to provide links and references for further study.

We present diversified examples of problems reduced to the RGT class, the software package, Solver, designed to search solutions to any specified RGT game and find human- comparable strategies outline the base of Solver, meaning processing, and conclude with ongoing research expectations.

Eventually, we aim to justify the following **assertions:**

1. Urgent and spacious RGT combating and competition problems are reducible to the standard kernel problem **K** of the class and ***we do focus chess as the K***

2. K- methodology multiplies the achievements for particular problems of RGT class

3. Distributed development of K-methodology is possible

4. K-centric methodology enhances the effectiveness of RGT Solvers providing answers to the urgent RGT questions including the following ones:

4.1. measurement of the effectiveness of Solvers

4.2. analysis and typifying of combating knowledge

4.3. construction of knowledge-based RGT Solvers

4.4. regular acquisition of RGT expert knowledge and enhancing the effectiveness of Solvers

5. The validity of K- methodology was proved for certain RGT problems, including Chess, Network Intrusion Protection, Navy Defense from Attacks, Management, Marketing, etc.

6. The shell of RGT Solvers is developed to provide a user-friendly Java environment for managing any RGT problem.

7.2. RGT Games

(Note, that the ongoing interpretation of RGT games was provided in the previous section of this Chapter).

We do solve *games* of *RGT* class (where the space of solutions are Reproducible Game Trees) with meanings we do specify by *states, situations, actors, actions* of players, *evaluators* of situations and *regularities* of the transformation of situations.

States, actions and *situations* are realities of certain types, presented by words in certain alphabets.

While states address the slowly changed realities, situations unite states with actions to present both static and dynamic realities.

Evaluators are algorithms that calculate utilities the actors gain in input situations (see the examples below).

Regularities are, in fact, algorithms, presented, as a rule, by pairs of words in the alphabets of situations and aimed to determine the transformations of situations.

Regularities (regs,) input situations, search inclusions of words of their left parts in situations and, if found, replace them with the right part words to pass the transformed situations to other regs, or, other wise, to output them.

Algorithms, laws, etc., the regs following transforming the left words of regs into the right ones, get interpretations witthin the frames of currently focused applications.
Players **do** affect situations by actions at time t and transform them by regs in new situations at time t+1 tending to achieve the best utilities over the chains of produced situations.

Starting from the situation So and consecutively applying regs **we** can generate the tree of all situations possible in interactions between players and, therefore, containing all possible policies of players, *strategies*.

Strategies are algorithms of actors that input situations and output their actions to affect situations. Evidently, the the best strategies gain max utilities.

Given criteria of quality of strategies based on evaluation of utilities of situations in performance trees of strategies at time t of run of strategies from So we can generate the game tree with the root at So, order strategies by the expected utilities at t and chose the best of them to apply in So.

7.3. RGT Interpretation of Chess, Mangement, Intrusion Protection Problems

Immense competing and combating problems can have an RGT interpretation since those problems are always interactions and RGT requirements include the most common of them.
We do argue this ***assumption*** by the diversity of our interpretations and applications we do present by Chess, Market Competitions, Intrusion ProtectionandOwnship Air Threats Defense (see Chapter 6) optimal strategy search games.

7.3.1. Chess [22-33]

States: composition of chess pieces on the board.
Situations: states and actions of players.
Actors: white and black players.
Actions: chess piece moves.
Evaluators: check mate.
Regularities: chess rules determining legal moves in positions.

7.3.2. Market Competitions for the Value War game [37-46]

*State s*are determined by the set of parameters of current competition and scenarios of competition, i.e., the competition template formed from the conceptual basis of management theory
Situations are determined by states and actions of competition.
Actors: a company competing against a few others
Actions: are changes in product price and quality.
Evaluators: algorithms calculating for input situations cumulative profit, Return-On-Investment, etc.
Regularities or transformation rules are determined by general micro- macro- economics laws applied to the situations.

7.3.3. Intrusion Protection [51]. Intrusion Protection (IP) is a game between two *actors*, the attacker (A) and the defender (D), with opposite *interests*. The game is described by a set of *situations, actions, evaluators* of situations and a collection of transformation rules, *regularities*, the conversion procedures from one *situation* to another defined as follows .

S system states depend on *system resources*: the processor time, buffers size, number of incoming packages. Different measuring scales, such as seconds, bytes, numbers of incorrect logins or incoming packages, etc., are used to measure different parameters.

Let R = {r} be a non-empty set of resources of the system and Q be the set of parameters of the r resource. Each $r \in R$ is associated with a pair <q; w>, $q \in Q$ and $w \in W$, where W is the set of possible scales of R.

Criterion functions F is a set of functions f with the range of values Z = [0, 1].

The *local state* of system's resources on a non-empty set $R` \subseteq R$, R` = {r1, r2, …, rk}, is called the value $e \in Z$ of the criterion function $f \in F$ on this set: e = f(r1, r2, …, rk).

The local state is called *normal* if e = 0 and *critical* if e = 1, while L is the set of local states. Intuitively, criterion functions measure the "distance" between current states and those that are considered as normal.

The ***evaluators*** in IP problem is a criterion function $g \in F$ on the sets L', $L' \subseteq L$, while the states of the system, or ***states,*** are the values of the evaluators.

A state is called *normal* if s = 0 and *critical* if s = 1, while S is the set of states s.

The main ***interests*** of attackers and defenders are to bring the system into critical states and avoid them, correspondingly, by ***actions,*** changes in resources.

The ***regularities***, or *conversation procedures* [14], of IP are the rules determining the states of the system subject of actions of attackers or defenders.

In an IP game, attackers act from the initial state $s0 \in S$ aimed at change the resources of the system, while defenders reply in turn aimed to prevent disturbances. The terminal nodes of the game correspond to the winning or losing states of the defender.

7.4. RGT Solvers

Figure 17 presents the *flow chart* of the ongoing version *of Solver*.

The basic units of Solver implemented in Java [71-75] are as follows:

- *Personalized Planning and Integrated Testing* (PPIT) program is the central module of Solver and is composed of the following basic units: Knowledge Base (KB), i.e. Store of Meanings, Store of t-prints and Graph of Abstracts, Reducing Hopeless Plans (RHP), Choosing Plans with Max Utility (CPMU) , Generating Moves by a Plan (GMP). Given a questioned position P1 and a store of plans in KB, RHP recommends the CPMU a list L1 of plans promising by some not necessary proved reasons to be analyzed in P1. RHP relies on hierarchies, classifications, etc. of knowledge in KB and provides the most perspective plans for further analysis of P1. If KB is rich and P1 is identified properly, it can provide a ready-to-use portion of knowledge to direct the further game playing process by GMP unit. Otherwise, RHP, realizing a reduced version of CPMU, identifies L1 and passes the control to CPMU.

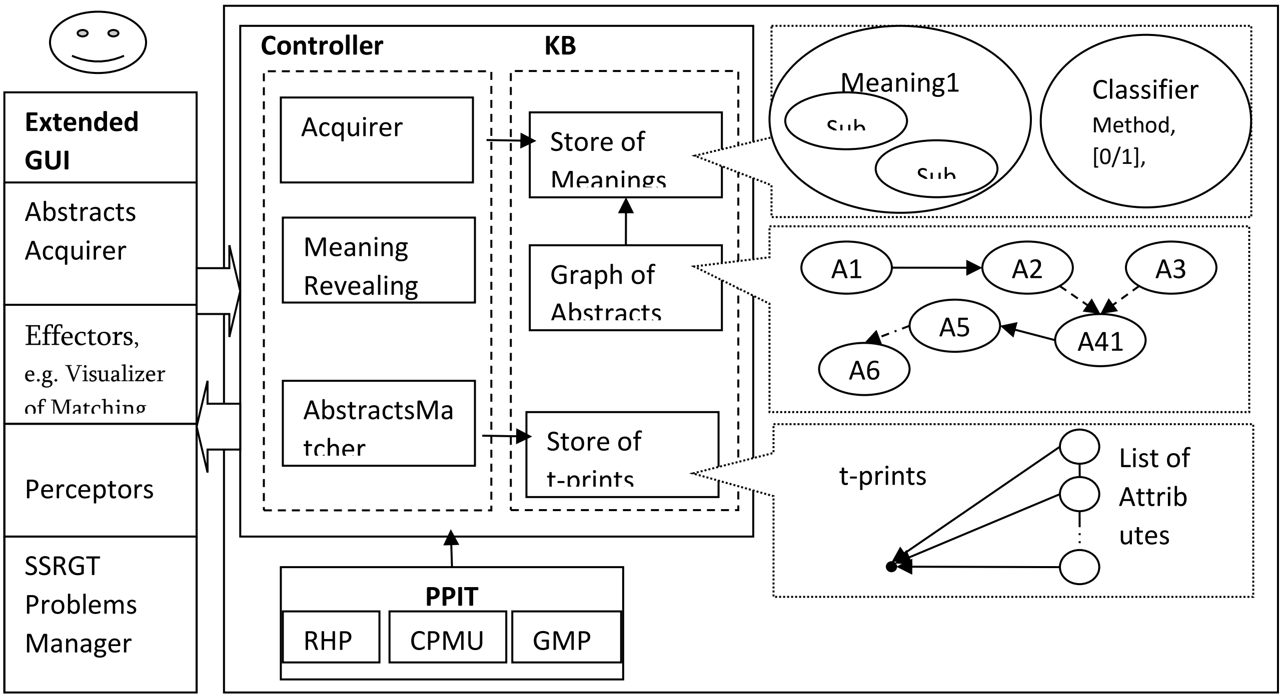


Fig.17. RGT Solver

- *The controller* is responsible for gaining, storing, organizing, activating and processing expert knowledge and meanings. It includes, particularly, constructors: abstractors, grounders, acquirers, assemblers, inferers, taskers, questioniers, modelers, predictors, understanders/interpreters, explainers, communicators, solvers. We will present the basics of meaning processing in the next section.
- *Extended Graphical User Interface* provides a web based graphical interface that allows users of the package to visually configure the package, insert an arbitrary problem from the RGT class and organize the knowledge insertion process in the regular way by using visual instruments including graphical languages.

Sufficiency of Solver for acquiring and effectively involving complex expert knowledge in decisions making was demonstrated by experiments in the acquisition of knowledge for two chess etudes which are intractable for exhaustive search chess programs -- the Reti and Nadareishvili etudes suggested by Botvinnik to test the quality of knowledge intensive solutions in computer chess [17,18]. Particularly, Nodareshvili etude can be solved only in the case of 36 depth of search in the game tree, while exhaustive search chess programs, in average, search for the depth 6 .

In 1912 Zermelo proved that all chess positions are strongly divided into three classes: winning, losing or drawing [12]. In [38] a correspondence was revealed between Zermelo's classes of winning chess positions, strategies and typical units of chess vocabulary. While this correspondence supports the constructive nature of human models of chess realities it, at the same time, states that the real implementation of these models, in principle, can only approximate (to a specification) the winning game tree structures due to the irresistible complexity of computations required to prove the correctness of the models. Thus, it follows that

i. for the same units of chess vocabulary both chess players and computers will, as a rule, use different models of chess realities essentially based on their individual experience

ii. any preferences between these models include uncertainty

iii. interpretations of the units of vocabulary should rely on some personalized experience.

Therefore, strategy formation systems should be able to integrate common chess knowledge with the personal expertise of particular players and be able to improve their performance by regular acquisition, learning or inferring the contents of the units of chess vocabulary.

The study reports on our progress in programs acquiring typical categories of chess knowledge and is structured as follows. Sections 2 and 3 define RGT problems, refine the framework of study and measures of effectiveness of competing programs to induce corresponding requirements to them. Section 4 describes the formal structures of attributes, goals, strategies, plans, etc., and our approach to categorize a repository of chess vocabulary in about 300 units [24] by these types. Section 5 specifies the *Personalized Planning and Integrated Testing* (PPIT) algorithms capable of elaborating moves in target positions dependent on typical categories of chess knowledge and demonstrates the effectiveness of PPIT algorithms by experiments in acquisition of expert knowledge-based solutions of two topBotvinnik'stestetudes in strategy discovery, the Reti and Nodareishvili etudes. The Conclusion underlines the main findings of the research.

Regularity of knowledge acquisition was ensured through developing the graphical language interface [68], programs of presentation, acquisition of meanings and matching to situations [71-79] .Instructions for operations with problems of RGT class are formulated in a unified, problem independent of allowing the specification and communalized knowledge of particular problems step by stepinput intoSolver.

Effectiveness of the Solver was successfully tested for network intrusion protection problems using corresponding expert rules and goals [51]. Experiments on protection against representatives of four classes of attacks: SYN-Flood, Fraggle, Smurf and Login-bomballow formulating, in particular, the following statements:

- The number of nodes searched by the IGAF2 version of Solver for making decisions with all expert rules and subgoals is the smallest compared to the IGAF1 version of the Solver with the minimax algorithm in which the depth of the search is increased up to 13
- The recommended version of IGAF2 with all expert rules and subgoals for the search depth of search 5 and 200 defending steps, is outperforming the productivity of the minmax algorithm by 14% using 6 times less computing time and searching 27 times fewer tree nodes.

Measures of effectiveness of solutions of RGT problems and proof of viability of measures were studied for chess, management and intrusion protection (IP) problems [].

Local tournament-based methodology for measuring the effectiveness of Solver for chess and management problems is presented in [42,65,70].

In [40], we argue that performance of max Sum criterion for certain constrains can be a common measure for Human/computer strategy performances. We state that rising the ranks of programs in the max Sum orderings, in contrast with chess players, is not necessarily caused by possessing more chess knowledge and skill.

If changes in the ranks of programs in the max Sum ordering are caused *ceteris paribus* by knowledge acquired by programs from the hierarchies of chess knowledge (e.g. from handbooks) then these changes can be interpreted as caused by knowledge acquisition and can be compared with a human ones by the same max Sum criterion.

[51] study the criteria for comparing IP algorithms, estimate the 'distance" of current states of protected systems from the normal ones and the level of performance that IP algorithms can preserve .

A special tool was developed and successfully applied to estimating the quality of protection against unauthorized access. The tool allows us to vary the component parts of experiments, such as estimated criteria, types of attacks, and parameters of IP systems.

7.5. **Meaning Processing** (we skip this part of the paper due our models of it were already presented in preceding Chapters).

7.6. RGT Interpretation of Ownship Air Threats Defense

7.6.1. Ownship Air Threats Defense Games in dimensions of RGT class were studied within of the framework of NATO CLG grant in 2007-09 with the University of Montreal, CA, [81].

States, Situations, Actors. The game involves two *parties* (actors) designated "defense" and "threats", respectively. Each party contains players. Each player responds to the actions taken by the opposite party.

The defense party has a single player, i.e., the ownship.

The threats party may have several players in the form of missiles and aircrafts. The types of threat players can be regrouped into categories, e.g., missile of type xxx, aircrafts of type yyy. An additional category can be defined for threat players whose type is uncertain.

In the simplified scenario, all threat players belong to a single category of missiles. Several threat players may attack concurrently.

Threat players are generated as follows:

a) All threat players are created at the start of the scenario.
b) The maximum number of threat players is $N_{threats}^{max} = 8$.
c) The initial position of each threat is uniformly and randomly selected in an area of space satisfying the conditions:
   - Initial range from 5 to 80 km from ownship,
   - Polar angle between 0° and 90° (i.e, angle in the vertical plane),
   - Any azimuth angle (i.e., angle in the horizontal plane).

   Assumptions:

   - Threats are assumed to be ranked by the defensive player. In the simplified scenario, the ranking function is the range: the closer the threat, the higher the threat rank .
   - It is assumed that the defensive player may bundle up concurrent defense actions. Admissible bundles must satisfy the engagement rules.
   - A bundle of defense actions must ensure that only one defense action per threat is undertaken at any given time.
   - Each action results in a transformation in the scenario. The sets of defense actions, defense bundles, threat actions, and their associated transformation rules are assumed to be finite and known.

Actions. It is assumed that all *defense actions* of the ownship involve a sequence of three steps:

a) search and lock on a target, b) fire and target intercept and c) kill assessment.

It is also assumed that fixed time periods are required for completion of the search and lock step, and of the kill assessment step.

Three types of defense action are available: i) launch a long-range surface-air missile (SAM), ii) shoot the medium range gun, and iii) shoot the short-range gun. Each defense action requires a period of time to elapse before completion. This time period, Δt_c, is calculated as $\Delta t_c = \Delta t_{search} + \Delta t_{fire} + \Delta t_{kill}$.

Characteristics of defense actions and time periods are provided by experts.

It is assumed that there is a single category of threats: an anti-ship missile (ASM). This ASM has a single available threat action: directly coming towards the ownship. The characteristics of this threat action are: speed: $v_{ASM} = 850\ m/s$, trajectory: straight line, kill probability: $P_{ASM}^{kill} = 0.50$.

Evaluators. In the simplified scenario, the ownship has the primary defense objective to survive, while the primary objective of the threat party is to damage the opponent. The defense and threat objectives are described by utility functions denoted $\boldsymbol{U_D}$ and $\boldsymbol{U_T}$, respectively.
Two utility functions are involved, one for the defense player and one for the threat players. The purpose of the defense utility function is to weight each defense strategy with respect to the defense objectives. Similarly, the threat utility function weights the threat strategies with respect to the threat objectives.

The defensive utility function $U_D: P_D \times P_T \longrightarrow R^1$ is selected to be the survival probability of the ownship in the worst-case scenario. The worst-case scenario is the one in which the threats and the ownship always survive for defense and threat actions.
In the simplified scenario with a single threat action, the probability of survival is calculated as:

$$\boldsymbol{U_D}(P_{\boldsymbol{D}}, P_T) = f(P_{\boldsymbol{D}}) \prod_{i=1}^{N_{engaged}^{threats}} \left(1 - P_{kill}^{i}(P_{\boldsymbol{D}}) P_{kill}^{threats}\right),$$

where

$$P_{kill}^{i}(P_{\boldsymbol{D}}) = \prod_{j=1}^{N_{actions}^{i}} \left(1 - P_{kill}^{actions}(i,j)\right)$$

$$f(P_{\boldsymbol{D}}) = \begin{cases} 1, \text{if } P_{\boldsymbol{D}} \text{ engages the } N_{engaged}^{threats} \text{ closest threats} \\ 0, \quad \text{otherwise} \end{cases}$$

$$N_{engaged}^{threats} = \begin{cases} N^{threats}, \text{if } N^{threats} \leq 3 \\ 3, \quad \text{otherwise} \end{cases}$$

with $N_{engaged}^{threats}$ is the number of threat that can be engaged, the function $f(P_{\boldsymbol{D}})$ ensures that only defense strategies that engage the closest threats first are considered, $P_{kill}^{i}(P_{\boldsymbol{D}})$ is the probability that threat *i* is destroyed by the defense strategy $P_{\boldsymbol{D}}$, and P_{kill}^{threat} is the probability that the threat action destroy the ownship. (The value P_{kill}^{threat} and value $P_{kill}^{actions}(i,j)$ of probability that the threat i is destroyed by the defense action j for each type of action are provided in the original paper).

The threat utility function is selected to be the opposite functions $U_T = -U_D$ due to the choice of defense and threat. Utility functions being opposite functions, a zero-sum game is generated. The solution of the zero-sum game guarantees that the utility of the defense player has at least the value
$U_D^*: U_D^* = \{max_{P_D \in S_D} min_{P_T \in S_T} U(P_D, P_T), U(P_D, P_T), U_D(P_D, P_T)\}$
provided that the defensive player adopts the game optimal defense strategy
$P_D^*: P_D^* = argmax_{P_D \in S_D}\{min_{P_T \in S_T} U(P_D, P_T)\}$,
$U_D^* = max_{P_D \in S_D} min_{P_T \in S_T} U(P_D, P_T),\ U(P_D, P_T),\ U_D(P_D, P_T)$

Regularities. Transformation rules (regs) describe the outcome of defense and threat actions. In this scenario (game), the outcome of all the defense and threat actions is uncertain. The outcome can be one of two possibilities: (i) the opponent is destroyed, or (ii) the opponent survived. Hence,

a. Transformation rules of defense actions
e. the threat is destroyed with probability P_{kill}, or
f. the threat survived with probability $1 - P_{kill}$
b. Transformation rules of threat' actions
i) the ownship is destroyed with probability P_{kill}, or
ii) the ownship survived with probability

7.6.2. We prove that a certain algorithm GNGTS in the given situation S and max depth H of analysis of strategies of the naval game tree (NGT) guarantees the generation of all NGT strategies rooted in S.

Thus, by standard Solver methodology we can generate NGT strategies, enhance them with expert knowledge and others to search the most acceptable of them.

The results of experiments with searching the best strategy in NGT for a single ownship are provided in [81]. It appears that for the case when waiting conditions are ignored, the search time in NGT for the best strategies is significantly less compared to those for TABU system.

7.7. Conclusions

7.7.1. We have presented evidence justifying that

-urgent combating and competition problems are reducible to the standard kernel problem K , while the achievements for K and other problems of RGT class are multiplied for the entire class.

- K-centric methodology enhances the effectiveness of RGT Solvers, answering the urgent questions onmeasuring the effectiveness ofsolutions, analysis and typification of combating knowledge, construction of knowledge based Solvers, regular acquisition of expert knowledge to enhance the effectiveness ofSolvers, etc.

The validity of K- methodology was proved for Chess like combinatorial games of Network Intrusion Protection, Navy Defense from Attacks, Management, Marketing, Ownship Air Threats Defenseand other problems.

The shell of RGT Solvers is developed to provide a user-friendly Java environment for managing any SSRGT problem

7.7.2. In totality of all problems the combinatorial ones, in contrast with regularized and algorithmically solved problems [17,18,27], require the entire power of human cognition to find solutions just in the same way as humans solve the problem of their being of *humans* in the *Universe* (HU).

RGT problems are combinatorial and HU can be interpreted in RGT class (Chapter 2, Part 1).

Developing RGT Solver, we synchronously do tend to refine HU and the ways to solve HU.

8. Contents of Appendices with Their References

Appendix 1. Solvers in General

1. E. Pogossian, “Effectiveness enhancing knowledge-based strategies for SSRGT class of defense problems”, NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems, Salamanca, Spain, 2011, pp. 16.
2. E. Pogossian, V. Vahradyan, A. Grigoryan. On Competing Agents Consistent with Expert Knowledge", Lecture Notes in Computer Science, AIS-ADM-07: The Intern. Workshop on Autonomous Intelligent Systems - Agents and Data Mining, June 5 - 7, 2007, St. Petersburg, p.11.
3. E. Pogossian, “Specifying personalized expertise. International Association for Development of the Information Society (IADIS)”, International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006), Barcelona, Spain, 2006, pp. 151-159.
4. Pogossian E. On Measurable Models of Promotion of Negentropic Strategies by Cognition, New Trends in Information Technologies, ITHEA, Sofia, 2010, pp.161-168.
5. Grigoryan S., Hakobyan N. and Baghdasaryan T., Knowledge-Based Solvers for RGT Combinatorial Problems, 13th International Conference in Computer Science and Information Technologies, CSIT2019, Yerevan, 2019, pp36-46, also reprinted in IEEE's Xplorehttps: //ieeexplore.ieee.org/xpl/conhome/8890749/proceeding.
6. Hakobyan N., “A System for Transforming Images to Symbolic Presentation for Combinatorial Defense and Competition Problems”, ISSN 0002-306X. Proc. of the RA NAS and NPUA Ser. of tech. sc. 2019. V. LXXII, N2, pp. 199-209.

Appendix 2. Developing OJP Methodology

1. Pogossian E. Management Strategy Search and Assessment Programming. CSIT99, Yerevan, August 1999.
2. Danielyan E., Pogossian E. On Expanding of Method of Local Measurement to Evaluation of Management Strategies, Proceedings of SEUA, 2009.
3. Pogossian E., On Performance Measures of Functions of Human Mind.
4. Pogossian E., On Assessment of Performance of Systems by Combining On-the-Job and Expert Attributes Scales International Conference in Computer Sciences and Information Technologies, Yerevan, Armenia, 2015 pp. 331-335.
5. Berberyan L. Modular Tool for Regulating and Analyzing Activities in Chess, Mathematical Problems of Computer Sciences, 47,2017.

Appendix 3. Solvers in Management

1. Danielyan E., Pogossian E., "An Application of Voting Methods to Strategy Assessment", Proceedings of the Conference on Computer Science and Information Technologies (CSIT'99), Yerevan, Armenia, 1999.

2. Baghdasaryan T., Danielyan E, Pogossian E. Testing Oligopoly Strategy Plans by Their on the Job Performance Simulation. //Proceedings of the International Conference CSIT2005, Yerevan, Armenia, 2005.

Appendix 4. Solvers in Intrusion Protection

1. Pogossian E.: Combinatorial Game Models for Security Systems. NATO ARW on "Security and Embedded Systems", Porto Rio, Patras, Greece, Aug. (2005) 8-18
2. E. Pogossian, A. Javadyan and E. Ivanyan., "Effective Discovery of Intrusion Protection Strategies.," The International Workshop on Agents and Data Mining, Lecture Notes in Computer Science, St. Petersburg, Russia, pp. 263-274, 2005.

Appendix 5. Solvers in Defense

1. E. Pogossian, D. Dionne, A. Grigoryan, J. Couture and E. Shahbazian, "Developing Goals Directed Search Models Empowering Strategies Against Single Ownership Air Threats," International Conference in Computer Sciences and Information Technologies, pp. 155-163, Yerevan, 2009.

Appendix 6 . Communication Solvers

1. Arakelova E. and Pogossian E., "Tools for testing and correction of the completeness of knowledge acquisition by autistic children," pp. 159-165, Yerevan, Armenia, 2011.
2. Pogossian E, Grigoryan S. and Berberyan L., Personalized Interactive Tutoring in Chess, 2016.

Appendix 7. The Repository of Chess Classifiers

1. Hambartsumyan M., Harutunyan Y., Pogossian E. The Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations, National Academy of Sciences of Armenia, IPIA, 1974-1980 research reports.

Appendix 8. On Background of RGT Solvers, OJP and Cognizers

1. Pogossian E. (1985) Adaptation of Combinatorial Classifiers and Controllers, DrSci, Computing Center of Academy of Sciences, Moscow, defense thesis.
2. B.K. Karapetyan, ОБ ОДНОЙ УНИФИКАЦИИ ПРЕДСТАВЛЕНИЯ ШАХМАТНЫХ ПОНЯТИЙ, Mathematical Problems of Computer Sciences, 1986, pp. 181-193.
3. B.K. Karapetyan, ЯЗЫК ВЫСОКОГО УРОВНЯ ДЛЯ ОПИСАНИЯ СТРАТЕГИИ В ИГРАХ, Mathematical Problems of Computer Sciences, 1986, pp. 167-183.
4. E. Pogossian, "On a transparent presentation of written English syntax", *5th Intern. Cognitive Linguistics Conference, VrijeUniversiteit*, Amsterdam, 1996, pp. 209-214.
5. A. Martirosian, E.Pogossian. "Learning by Revelation". Reference book on Intellectual Systems. Radio i Svjas Publishing Company, Moscow, (in Russian), vol. 2, pp. 206-231, 1990

Appendix 9. On Background of Empowering Cognizers by Tournments

1. Pogossian E. Systems of sets with minimum number of subsets and application to the theory of testing". PhD, Computing Center of Academy of Sciences, Moscow, defense thesis, 1970.

Summary of Chapter 3/1

1. Mentals are ontological, systemic, and consistent with natural languages models of m systems.

Criteria for the adequacy of mentals in modeling msystems are chosen analogously with the criteria for the adequacy of algorithms in modeling computable functions by Church, or deterministic methods by Turing, Post and Markov.

Following algorithmic criteria, the adequacy of mentals in modeling explaining, understanding and human-computer communicating is discussed, as well as criteria of the adequacy of job performance are discussed to apply, then, to examine the adequacy of selected cognitive doings within of the framework of RGT problems.

2. The statements we make in this Chapter 3/1 can be summarized as follows:

Stt 1. *For languages L of communities C allowing the members x of C to communicate, i.e., to explain and understand mss of each other expressed in L, communication algorithms LC can be constructed letting computers communicate mental models Mns of mssMs of C equally wrt the members of C if Mns and Ms are equal to each other.*

Stt2. *Mentlals can consist of functional and connectivity mental models.*

Stt5. *Mentals, in principle, can be adequately modeled by ANN that, at first, represent 1-/2-place rels of mentals by equal to them unit ANN, then, compose those units into ANN equal to the basic constituents of mentals and, finally, unite these constituents into ANN equal to the target mentals.*

Stt6: *Given A and BNN classifiers , the rels1 between A and B can be formed as a new NN classifier A rels1B , linked by synapses, necessarily, to NN classifiers of A, B and, expectably, to other ones.*

3. Acknowledging that mss are far from being perfect, thus, their effectiveness, in principle, cannot be unanimously and separately measured we conclude they should be evaluated by the integrative outcome of all mss involved in the performance wrt utilities, i.e., by their On the Job Performances (OJP) and we strengthen the OJP evaluation of the performance of Solvers by analyzing them in competititive tournaments with counter partners.

3.1. We introduce conceptual frameworks for OJP and RGT problems/Solvers that are detailed and deepen in *Appendicies 1-9* along with a description of their successful applications.

Then, present an approach to scaling the effectiveness of models by their On the Job Performances and argued the reasons why the estimates of examination of the adequacy of RGT cognizers can be transferred to the estimation of adequacy of models of Human cognizers in the Universe (HU).

As an introduction to RGT Solvers and some of their applications we reproduce our presentation at NATO ASI 2011, Prediction and Recognition of Piracy Efforts, Salamanca, Spain, September.

4. At present, we focuse on adequate modeling of nuclear constituents of cognizers in frame of RGT problems, on the development of the shell of RGT Solversas, well as on the design of compilers of chess texts in their meanings and vise versa.

We have also a positive attitude to the projects:

4.1. Development of Autonomous Knowledge-Based Systems for Decision-Making on Battle Fields

4.2. Expert System for Prognostication of Infections /*starting from Viral*/ of Defense Strategies

5. Finally, let us provide research inquiries expanding and developing the ad hoc ones:

5.1. To constructing RGT Solvers based on the of mentals.

5.2. To extend the chess scales of the effectiveness of strategies to the entire RGT problems.

5.3. To consiste scales of the effectiveness of mental doings of humans with those of doings of Solvers.

5.4. To compare the adequacy of OJP and attributive estimates of the effectiveness in attaining certain solutions.

Chapter 4/1. Specifying Constituents of Root Cognizers

1. Preface

In what follows we will address the problem on learning mss.

At first, we interpret that already introduced constructive models of mss, *mentals* by the basics of attributive classification of theoretical computer science, linking to each other, particularly, the attributes and the types of classifiers of mentals, teaching samples of experts and matrices of stored prints, addressing to the ways of regular extension of attributes and to the origin of expert-independent criteria to split realities aspositive or not.

Then, referring to some relevant models of classifying, we provide an approach to inductive formation of two place relationships that, beingthe basic units of mentals, similar to those in clauses of languages, induce guides of formation of mentals, particularly, in the artificial neuron nets (ANN) modes.

Finally, we discuss the formation and acquisition of systemic classifiers and the ways of coping with imperfect classifying.

2. Refining Models of Mental Classifying

1. Let us recall that so far the following types of classifiers were introduced:

-sensors xSns,

-genomic goals xGn, including evolutionary approved goals, i.e., the classifiers of, at first,roots and utilities wrt the roots, then, the consequent classifiers, goals chained wrt the preceding ones,

-genomic classifiers *xClgn* uniting Sns and sGn,

-goals *Gl* corresponding to classifiers *xClt learned* in the life time (lt) of *x* and uniting *do* classifiers *xdoCl* by themselves and systemic classifiers xsCl induced by ad hoc *do* classifiers of the constituents of mentals of the thesauri *xTh*,

-goals *xG* uniting *xGn* and goals *Gl* and, finally,

-attributes *xAtr = x(ClgnU xdoCl)* comprising all *do* classifiers of *x* available at time *t*.

Note, that genomic goals are classifying utilities *wrt* the roots and , we assume, include, to some extent, the classifiers of roots themselves. These classifiers are enriched throughout life times by those of high layers of humans studying all over including the roots.

1.1. The unit classifiers *Cl* of *xAtr*, the *attributes*, at time *t* output their IDs for input positives of +*Cl* at *t*, i.e., when they are activated by positives of +*Cl* at *t*.

Bundles of IDs of activated at *t*+ Δ attributes, or *t prints*, of *x* are accumulated at stores *xSP* since the moment *t=0*, i.e., the time when *x*, we assume, can be classified as organisms, while Δ is the discretion time sufficient for outputting of attributes of any layers depended from activation of the attributes of the lower layers.

The diversity of all possible prints for *xAtr* at *t* comprise the space xSP* of possible prints at *t*. Apparently xSP* is a representation of the Universe xU of *x* at *t*.

2. Communities *C*, we assume, have the ability to address and to certain extent control the classifiers of C members.

Thus, we can assume, that *C* can control the unions of classifiers of the types of those of *C* members, i.e., *C* can control classifiers of the types of *Sns* and *Gn*, since the members of *C* have almost identical genomes, then the types of doClC, sClC, ThC, GC, GlC and AtrC .

We also assume the common for *C* Universe UC and space of prints SP* uniting xU and xSP* of all *x@C*, correspondingly, and determined by AtrC .

2.1. These days mental doings of members *x* of *C* can be deeply analyzed mainly personally, while analyzing the integrative power of *C* it is worth addressing to the unions of mental doings of x@ *C*.

For convenience, therefore, let's address to the ideal members ***z*** **o**f *C* identified by classifiers of Sns, Gn, Th, Clt, G, Gl, Atr, SP* and U coinciding with analogous integrative ones of *C*.

3. While prints and their spaces are determined by Attributes at *t*, the outputs of sensors, *percepts,* united with the outputs of genomic goals Gn and the outputs of classifiers Clt, i.e., the *learned prints*, determine the space of genomic prints SPgn and the learned space of prints SPl,correspondingly, with the powers 2^n and 2^m where n and m are the powers of SnsUGn and Clt at time *t*, correspondingly.

Apparently, the spaces of prints SP at *t* will have the power 2^{n+m}.

3. Referring to Attributive Classifying

1. Mental *learning* to classifying includes *reveling, discovering,* in general, the *formation* of classifiers by members *x* of communities *C* over lifetimes, where learning, particularly inductive one, necessarily assumesa unique contributions of *x* to this formation, and includes the *acquisition* of ready-to-use classifiers of *C* with more standardized, at the same time minimized doings of *x*.

1.1. Addressing to the learning of classifiers (LC) Cl of *z@C* it is worth, first of all, recalling the classifiers in advanced models and theories where classifiers are represented by compositions of certain attributes for the followed then transition of the statements of these theories to the modeling LC by mentals.

1.2. Let's also remind that mentals aim to represent, at least, realities having corresponded and identifying them units of languages (ideally, natural), while represent realities consistently with the scope of those units.

For example, mentals representing chess positions should include relevant classifiers of game trees, the best strategies or their search algorithms.

2. In logics classifiers can be interpreted as propositional formulas, while LC can be interpreted as a kind of logical inference of theorems from the given axioms.

Axioms can be interpreted as classifiers with certain trustful utilities while theorems as those the utilities of which should be inferred from axioms applying trustful logical rules, such as, modus pones.

Apparently, the utilities of theorems will be transferred from those of axioms, thus, compared with the axioms will not enhance the reliability of those utilities wrt, for example, the genomic goals.

2.1. Instead of logical rules the inference of new classifiers from those with certain utilities can proceed by means of some case- effect regularities.

For example, from the given chess positions P strategies can infer P from positions P^* with already known utilities, thus, transferring these utilities to P as well [9].

2.2. Along with targeting inferences of theorems or winning positions, targets can be classifiers of inferences themselves.

For example, in chess game trees for the given positions *P,* the classifiers of proper in P strategies can be questioned [9,10].

Search for classifiers of proper strategies in the entire thesauri of mentals, can be formulated similarly.

2.3. Some studies by N.Nepeivoda question the transition from the classifiers of inferences to the classifiers of algorithms of these inferences.

Others focus on the transition from implicit classifiers to the explicit ones.

For example, in [11], implicit classifiers can be given by sets of equations and the explicit ones are searched using the theorem of the fixed point among the space of codes of all possibly relevant classifiers.

3. A type of representations of propositional formulae (in fact, the classifiers) are Boolean functions [12].

Boolean models of LC address, particularly, to the inductive restoration of Boolean functions Cl, potentially being justified, say, by experts or oracles, by consistent and representing Cl *matrices of* samples of *Boolean vectors* mbv either from +Cl or not.

Algorithms of generalization of matrices mv*, inductors,* and *algorithms of inductive inference* of Cl by regular processing of inductors over the sequences of matrices mbv and the frontiers of such approximation of target Cl are broadly analyzed and represented, particularly in [9].

For example, a type of inductors compacts the mbv matrices representing them by equal propositional formulae Clgi of the types of conjunctive of disjunctive forms composed by logical operators from the attributes.

Let's emphasize that the matrices *mbv* provided to the *inductors of propositional formulae* (IPF), presumingly, should be enough representable to restore these formulae, can be entirely available at ones, in parts, repeatedly or not, etc., following some heuristics of IPF.

4. Learning in ANN is carried out by inductors and inductive inferences of certain types as well [9].

Addressing to *learning by back propaganda* (LBP), it can be classified as a type of inductive inference where the outputs of target classifiers built by certain inductors are cyclically compared with those expected by teachers and, correspondingly to the coincidence or not of the outputs with the expectations, the cycle of applying the inductors is continued or not.

The uniqueness of the currently used LBP compared with IPF is in the sequential, one by one provision of the vectors of the matrices *mbv*.

It could be an advantage of LBP or analogous inductors when the analysis of the entire matrices by some reasons is meets some difficulties.

5. Assuming that LC in addition meets certain statistical requirements, the chains of consequent statements on the formation and quality of classifiers are presented, particularly in [13].

4. Continuing Refinement of Mental Classifying Models

1. Mentals are defined as systems over the outputs of growing up attributes where the basic ones comprise genomic classifiers including sensors and those formed evolutionary.

Assumingly, genomic classifiers are permanent, lasting constituents of attributes, while new classifiers enriching them, not necessary can be involved in ongoing classifying.

1.1. Sensors have a unique physical nature related to vision, touching, hearing, etc.

Ongoing models of human vision, for example, are based on the light sensitive units of retina linked by weighted synapses to a type of neurons, the visual sensors.

The distribution of synapses over the retina and their number for each of these primary neurons, sensors, seemingly, should be normal to ensure equal inputs to the sensors from all parts of the retina while their density, similar to eyes, should be much higher for the neurons linked to the center of the retina than for the periphery ones.

It is not excluded that the weights of sensor synapses of can be changed to correspond to the regular peculiarities of the inputs analogously, in a similar way, with the facets of frogs specialized to react only to particular inputs of types of lines, motions, etc.

1.2. For further reference the above assumptions we comprise as follows

Ass1. *Genomic classifiers are permanently involved in new classifying, while those of the sensor type are specialized in representing certain fields of realities and vary in the types of representations of each field.*

2. In machine learning of classifiers *Cl* positives of *+Cl* are provided by experts, while in autonomous inductive learning the sources should preliminary be refined .

The given sources of positivity learning can be processed on the base of the positives provided either by matrices of samples of *+Cl* similar to the case of Boolean IPF or sequentially similar to LBP in ANN.

2.1. In refining *positivity of realities wrt classifiers* we believe they comprise positives *+Cli* if they have a common relationship with certain identified utilities of humans that can be interpreted as goals.

For example, positives of the class *Sweets* necessarily should provide utilities *being sweet.*

Those utilities can address to the tangible realities of the classes *Sweets, Soars, Cold, Hot,* etc. or to the classes *Nutrients, Damagers*, etc., but they can also refer to non tangible, highly abstract goals like *Classify All Over.*

In the early stages of childhood, these classifiers are Piaget's sensori-motor classifiers and are tidily governed by the right hemisphere. And only at later stages they incrementally enhance their abstractness and can be located in the left hemisphere as well.

2.2. Recall also that in the attributive formation of classifiers *Cl* there are certain criteria, goals *gi, wrt* which those *Cl* are formed.

For example, inductors in Boolean models generalize matrices of Boolean vectors mbv consistent with positives *+Cl* of target classifiers *Cl*. In these models, the question of positivity or not of the vectors of *mbv* is prearranged by humans, and their origin is not discussed.

2.3.We resume the above notes by the following

Ass2. *Formation of do classifiers Cl of **z** @C at time t is always processed wrt certain goals g available to z at t.*

Clr.1.2. *Do classifiers Cl of thesauri Th are always accompanied by certain explicit or implicit goals g representing those that guide the formation of Cl.*

2.4. Thus, in modeling or processing of *do* classifiers *Cl,* it is inevitable to be aware of the goals *g* guiding their formation.

And it is reasonable to assign Clgi to classifiers formed *wrt* goals *gi* , while, similarly, *mpgi,* or *matrix of prints wrt gi*, to assign to the sub sets of the stores of *t* prints SP comprising ones accompanied by activation at *t* with taken place goals *gi*.

2.5. Addressing to inductors above let's` induce the following corollary

Clr.2.2..*Inductors have to be interpreted as algorithms providing hypothesisis on classifiers Clgi by extension of given matrices mpgi consistent, in the essence, with +Clgi* [9].

2.6. When questioning goals accompanying the systemic classifiers sCl, similarl to the *do* ones, let's recall that sCl are induced by *mss m* (or by mentals *m*' modeling *m*) which, in turn, are composed of mdoers (mdoins) why the goals of *m* (m') and sClis reasonable should be determined in certain dependency from the goals of those mdoers (doins).

3. Recalling the views *"Knowledge is Power"*, *"Seeing by the Mind* [14] "we interpret them as ***"Classifiers are Power "*** and ***"Seeing by Classifiers"***, thus, following the assumption below.

Ass3.*All ad hoc classifiers can have a positive impact on the formation of new classifiers*

3.1. Particularly, it follows that in the mpgi based IPF and in the sequential LBP formation of new classifiers it is recommended to take into account the attributes corresponding to the representation of all ad hoc classifiers.

Apparently, which parts of, in general, lengthy prints have to be involved in the formation of classifiers and how to do that needs special refinements.

For example, Boolean inductors, before generalization of *mpgi* matrices into conjunctive of disjunctive forms compress the attributes of initially given *mpgi* to the min necessary ones, i.e., to the min tests [12], to preserve the correctness of *mpgi*.

5. Learning Mental Systems

1. rC* construction problem address to provision of constructive cognizers capable to learn and organize mss with min complexity up to attaining effectiveness, at least, comparable to the highest level attained by human cognizers Cogs.

Cognizers rC* tending to adequacy with human ones necessarily should be analyzed within the framework of certain communities and have to possess constituents inevitable for any cognizers,including roots, energizers, sensors, effectors , controllers, stores of prints, means of extraction of matrices of imprints, revelation mss and acquisition mss from communities as well as means of enhancing the effectiveness of mss.

While autonomous agents meet many of these requirements, learning of mss continues to be one of the most questioned.

2. In what follows we are going to study cognizers containing, at least, the above constituents and, first of all, are going to focus on the learning of mentals argued earlier as adequate models of mss.

And we are going to equally reduce the learning of mentals to the learning of the constituents of mss.

3. Following the above plan, let's first recall that mentals are incremental systems of doins, i.e, doers over IDs of nominals, whereas doers can, in fact, be interpreted as a type of algorithms.

And since doins are assumed to be algorithms they can be represented in the mode of Markov algorithms, defined as compositions of 1-/2-place rels, or 1/2rels, over certain alphabets.

Recalling that 1/2rels are defined as *do* classifiers over certain alphabets of IDs it follows that Markov algorithms , particularly, can be composed of 1/2 rels represented by classifiers induced by mentals.

Nets representing the unions of *do* 1/2rels with 1/2rels represented by systemic classifiers of nets of mentals N will be denoted by N1/2nets.

4. Then, let us remind that mentals are constructed incrementally, layer by layer, as systems where those of higher levels are assembled by rels between already available systems.
In other words, considering 1/2/rels as starting, basic systems, the next layer systems, i.e., algorithms/doins, are assemblies of available 1/2rels, while systems of the next higher layers are compositions of already constructed systems and 1/2/rels.

5. The above assumptions make it possible to equally reduce the learning of mss to the learning of the following constituents of mss, namely, to the learning of

- 1/2rels
- Markov algorithms/doins given nets of 1/2rels
- mss given algorithms/doins and 1/2rels.

And, apparently, learning of each of the above constituent should be highlighted both by revelation and acquisition dimensions of learning mss, as well as by the dimension of learning communicatives (cms) of languages of communities corresponded to IDs of mss and their constituents.

Let us analyze those constituents in more details.

6. Learning Relationships and Communicatives

1. The fundamental constituents of mentals are relationships and doins that, in turn, can, in principle, be reduced to *n*-place *do* classifiers, relationships, over the nominals .

Although learning of 1-place classifiers is so far widely illuminated it needs to be refined, at least, for 2-place ones as it follows.

2. *mpgi* based IPF can form 1-place classifiers *Clgi* analyzing only positives of *mpgi* but become expectably more consistent with *Clgi* if accompanied by matrices *–mpgi* representing non positives, negatives of *Clgi*, i.e., realities not activating goals *gi*.

2.1. Rels, in general, classify being in space and time of two classes of realities represented by their IDs.

So, A*rels1*B means that nominals with IDs A relate to ones with IDs B as *rels1*. Analogously with IPF learning of 1-place classifiers by matrices ±*mpgi* , the **IPF formation of A*rels1*B** can be proceeded for mpAB matrices of prints with positive values of attributes A and B split into two parts ± mpAB, where the prints of the +mpAB parts meet the rels1, while the rest of mpAB, i.e., -mpAB, don't.

Apparently, classifiers *Cl* correctly classifying ± mpAB are algorithmic, while the problem of classifiers *Cl* consistent not only with ± mpab, but also with the target A*rels1*B, have not universal algorithmic solutions and for each ArelsB has to be uniquely analyzed given additional constraints.

3. Classifiers A*rels1*B identify rels1 between the particular classes of realities with IDs A and B.

For± *mp* based **learning of classifiers of *rels1*** themselves, **in general,** i.e., the formation of *xrels1*y where *x* and *y* can be arbitrary IDs, the +mprel1 matrices can comprise available prints of SP stating the presence of activated attributes Arel1B between any IDs A and B, while –mprel1 should include certain opposite prints where, at least, there is no rels1 between these IDs.

4. LBP learningof Arels1B and *x* rels1*y* can b**e** proceed in similar way with the sequential provision and learning to the prints of ± *mp* matrices.

Particularly, realities *r* given in the discretion time Δ to the interface of, say RGT Solvers [17], form prints comprised of IDs of all activated at each Δ time attributes caused by r.

Then, LBP should be processed for each of these realities *r* at the interface until the target classifiers *Clgi* will be formed that will be equal to one formed for the same prints but provided totally as the *mpgi* matrices.

And these newly formed *Clgi* enrich the attributes to be used in further multilayer LBP.

5. Let's state the above findings in learning of rels as follows:

Ass1. *Inductive algorithms of learningof rels can be particularly constructed of the types IPF or LBP and be based on mpgi matrices that are not necessarily numerical and can be both between personalized classifiers of sole realities, as well as between variables of classifiers of rels of sole realities, thus, representing sole rels themselves,*

Ass. 2. *The above algorithms can be processed, particularly, within the framework of ANN*

Ass. 3. *Analogies in learning and storing rels in NN can essentially enrich those for mentals and ANN, while these models can enrich the understanding of NN.*

6. Thus, learning rels can be processed by inductive revelation/ discovery within the framework of genomic and cultural heritage of communities, from case - to rule-basedrepresentations of rels and be either supervised or not.

Both of the aforementioned learners of 1/2rels, namely inductors, transfer classifiers represented by mpgi or mpgigj case–based-matrices into generalized rule-based parent classifiers represented, for example, as DNF over attributes of matrices. Let us remind also that these matrices are extracted from regularly enriched stores of t-prints by certain algorithms, say, g casers and gg casers, extracting mpgi and mpgigj, correspondingly. Then acknowledge that case based matrices have IDs and already represent rels, or *1/2 caserels,* given by sets of cases.

Apparently, these 1/2caserels developed into rule based ones, or 1/2rulerels, also need their individual IDs to be referred.

7. Learning communicatives. When classifying imprints, we learn the ground classifiers, i.e., the rels themselves, and classifiers of cms of community language units assigned to IDs of those rels and their systems, mss.

Cms are learned by samples of corresponding classifiers and , as a rule, inductively and supervised.

In other words, the following statement is acknowledged:

St. *Learning of rels and mss is, as a rule, supervised by communities, explicitly or implicitly, and is accompanied by inductive learning of classifiers of cms of community language units assigned to IDs of these rels and mss.*

7. Acquisition of Mental Systems

1. Let us preliminarily remind that the methods of abstract classes, if represented by Markov algorithms, are systems of non static rules, regs of the types of "realities *y* follow realities *x*", "*y* is a function of *x*", "*x* cause *y*", etc., where the order of appearance of x,y in time is essential and cannot be reduced to the static, space rels.

Recall also that these rules can be reduced to a type of time, case-effect, dependency rels as it was already argueed.

And although we learn classifiers of rels and cms of mss by samples of corresponding classifiers, and, as a rule, inductively, the vast majority of mss we acquire from communities layer by layer, where mentals of higher levels comprise systems linking mentals of lower ones by 2rels of some, in fact, steady enriching sets.

1.1. Indeed, there are premises that 2rels were initially learned as 2rels between two particular mss x1, x2 to be in the next steps generalized into particular 2rels between arbitrary mss x, y and the number of these abstracted 2rels should be grow up slowly [53].

This means that thesauri can be regularly enriched by assembling new mss using a rather stable set of 2-place rels.

1.2. We acquire mss by assembling the already learned rels into mss equal to those of communities accompanying the acquired mss by learning cms assigned to these mss in communities.

Assembling itself is guided by expressions of cms provided to the learners either from hand to hand explicitly by teachers or estranged, by some records of these cms that, in turn, have already been learned.

Cms, in fact, provide guiding instructions for assembling mss. This process is similar to compiling from programming languages into computer codes.

For example, OOP compilers do that addressing only to have/be/do rels between abstract classes. Likewise, to compile mentals hbd rels should be expanded to any rels of NL that can be embedded.

If some guiding instructions of cms on assembling target msssome rels, in fact, could be unknown to the learners, thus, teaching of these rels is needed and , preferably, hand to hand.

Let us remind also that mentals extend abstract classes and their storages, and why they can be designed similarly to the libraries in OOP, say, in Java.

1.3. The above observations comprise the following statement and corollaries: **Ass1.** *Mentals can be represented as systems of do classifiers of the types of one-, two- place rels.*

Clr1.1. *Learning mss, as a rule, is reduced to* ***assembling*** *the already learned mentals and 2rels into mss equal tothose of communities and this assembling is supervised , explicitly or not,by instructions provided by cms of these mss .*

Clr2.1. *Storages of thesauri Th of mentals induced by the libraries of OO abstract classes can, in principle, be organized as storages for representing these mentals systems of do classifiers of the types of one and two place rels.*

1.3.1. Since the above assumption does not refer to a particular representation of 1-/2-place rels they can, particularly, be realized by a certain module ANN, or a unit ANN (*unn*), thus, allowing to state that

Ass2. *Mentals, in principle, can be adequately modeled by ANN that, at first, represent 1-/2-place rels of mentals equal to unit ANN, then, compose these units into ANN equal to the basic constituents of mentals and, finally, unite these constituents into ANN equal to the target mentals.*

To make the above reduction not only existing, in principle but also constructive, the storage and linkage of the unit ANN into the target compound ANN should be refined.

1.3.2. Linkages of mentals are based on the recalling of IDs of mentals, similar to OOP,

In general, a way for assuring the linkages in ANN could be the modeling of those in OOP.

Another more realistic approach can be based on modeling linkages in NN, where the rels between neurons of the same or different layers are assumingly represented by synapses while the number of those synaptic links of NN of any human concurs with the number of particles in the Universe.

To model rels of NN based on synapses recall our aforementioned approach to formation of rels where, first, the rels *Arels1 B* are formed between the personalized classifiers A and B of unique realities, then, using these ground rels, the rels *x rels1 y* are formed between the variables *x,y*, of any classifiers to represent the rels themselves.

The above allows assuming that rels in NN can be represented as follows

Ass3. *Given A and B NN classifiers the rels1 between A and B can be formed as a new NN classifier A rels1B inked by synapses, necessarily, to NN classifiers of A, B and, expectedly, to other ones.*

This assumption, assumingly, can be a hint to the models of mental classifying, either of the types of mentals or ANN, that like to children, will incrementally enhance the complexity of learned rels starting with the units of the types of conditional reflexes, then, step by step uniting them into more and more compound classifiers.

3. Let's now focus on other aspects of acquiring mss.

3.1. Outlining the second dimension of mental learning, **the acquisition of mentals**, let us remind, that although inductors are important for communities *C* in revealing, discovering new classifiers the majority of classifiers are transferred from generations to generations of their members from hand to hand assuming the abilities to explain, teach, tutor for knowledge able members of *C*, experts, while for students the abilities, at least, to acquire those classifiers represented mainly by texts in their languages.

3.2. Thus, at least, two types of acquisition of classifiers *Cl* of *C* need to be analyzed

- from *hand to hand ,* when Cl is only the personal expertise of membersof *C,* and
- *estranged*, when *Cl* are estranged from members of *C* by certain records and can be learned by others if certain keys of C were preliminarily acquired fromehand to hand.

Those records can be thesauri represented by texts in languages of communities, while keys can comprise alphabets and basics of those languages.

3.3. Constructive models of estranged acquisition of human classifiers present translators, interpreters from OO programs allowing the classifiers *Cl* of abstract classes represented, for example, in Java, to correspond to equal *Cl* classifiers in computer codes.

3.3.1. Recalling that mentals extend abstract classes, the OO models of estranged acquisition of OO classifiers can be extended to similar ones to acquire classifiers of mentals, which *wrt* OO programs extend *attributing, parenting* and *doing* rels to all those rels that can be formed, particularly inductively, pursuing to cover as many rels of natural languages as can be modeled.

3.3.2. Note, that the models of natural languages (NL) like UNL, although similar to mentals, tend to extend OO rels and represent all rels of NL, operate, in fact, only with the IDs of mental classifiers of humans already represented in NL, thus, being, at least, restricted in adequate modeling of mental doings [2].

3.4. The stages of estranged acquisition of human thesauri, presented ideally by the encyclopedic Wikipedia texts, in our vision, should, apparently, include the transformation of these texts into an isomorphic composition of clauses and the transition from these clauses to the equally representing them mentals

Various heuristics will inevitably be involved in these acquisitions, so the question arises of scaling the acquisition quality for proper choices.

The approach to making that choice within the framework of RGT Solvers could be as follows.

3.4.1. RGT iSolvers personalized by their acquisition algorithms iAlgs learn Wiki and form thesauri iTzs.

Search of the best *Solvers can be arranged by tournaments between iSolvers, particularly in ways similar to those in [9], namely:

-the diversity of all possible learning iAlgs can be enumerated

- by tournaments similar to those in [9], an exhaustive wrt all sequences of proper bundles of iSolvers can be arranged to compete for the best RGT *Solvers wrt to a comprehensive diversity of RGT problems

-as proved in[9], these competitions will converge to the best RGT *Solvers.

3.4.2. As a step of the above project, the idea can be examined preliminary for chess Solvers, where Wiki will be reduced to a Chess Repository (CR) of [19], expanded with chess books (CR+).

Based on the experience of acquiring chess classifiers [18] chess iSolvers learning then competing by the [9] tournaments are guaranteed to converge to the best in the class of chess Solvers.

8. Revelation of Mental Systems

1. While we acquire the vast majority of mss from communities, it is reasonable to assume that any mss was sometimes revealed, discovered and developed then why construction of cognizers inevitably shoud address to the question of the first appearance, revelation of mss in communities.

Particularly, it should be refined a revelation of the type of mss , algorithms that include algorithms of inductive, deductive, imaginary and intuitive inferring of mss, processing mss to search or prognosticate classifiers and strategies, enhancement of effectiveness of mss by constructive regularization, constructive and adequate modeling.

While the appearance, discovery of each of the above algorithms needs comprehensive enlightening in what follows, we only outline a way of revelation of algorithms of the type of strategies.

2.1. Recalling once more that

- algorithms in Markov' mode are compositions of s1/2rels and equal to any other modes of algorithms

-thesauri of mss are equally represented by colored graphs,

then assuming that

Ass. For certain meanings with roots r (possibly in any of them), mental game trees with the same roots r can be extracted, where terminal nodes are certain utilities, it can be stated that

St. *Inferences, strategies in the mental trees that deliver certain terminal utilities starting from roots r , determine algorithms of the type of strategies delivering , at least, the min of these utilities for given r.*

2.2. The search for these strategies/inferences can follow the ideas of TZT and PPIT [51], followed by discussed ideas of transition from inferences to algorithms as well as visible transition from strategies to Markov algorithms.

3. Note also a case of formation of mss caused by a failore of matching of some realities r to mss of ad hoc thesauri Th. In that case a new mss matching r can be composed from those of rels of mss of ad hoc Th that r was already matching.

9. Questioning the Perfectness of Classifying

1. First recall the assumption that classifier sare always accompanied by certain explicit or implicit goals representing those guiding their formation why in modeling or processing classifiers it is inevitable not to be aware of these goals.

Then, all the goals are "rooted", i.e., the roots induce various classifiers to govern the doings including genomic goals *Gn* classifying evolutionary approved utilities *wrt* the roots and classifiers/goals *Gl* learned in the life time.

1.1. Ideally, it could be assumed that all classifiers were formed in the genomic spaces of humans and *wrt* the genomic goals *Gn* that are highly identical for all humans.

In other words, we assume that *perfect* classifiers *wrt gi* or *gi perfect* are classifiers *Clgi* correctly classifying all prints of SPgn wrt some *gi* from *Gn* , while *Gn perfect* are those correctly classifying all prints of SPgn wrt to any *gi* from *Gn*. In anther words, all positives of perfect classifiers *Clgi* are certain utilities *wrt* to genomic goals gi.

2. The assumption of perfect classifiers induce a view on a "paradise" of humans as follows

Ass1. *Ideally, if Gn perfect classifiers were available with proper complexity of identification of positives of +Clgi, then humans, and assumingly, cellulars too, would do perfectly wrt their genomic goals, thus, having a high degree stability of their being in the stable Universe.*

Accepting the assumption, we need to answer whether Gn perfect classification can be achieved, while, apparently, we need to preliminarily refine whether do exist gi perfect classifiers for some gi@Gn.

3. The existence of gi perfect classifiers for some gi can be argued as follows.

We assume that Sweets, Soars, Hots, etc. are examples of perfect classifiers or their acceptable approximations, since some evolutionary utilized realities by a few attributes can be uniquely identified by all humans due to their high genomic equality.

Then, the innate invariance of outputs of sensors of humans to space transformations, for example, positioning, scaling, spinning or the ability to be activated only by topological invariants, makes it possible to form almost perfect classifiers of some types of realities *wrt* to *Gn*.

Seemingly, they are classifiers accumulated in the right hemispheres and are responsible for humans and to a various extents of cellular innate emotional, communication, survival and other doings.

4. Now, let us argue that the entire perfect classification of SPgn is intractable.

Recall, first, that perfect classification, in general, is enormously hard work since the classifiers should be identified among the possible 2^2^n ones.

NP-completeness of proving the consistency of classifiers *wrt* certain gi seems to be similar to proving that propositional formulas are tautologies

4.1. Then, following Kolmogorov, we believe that the units of the Universe highly vary in their complexity and include extremely complicated ones.
And humans have been trying for a long time to represent the entire Universe as comprised, maximum unitary and plain, therefore, more manageable classifiers are not yet successful.

At present, these attempts are resulted in revealing only several perfect classifiers, while they are mainly represented by incrementally enhancing mosaic of systems of imperfect classifiers, where classifiers of some layers are composed of *ad hoc* ones of the lower layers with various rels between them.

4.2. Each classifier must identify +*Clgi* in a set of, at least, 2^n percepts, while their formation is inevitably based on the restricted expertise provided either by the given mpgi matrices or by sets of similarly formed classifiers.

Apparently, restricted matrices, as a rule, don't represent target classifiers.

4.3. Then, the classifiers either perfect or not, should be compact enough for the tractable storage and processing time complexities that excludes their storage barely by mpgi matrices even in the case they are correct. In turn, the compactness of original matrices causes the loss of their details, therefore, the inclusion in *+Clgi* doubtful positives, followed by the rise of the mistaken.

Inductive formation or inference of classifiers based on the compactors of *mpgi* matrices, inductors [9], regularly causes imperfection, since compression inevitably expands mpgi by non examined positives that with any consequent formation of classifiers, rise the risk ofincorrectness.

For example, a type of inductors to compact mpgi matrices represents them by equal propositional formulae *Clgi* of the types of conjunctive of disjunctive forms composed by logical operators from the attributes.

These attributes, in turn, are compressed into possibly min sets preserving yet the distinction between +*Clgi* and the opposite prints of the initial mpgi.

Because of unexamined expansion, *Clgi* can identify positives as unfavorable, i.e., no utilities, therefore for *gi* in the formation of consequent classifiers *Clgk wrt Clgi*, i.e., for *gk = Clgi,* the unexamined expanded positives of Clgj will enhance the incorrectness of consequent classifiers.

4.4. The above argumentation allows us to draw the following conclusion
Ass1. *The disclosure of entire perfect classifiers is an intractable problem.*
5. Note, that intractability of perfect classification is similar to that in combinatorial games, for example, in chess, where perfect classifiers of winning, losing and drawing positions could provide perfect chess players. In fact, the construction of such classifiers, in principle, is impossible, due the combinatorial nature of positions, what means that classifiers of winning, losing and drawing positions with acceptable complexity do not exist, while classifiers based on memorization of positions of the classed do not exist as well , since the total number of those positions, say, in chess, exceeds the number of all elementary units of our Universe.

6. Imperfectness of the models of human classifiers is caused both by the abovementioned sources of imperfection of human classifiers themselves and those caused by malicious modeling.

6.1. The means of coping with the first type of imperfection are based on the inductive inference of classifiers by their consisting of regularly enriched matrices representing these classifiers.

Indeed, inductive inferences of classifiers are always based on the restricted *mpgi* matrices, therefore, at each step of applying the inductors, the perfectness of the hypothesis on target classifiers cannot be guaranteed.

Nevertheless, with the enrichment of these matrices with new positives of these classifiers the hypothesis is regularly improved to be consistent with the input matrices and is correctly classifying until failing for some new entry to be modified.

Summary of Chapter 4/1

1. *Mentals* are interpreted by the basics of attributive classification of theoretical computer science. Referring to some relevant models of classification, an approach was provided to inductive formation of two-place relationships, and guides are induced to the formation of mentals, particularly, in the artificial neuron net (ANN) modes.

2. Learning by acquisition of systemic classifiers and the ways of coping with imperfect classification are discussed.

We state that learning of mentals is reduced to composing the already available mentals and by means of adapting inductors to the learning of doins.

3. It was argued that not only 1-place classifiers can be formed inductively, but similarl, at least, 2-place ones, i.e., rels between classified realities having corresponding IDs.

4. And it was stated that, although certain inductors are equally applicable to the formation of 1- and 2-place classifiers of mentals, as well as to ANN, the principles of organizing the storages of mentals allowing, particularly, to process net-based inferences are refined only for the mentals.

Thus, questions arise on how to organize ANN capable of performing all mental doings available to the mentals and equal to them?

5. Acknowledging that:

-mentals are systems over doins reducible to the 1-, 2- place relationships (rels) that can be formed inductively, then,

- 1/2 rels represent the types of classifiers, rules, regularities, while their systems represent, particularly, algorithms, abstract classes and abstracts,

- that mentals are adequate models of thesauri Th of mental systems (mss)

-thesauri Th of mss can be adequately modeled by colored netsTh decomposable to , say, abstracts or 1-/2 rels,

the question arises whether netsTh can be adequately modeled by computers, particularly, be represented by the memory of computers in a way that any mental doings processed in the netsTh could be equally reproduced in computers.

6. The primary role of classifiers in ordinary, common life is in distinguishing, recognizing utilities, and it can be acknowledged that inductively inferred classifiers manage utilities with a proper reliability.

So, by imperfect classification of the Universe, humans, nevertheless, progress in doings is perfect.

7. Imperfection of inductively or deductively learned classifiers is caused, particularly, by erroneous chains arising in the process of compacting via generalization of the matrices mpgi that only partly represent the target goals Clg.

The ways of advanced usage and managment of imperfect classifiers include regularization of classifiers constructively or not, their adequate modeling and tending to be consistent with relevant theories

Then the sets of attributes necessary to preserve classifications of the given mpgi matrices seem might be minimized to make the matrices max compressed.

Chapter 5/1: Specifying the Origination of Cognizers

1. Cognizers: Questioning the Origination

1. Interpreting cognitive doings, so far we have assume that cognizing is doings on learning and organizing mss to support the promotion of predetermined by communities utilities, including inherited roots and those induced by the roots , while learning and organizing mss are mostly reduced to revelation and acquisition of mss accumulated in communities and, in essence, guided, supervised by them.

1.1. Revelation of mss, in turn, is assumed to be goal oriented, thus, motivated, and includes doings of inductive, deductive, imaginary and intuitive inferring of mss, enhancement of effectiveness of mss, processing mss to search or prognosticate classifiers and strategies. In turn, the effectiveness of mss can be raised by cellular or constructive regularizing, constructive and adequate modeling, etc.

Acquisition of mss includes learning from teachers, texts, other presentations of cms.

1.2. Cognizers we have defined as compositions of realities capable of cognizing within of the framework of communities of humans or other cognizing realities and including mental systems (mss), including cognitive algorithms, and octaves of mss, including controllers, sensors, effectors, storages of imprints and energizers, i.e., means to supply energy.

2. Then, we have acknowledged the fundamental question of AI on whether it is possible to specify root cognizers, rcogsai so that mss formed by rcogsai could attain the highest cognizers of humans Cogs, in fact, not yet completely resolved.

Nevertheless, certain progress can still be stated in specifying such rcogsai.
Particularly, statements presented in preceding Chapters let us assume that root cognizers rcogsai include certain inductors that incrementally construct 1-/2- place rels by generalizing matrices of classified imprints that were preliminarily extracted from the stores of prints by the algorithms gcasers and ggcasers.

These accumulated rels and their systems comprise nets of mss representing realities in different modes. The modes can be, for example, literal, personal or abstract, say topological, that at the next steps of development of mss can be generalized or differentiated, correspondingly, by certain inductors.

It can be also assumed that certain algorithms of rC^* regularly develop chains of mss with incrementally rising complexity.

These chains could start with case-based rels represented by matrices of imprints followed by rule based rels generalizing those matrices followed by their assembles into mss representing, particularly, algorithms, say, inductors.

3. The above findings, progressing outcomes of AI in modeling cells and constructing root cognizers rC* bring on optimistic expectations in constructing rC* comprised as the assumption AI-Ass. *The highest human cognizers Cogs can construct root cognizers in both modes based on the models of cells, i.e., rcogsaic, and not based on these models, i.e., rcogsai, that developing can attain CogsAIc and CogsAI, correspondingly, functionally, at least, equal to Cogs.*

3.1. The immediate consequence of the assumption is the corollary

AI-Crl1. *CogsAIc (CogsAI) being equal to Cogs will inherit from Cogs the ability to reproduce themselves, i.e., to construct cognizers equal to themselves, particularly by constructing rcogsAIc' (rcogsAI') developing to CogsAIc'(CogsAI') equal to CogsAIc (CogsAI), thus, equal to Cogs.*

3.2. Then, since rcogsai can be constructed from ad hoc available not cellular constituents that are unbeatably less complex than even uncials, the reasons and chances in favor of origination in Nature root cognizers rcogsntr equal to rcogsai become unbeatably higher than the chances for origination of unicials.

The above reasoning let comprise the following corollary:

AI-Crl2. *Root cognisers rcogsntr consistently with the basics of ad hoc physics can, in principle, be originated in Nature then develop themselves to cogsNtr equal to Cogs.*

3.3. Premises to the corollary are also our findings that

- constituents of rcogsai are constructed and, so far, are either reduced or seem to be reducible to certain basic classifiers, namely to 1-/2-place rels

-1-/2-rels are not as extremely complex as the cells, their nature highly correlates with the origin of negentropicity [6] and information [59], one of targets of contemporary physics [62]

-meaningful compositions of 1-/2-rels, say, inductors, expectably can be constructively chained, assembled within the framework of basic laws of physics

3.4. Corollary AI-Crl2, in fact, states that realities not only of a cellular nature but also those that can be constructed, assembled from ordinary available to humans' units of matter can attain the highest cognizing of the Universe, at least, comparable with that of humans.

And since these constructive highest cognizers are only the assembles of units of matter unconstrained from the mysteries of the origin of cells, it would be expected to question the necessity of humans or some ones else in assembling, constructing the highest cognizers.

In other words, to question the feasibility of origin of the highest cognizers within the framework of acknowledged physics.

3.5. Finally, uniting the above two corollaries, a scenario follows, where the originated root rcogsntr, developing to cogsNtr, capableof reproducing themselves, could find it reasonable to reproduce themselves in a cellular way, namely, to construct cells with a known nova days functionality, and would let them evolve as we are now observing.

AI-Clr3. *Originated in Nature cognizers cogsNtr by analogy with Cogs have to be able to reproduce themselves in various modes, particularly, in a cellular mode based on the construction of cellular root cognizers rcogsAI.*

4. The above corollaries can be comprised as the problem of origination of cognizers rG* in Nature, or **the problem of the Origin of rC *** questioning *whether rC* can be grounded or can rC* be originated* within the framework of *the fundamental laws of the Universe?*

In other words, the feasibility of origin of the highest cognizers is questioned within the framework of acknowledged physics.

4.1. Let's emphasize that the fundamental consequence of solving the **Origin of rC *** would be in providing significant arguments that the kernel of effective cognition is one of the general universal means for sustainable being in the Universe and is not the privilege of cellulars only.

5. Feasibility of origination of rC* in Nature can be effectively argued by their adequate constructive modeling.

To approach the modeling, it is reasonable

- to account the constituents of cognizers
- to reduce the constituents to the inevitable for cognizing basic, root cognizers rC* necessary to develop themselves to the higher levels of cognizing as much effectively and efficiently as it is possible
- to ground the development of rC* to the advanced cognizing
- to chain the most general forms of existence of matter to the acceptable by physicists transition to the roots of cognizers rC*.

2. Revealing Roots of Cognizers

1. Accounting constituents of cognizers.

At first, let us remind that cognizers rC* tending to adequacy with human ones should necessarily be analyzed in frame of within the framework of certain communities.

Then, as it was assumed that the constituents of rC*, particularly, the providers of matrices of imprints, in fact, one of the primary classifiers, inductors generalizing these matrices, assemblers of mss and enhancers of effectiveness of mss should be based on the inevitably necessary octaves of cognizers.

Those octaves, first of all, should include the kernel of negentropicity, i.e., energizers with their own sensors of energy, storages and savers of energy, controllers of energizers to coordinate their doings, particularly, to preserve them.

Then, octaves, should include sensors, effectors, savers of imprints, storages of imprints and systems of classifiers enriching and developing those of energizers to become the base of mental systems, in general, an instrument to support negentropicity.

2. As follows from this, the constituents of **cognizers**, in general, can be interpreted as doers of the types of classifiers and corresponders, as a type of corresponding doers, algorithms, and as storages of imprints or energy.

What is listed underneath are categorized as assumingly inevitable constituents of cognizers:

--sensors/doer classifiers,

--effectors/doer corresponders

--savers of energy, imprints/effectors,

--energizers: sensors of energy/ doer classifiers,savers of energy/effectors,storages of energy, controllers of energizers/doer corresponders, effectors,

-- storages of imprints,

-- mss controllers, or +controllers/algorithms,

--algorithms for revelation of mss, including

--- algorithms for extraction of case-based classifiers/rels as matrices of imprints, i.e., g- and gg-casers

---algorithms, inductors, generalizing case-based classifiers into rule-based ones,

--- algorithms for explaining and interpreting mss

---algorithms for planning in nets N1/2rels, enriching plans to strategies, then from strategies to case-/rule-based algorithms

---storages of mss

--- algorithms for enhancing the effectiveness of mss, particularly, regularizers, constructive and adequate modelers and reorganizers of mss

--algorithms for acquisition mss from communities.

3. The above categorization of constituents of cognizing let's question the reduction of categories of constituents, namely, algorithms, doers, storages of imprints, mss and energy to corresponding them basic units contained, we assume, in the roots of cognizers.

And what follows is a version of that reduction.

3.1.1. Since it was argued that algorithms can be represented as compositions of 1-/2- rels the revelation of algorithms is reduced to the revelation of classifiers of the types of 1-/2- rels and composers of 1/2rels into algorithms.

3.1.2. In turn, as it was already argued 1-/2-rels can be learned to generalize g/gg-matrices of imprints by corresponding inductors .

Thus, the *revelation of algorithms is reduced* to the revelation of primary classifiers -*g/gg-matrices,* revelation of *primary inductors* to transit to 1/2 rels and *composers of 1/2 rels into algorithms*.

3.2. To reduce *doers* to the roots, let us recall that, as we assume, they are, in general, realities having input/ output parts and for available *inrealities,* i.e., for realities, and moreover, for somewhat at the input parts either elaborate certain output realities or remain passive.

3.2.1. Comparing with Markov algorithms, we can state that

- doers are not necessarily deterministic

-in/out-domains of doers are not necessarily outids or symbolic, correspondingly,

- elaboration of outputs is not necessarily processed symbolically.

Along with the aforementioned, we assume, that doers are compose of not necessarily deterministic 1-/2-rels, are processed (sensors) over and output (effectors) not necessarily symbolic realities, while 1/2 rels are symbolic.

This assumption can be supported by the following premises.

3.2.1.1. Recaling the sensory-motoricalal, operational and abstraction stages of human mental development according to Piaget [4] it seems correct to gravitate doers to the first of these stages, while the algorithms refer to the operational and abstract stages.

3.2.1.2. In the interpretation of not fully symbolic doers it is also relevant to recall, particularly, mechanisms, machines, robots and autonomous agents.

For example, mechanisms by Wikipedia are devices that transform input forces and movements into a desired set of output forces and movements, while mechanisms generally consist of moving components and can internally use algorithmic controllers.

3.2.2. Similarly, with the reduction of revelation of 1/2rels to the revelation of g/gg-matrices and primarily symbolic inductors, it would be reasonable to assume that the *revelation of counterparts of 1/2rels in doers can be reduced to the revelation of certain* *matrices of not necessarily symbolic realities and some analogs of inductors of not necessarily symbolic nature.*

3.3. Storages of imprints, mss or energy address to memorization of either symbolic or energy types of realities.

3.3.1. *Memorization of imprints is reduced to memorization of IDs of classifiers, while for mss along with memorization of IDs of mss m, it would be required, in general, to memorize the bundles of 1/2rels associated with these IDs indicating, in general, IDs of mss linked with m.*

3.3.2. Assumingly, root memorizers of symbols can be analogous to those in computers or in recorders of visual and sound realities.

Root storages of energy, in turn, can be reduced to some types of accumulators of various types of energies.

3.4. Summarizing, we state that categorization of constituents of cognizers allows to reduce them to uniform units, some alphabet seemingly inevitable in assembling these constituents, therefore, assumingly incorporated into the roots of cognizers.

Doings of these uniform units can be split into two groups as follows:

-*octaves of mss,* including

--*energizers* of cognizers with their constituents inevitable for energy supply and

--means inevitable for appearance and doing of mss including +sensors, +effectors, classifiers, possibly embedded or inherited, storages of imprints and 1/2 rels comprising the base for learning and organizing mental systems,

-*mental systems* themselves, including

--means of revelation of g/gg- matrices of imprints or other realities of not necessarily symbolic types

--means of transition from g/gg matrices to 1/2 rels based representations

-means of composing rels into algorithms and doers

-means of memorizing and organizing mss, particularly, mss of the types of controllers.

While case based rels represented by g/gg matrices already seems to be sufficient to look for algorithms and doers in spaces of those rels, the expectedly enormous sizes of such matrices would restrict, if not make impossible, their regular processing and composition into

primary inductors not necessary over symbolic realities

That is why we assume that algorithms need to be searched not in case- but in rule-based spaces of rels, thus, preliminarily, the roots, namely, mss or, possibly, octaves, should include certain means, say, primary inductors, of transition from case- to rule-based rels.

3. Constituents of Enhancement of Cognizing

1. Let us recall, first, that within the framework of certain units of cognizers, we have already started to specify both the algorithms of transition from ad hoc constituents of cognizing to the new advanced ones and the algorithms providing the required lines of cognizing.

Namely, we have specified

- inductors that at any level of expansion of thesauri let transit from matrices of imprints to 1/2 rels representing communicators of mss and incrementally generalized classifiers

- algorithms of searching chains of rels that let in spaces of 1/2 rels to form, at least, strategy-type algorithms

- assemblers that given storages of 1/2 rels let form mss representing, particularly, the algorithms themselves.

Then algorithms for explaining and interpreting mss are specified, as well as the models of communicating in languages approaching to the natural ones and the ways of arguing their adequacy are specified.

These algorithms, in fact, can preserve the acquisition of mss in communities

2. Let us now continue to question the transitions from ad hoc constituents of mss to the new higher ones for the rest of cognizing algorithms and try to outline possible scenarios.

Inevitable constituents of mss, their reduction to the roots and formation			
		reduction of mss to the roots and their learning over given octaves	
constituents of mss	roots* =>		roots** =>
storages of mss			by memorization of mIDs of mss m and associated with mIDs bundles of 1/2rels indicating, in general, IDs of mss linked with m
mss controllers, i.e., +controllers/ revelers of mss, including: -extractors of case-based classifiers/rels as matrices of imprints, i.e., g- and gg -casers -inductors, generalizing case-based classifiers into rule-based ones -explainers and interpreters of mss -planners in the nets of 1/2rels, enricher of plans to strategies, then from strategies to case-/rule-based algorithms -enhancers of effectiveness of mss, particularly, regularizers, constructive and -adequate modelers and reorganizers of mss learners of mss from communities.	Algorithms	are reduced to revelation of : classifiers of the types of 1-/2- rels and composers of Markov algorithms by those 1/2rels	1. In paragraph 3/5 are outlined either referred to the views on and advances of the learners of mss including : inductors, transferors from case based classifiers to the rule based ones, explainers and infers of mss, planners and searchers of strategies, explainers and deducers of mss, intuitive infers, constructors, regularizers and modelers of classifiers, particularly, case based ones, assemblers of mss and controllers. 2. 1/2 rels can be learned spirally , step by step - 1st -circle: starting from matrices of imprints, i.e., from case1/2 rels then -generalizing matrices to the rule based 1/2rels by first level inductors , 1/inds , that just chose some conjunctions attributes of the matrices -enriching attributes by new 1/2rels composing second level 2inds by searching strategies in spaces of 2rels, -2nd -circle …… 3. Inductors can be composed from 1/2rels
mss	systems of algorithms		

Fig.18. On inevitable constituents of mss, their reduction to the roots and formation.

2.1. Representation of case-based classifiers by matrices of imprints v requires certain sets of attributes for presenting the imprints. So, algorithms of transition from $\boldsymbol{v}$ to the corresponding rule-based classifiers could be suggested that mimicking the formation of DNF [55], are uniting some subsets of attributes into bundles by AND rels, which, in turn, by OR rels unite them in the units aimed to distinguish the positives of original matrices from the other imprints.

2.2. Algorithms for explaining inferring mss, planning and searching strategies are based on chaining of 2rels in spaces of 1/2rels aimed to select some types of either chains or compositions of chains of 2rels. Thus, those algorithms could be constructed as uniformly chaining and searching compositions of chains algorithms that convene the given criteria.

Class of those algorithms include, we assume, algorithms for logical inference, search of trajectories in graphs or strategies in game trees.

2.3. Regularizing algorithms could address the constituents of classifiers and try to regularize each of them, i.e., try to provide sub algorithms capable to produce samples matching these constituents regularly, followed by the composition of subalgorithms into the target regularizers.

2.4. Classifiers can be modeled by seeking their regularized equal versions either by regularizing the classifiers themselves or by searching already regularized classifiers that max approximate the models of the targets.

This search can be processed by looking for max coincidence of constituents between the targets and other classifiers.

2.5. Regularizing and modeling algorithms enhance the effectiveness of mss.
Their development to ones for adequate modeling is seen in detailing the requirements of adequacy and their gradual attaining, first, for the constituents of classifiers, and then, for the entire ones.

A productive hint to these algorithms could provide a detailed analysis of examples of the successive complete transition from commonly given classifiers to their adequate models, say, as it was attained for plains, motor vehicles, algorithms.

2.6. Intuitive findings of some hints or ideas can be modeled, we assume, by searching in the storages of imprints where the requests could be formed as some important fragments of mss representing the problems, then, are searched in the storages of imprints for ones that by values of the request attributes intersect with ones in the requests.

2.7. Algorithms of extraction of case-based classifiers/rels represented by matrices of imprints seem to have to start with some target goals g/gg, then bring together imprints where g/gg are attained , in parallel, contrasting them to imprints where g/gg do not occur.

2.8. Hints on constructing the controllers seems reasonable to be searched by analyzing of the design of existing controllers, for example, those of cellulars, operational systems of computing both centralized and distributed.

4. Grounding Transition from Matter to Roots of Cognizers

1. So far, the roots of cognizers rC* include, we assume,
-octaves with inevitably necessary energizers and means necessary for doings of mss including +sensors, storages of imprints and classifiers, +effectors and +controllers,
-mental systems themselves.

Then, we have reduced the revelation of constituents of cognizers to the revelation of classifiers of the types of sensors, doers composing classifiers into effectors, particularly, of the types of savers of energy, imprints or classifiers, as well as to the revelation of storages of energy, imprints and systems of classifiers.

Finally, similar with a reduction of the revelation of 1/2 rels to the revelation of g/gg-matrices and primarily symbolic inductors, we assumed, that the revelation of counterparts of 1/2 rels in sensors and effectors can be reduced to the revelation of certain g/gg matrices of not necessary symbolic realities and to some analogs of primary inductors of not necessary symbolic nature.

Thus, the question of grounding roots of cognizers to matter is reduced to the questions of origination in nature of a variety types of sensors, effectors, storages of energy and imprints, g/gg casers forming primary classifiers of the types of g/gg matrices and composers of classifiers into effectors.

While the core of answers to above questions have to be expected from findings of physicists followed by their proves by constructive adequate modeling, let us provide some premises to grounding energizers, octaves of mss and mss.

Inevitable constituents of octaves / energizers , their reduction to the roots and formation		
constituents of energizers	reduction to the roots and their appearance	
sensors of energy	1. Energizers, are assumed, can originate in nature, what means their roots are, in fact,	
effectors to approach to the sensed energy	the appearances of matter why the question of their transition to energizers coincides with ones of transition from matter to life , negentropics and, assumingly, to information, or to classifiers	
controllers of energizers	2. Paragraph 4/5 outlines the ways of origination of constituents of energizers including one , based on the one based on the energy of the warmth as follows: Sensors of the warmth: all realities , as a rule, can be the sensors of warmth	
storages of energy	Storages of the warmth: some realities can save energy, say as sugars, based on chemical interactions and not necessary requiring the involvement of cellulars. Means preserving the abilities of energizers to sense, approach to the sources of warmth and store it can be arranged by effectors and controllers of the energizers.	
constituents of octaves over the energizers	reduction to the roots and their learning	
	roots* =>	roots** => roots***
+sensors of all over +effectors for all over +controllers for all over	doer classifiers doer corresponders	comprised from not necessarily deterministic 1/2 -place rels , processed over (the sensors) and output (the effectors) not necessarily symbolic realities, formed analogously with ones of mss
storages of imprints	storages	by memorization of IDs of attributes

Fig.19. On inevitable constituents of octaves over energizers, their reduction to the roots and formation.

2. The origination of energizers, apparently, is reduced, if not coincide, to the fundamental question of transition from matter to life, pioneered by the ideas of Schrödinger attributing living, first of all, by their ability to be not entropic, negentropic, in contrast with the vast majority of other entropic realities [6].

Within the frame of these ideas, let us provide a scenario of origination of negentropics, thus, the energizers of cognizers.

2.1. Let's define *durables* as realities that in contrast with others, *temporal*s, are something, that can be properly identified in the mean time.

For example*, refers* are durables causing certain outputs from certain types of realities. A type of refers, sensors, can cause imprints from warmth, light, sound, chemicals, etc.
Other durables, *absorbers,* let be stable in absorbing liquids and some types of chemicals.
Realities synthesizing sugars from carbon dioxide and water are types of *energy saving* durables while sugars, in fact, are types of storages of light, warmth energy.

2.2. Durables can be *static*, like stones, rocks or solar systems, and *dynamic*, like rivers, oceans or star systems.

2.3. *Active* durables do by themselves for durability, in contrast with *passive* ones that are only the objects of doings of others.

2.3.1. Note, that durables, seem to be tightly related to the concept of *strings* in cosmology [33] and even can be reduced to them .

2.4. *Root negentropics*, rNg, are active durables involving refers, storages and energy savers along with consumers of stored energy aimed at preserving doings of rNg, while *negentropics, negs,* are realities containing the root ones, rNg.

2.5. While cellulars are a type of negs, cognizers, as they were defined, are negs and some of them can be constructed artificially, say, as autonomous agents, whether the negs can be originated in Nature?

2.5.1. Compared to the question of the transition from matter to life [62], the question on the origination of negs appears to be less complicated than seems more tractable.

The chain of premises on the origination of negs based on the warmth type of energy, can be sketched as follows.

2.5.2. Let's, first of all, state that realities, as a rule, are sensors of warmth since having atomic structures, atoms with changes in surrounding warmth change their activation causing corresponding changes in realities that can be indicated, for example, by their temperature.
Some realities seem to be able to save energy, say as sugars, based on chemical interactions that, we assume, not necessary requires the involvement of cellulars.

In search of ideas on how these realities could actively preserve their abilities to sense, save and store the warmth let us note that even if those realities don't actively preserve their energizers, nevertheless, they could be supplied by warmth since the sources of warmth can be active and permanently or periodically supply realities with warmth directly.

For example, plants, insects, reptilians use the periodicity of returning of the Sun to the areas of their residence for direct consumption of warmth.

2.5.3. Note, that questioning of origination of negs is tightly related to the questioning of origination of information [59], which in our interpretation is equal to questioning of origination of some types of classifiers.

In our approach negs are classified as systems with certain inevitable constituents, thus, requiring to question the origination of each of them.

3. Questioning the origination and formation of constituents of octaves of mss of the types of +sensors, storages of imprints and classifiers, +effectors and +controllers, then, the formation of mss let us refer to their already discussed reduction to different types of classifiers.

3.1. Nova days studying let assume that primary classifiers in form of g/gg matrices can be originated in Nature within the frame of ad hoc laws of physics.

Indeed, classifying is inseparable from gaining "information" by Shannon [15] since we get "information" when we resolve certain diversities of possible options selecting, identifying, classifying some of the options.

In turn, the coincidence of categories of information by Shannon and anti entropicity, or negentropicity, in thermodynamics up to the coincidence of their measures cannot be accidental.

And since nova days studies [59] argue that information, thus, negentropicity can be originated in Nature it follows that g/gg matrices - classifiers can be also originated in nature following the laws of physics.

3.2. Some hints on the origination of classifiers of cognizers can be gained from psychology and biology.

Following Piaget [4], the steps of mental development are universal and hierarchic, therefore, in constructive modeling one can orient in the sequencing of these steps.

3.3. The hypothesis on plurality of lines of evolution in [13] states that in parallel with the evolution of cellulars there are pieces of evidence on the unique line of evolution of viruses, while their appearance remains much questioned. The commonality of lines of these evolutions can also provide hints on the inevitable steps of development.

5. On Modeling the Origination of Cognizers

1. Scenarios towards origination of cognizers inevitably need to illuminate the origination of primary classifiers and, first of all, the sensors.

The scenario could be based on the premises that interactions of realities if not destroying them could cause reciprocal imprints that, we assume, could be chained to the appearances of classifiers of the type of matrices of imprints, particularly, classifying sources of energy.

Restraining ourselves from the interpretation of physicists how the Big Bang explosion of energy is chained to the origination of primary forms of matter, let's provide a view on the origination of primary classifiers based on the properties of a type of realities, durables.

We believe that the origination of durables of the types of negs in Nature can be proved and argumentation, particularly in [59], can be interpreted as one of optimistic promises on the way to the proof.

2. Other promises may include those that havr already been sketched.

2.1. To begin with, we note that classifiers of energy types are, in fact, sensors of certain type of realities, namely, the energy.

While along with sensors of infrared or warmth and light ones typical of animals living under the Sun, there are life forms sensing and using other types of energy, at this stage we find the most promising to model origination of refer durables of the types of sensors of Sun energy, or warmth sensors.

2.2. All realities to different degrees are warmed under the Sun and at different times preserve, keep, accumulate, save their warmth.

Then, sensing lights and memorizing the corresponding imprints are also, as a rule, caused by sunlight.

So, it seems reasonable to assume that these warmth sensors, if originate, could resolve the first three of the above requirements to the energizers.

Apparently, it is worth looking for analogies in the theories of evolution of cellular, where these primary solar sensors play a key roles.

2.3. Then, the causers of classifiers of the types of g/gg matrices, as well as the inductors each time assume some given g/gg goals to form new classifiers wrt to these goals.

Apparently, questions arise on the first of these goals and if warmth sensors were originated, they could, in fact, be those primary goals.

3. Answering the question of how the stored classifiers can be self-organized into algorithms, it is worth addressing the ideas of self- organization of units of other types [63].

Actually, arises the following question:

whether following the laws of physics in the spaces of originated 1-/2-place rels, exhaustive search procedures can be originated that seeking chains from the ongoing roots-situations to the acknowledged utilities could regularly reveal algorithms, represented, for example, as conjunctions of rels.

3.1. We also find it promising to refer to the search models, where solvers of game represented problems evolutionary develop their effectiveness in competitions with other solvers.

A model of such development based on the provision of constructive chains of tractable local tournaments guaranteeing the attainment of the highest effectiveness of solves by competitions is provided in [42] and briefly outlined underneath.

4. Thus, the origination of cognizers and their constituents, seemingly, can be modeled by the following, in sketch, scenario:
-origination of energizers including, say, warmth sensors, savers and storages of warmth energy
-origination of primary classifiers of the octaves of mss

A scenario of modeling the origination of cognizers and their constituents:

- origination of energizers including, say, warmth sensors, savers and storages of warmth energy
- origination of primary classifiers of the octaves of mss.
- formation of g/gg matrices followed by their transfer to 1/2 rels by primary inductors that incrementally enrich the attributes of g/gg matrices.
- formation of storages of 1/2 rels inducing corresponding spaces where mental systems, particularly algorithms, can be formed by a variety of cognitive doings.
  formation of cognizers and their processing in parallel for the members of communities causing the prospects of enrichment of each other by communicating their discoveries.

Fig. 20. A scenario for medeling the origination of congnizers and their constituents.

-formation of g/gg matrices followed by their transfer to 1/2 rels by primary inductors that incrementally enrich the attributes of g/gg matrices
-formation of storages of 1/2 rels, inducing the corresponding spaces, where mental systems, particularly algorithms, can be formed by various cognitive doings.
- formation of cognizers and their parallel processing for the members of communities causing the prospects of enriching each other by communicating their discoveries.

5. Note, that NN neurons or their ANN models are capable to unite the steps of transition from matrices of imprints to the rule-based classifiers.

Indeed, while in the above reasoning, first the durable infers as attributes to form then case-based matrices, followed by, their transition to rule-based classifiers, the neurons, in fact, do them in parallel.

5.1. Namely, neurons can start with attributes- infers by a chance linked with input realities supplied either from screens or outputs of other neurons, and their initial weights can be chosen by chance.

Input samples in neuron learning can be provided sequentially, thus, without the requirement to provide g/gg matrices holistically.

The output neuron always generalizes the imprints, thus, there is no need for additional inductors.

Thus, it is worth finding ways to originate constructive neuron type classifiers with the above advantages.

6. Modeling Convergences to Paradise

1. In [42] it was proved that in classes A of algorithms letting toenumerate all possibleappearances of algorithms by enumerating the states of regulators determining any of their appearances, it is possible to arrange the sequences of tournaments between those algorithms similar to those by Elo in chess [19] that allow to converging the sequences to the best algorithms in A.

1.1. In general, the theorem provides a framework of the models of Adaptation for Being in the Universe for a variety of regulators, leverages of adaptation of algorithms.

What follows sketches the steps how within the framework of the kernel of models of cognizers to prove the *theorem on the convergence of cognizers to the Being in the Paradise* (TCBP), i.e., to a Happy Being in the Universe, arguing, in fact, that cognizers would be able to dream that happiness.

2. Outlining the proof of TCPB, *first* it is worth understanding how one of the pillars of cognizing, the acquisition of mental systems from communities, provided by certain cms (communicatives) and dreaming to be possible in natural languages, can be modeled in cognizers.

The next step of approaching the proof of TCBP could be the construction of algorithms of acquisition of thesauri of communities and ones enriching the thesauri, particularly by discovering, revealing new mental systems along with scrutinizing other leverages of cognizing to try to enrich the constructed algorithms.

Assuming that the class of algorithms A* containing the solution S* of TCPB along with leverages for producing their entire diversity are specified, we need to arrange a search of the best of them.

The former search [42] was based on the enumeration of all possible states of leverages and a technique for extracting all possible bundles b from A* of these algorithms, local tournaments between them guaranteeing the convergence of the winners of the tournaments to S*.
Instead of this ideal enumeration, a more realistic search can be based on the extraction of bundles b from A* by chance, guaranteeing the coverage of all b in A* (maybe by analogy with measuring areas by tossing lines on the area by chance).

7. Are Cellulars Constructed?

1. Expectations for positive answer to the question, we assume, follow from the following premises.

1.1. Physicists exclude the origin of cellulars by chance. Evolutionary and genomic theories tolerably explain how elementary cells could attain the highest Cogs of humans but they are helpless in explaining of the mysteries of universality and the highest complexity of procedures of genomic reproduction of cells.

The appearance of these procedures in Nature by chance is acknowledged by physicists as principally unfeasible.

1.2. *Feasibility of Creators.*

1.2.1. The mysteries of high complexity and universality of cells irresistibly provoke the vast majority of theories and religious beliefs to the conviction about the existence of Creators of cells, while that conviction inevitably questioning the mystery of the appearance of Creators themselves.

1.2.2. Another premise on Creators follows from the belief of Buddhists supported by their manuscripts in preceding us highly advanced lemuroids [35].

And it should not be excluded that these lemuroids have found perspective of creatimg new additional to them carriers of the roots of their being in the cellular mode. Particularly, they could create and implement cells with genomic programs and procedures for their diversified reproduction.

1.2.3. Our solar system is extremely tuned to living there cellulars, and the anthropic principle [33] explains this fact by the gigantic diversity of the regions of the Universe, where a various conditions may happen, including those of our solar system favorable for living.

Nevertheless, a scenario is not excluded, where the high tuning of the solar system and appearance of celluars there were attained constructively.

1.3. *Feasibility of constructive models of cells and models of their constructors.*

1.3.1. If cells cannot be originated in Nature, but they only be constructed, is it possible that Creators of cells themselves can be originated?

Indeed, acknowledging that the highest human Cogs approach to the constructive cells and, simultaneously, approach to cogsAI functionally equal to Cogs, it would be reasonable to look for premises that within the framework of physics rC*could be originated, then developing would attain to the cognizers cogsNtr functionally equal to Cogs that, in turn, for certain reasons would create cellulars with implemented procedures of biological evolution, Fig.9.

The reasons for creating cellulars could be similar and resemble, for example, those motivating nowadays humans to intellectual robotics.

1.3.2. Questioning the feasibility of constructive Creators of cellulars undoubtedly questions the feasibility of modeling God. Following the argumentation of Thomas Sheridan [53] the question remains uncertain until the connotative classifiers of God are replaced by their denotative ones.

Based on our reasoning, we can assume that focusing on constructive cognizing and creativity allows us to simultaneously extract analogous denotative aspects in classifiers of God, followed by the prospects for its modeling.

1.3.3. At the end of the above reasoning, let us remind that we are restraining from an unprofessional interpretation of models of physicists on energy and entropy, their connection with information and its possible origination but we very much hope that collaboration of researches to establish connection between the roots of cognizing and those on origination of primary classifiers by physicists can be very fruitful.

Conclusions Part 1

1. Let us now bring together some basic statements we argued in Part 1.

St.1. *Inductive algorithms, inductors, can form identified basic classifiers of the types of relationships over the matrices of imprints, primarily, of 1-, 2- place relationships, named later on* **do** *classifiers for n=1 and* **rels** *for n=2, correspondingly.*

St.2. *1- / 2-place rels along with covering the given matrices of classified imprints, extend them into new classifiers with certain IDs followed by uniting these classifiers with the ad hoc ones to use all of them, particularly, for the formation of further matrices of imprints.*

For example, if the representation of original matrices was based on n attributes/classifiers, then inductors can create new classifiers by proper reduction of those attributes, say by transition to min tests used in the reduction of Boolean matrices, then unite the reduced attributes by AND rels into new classifiers with certain IDs and later on form matrices of imprints using n+1 attributes/classifiers.

St.3. *Learning of mss, as a rule, is reduced to assembling the already learned rels into mss equal to those of communities, and this assembling is supervised, explicitly or not, by instructions provided by cms of those mss.*

For example, Markov algorithms are systems composed of 2-place rels of the types if/then, particularly, with disposition and ordering ones.

St.4. *Learning of rels or mss, explicitly or not, is supervised by communities and accompanied by inductive learning of classifiers of communicatives of the units of languages of communities assigned to IDs of these rels and mss in these communities.*

St.5. *Inductors enrich thesauri of mss with new classifiers corresponded to the matrices of imprints , universally wrt available at the time attributes/classifiers and thesauri.*

St.6. *Thesauri of mss can be communicated at any stage of their development.*

Indeed, colored graphs representing thesauri can be represented by matrices of incidents of their cms and vice versa.

St.6 is consistent with the view that the basics of the ability of cellulars to communicate their thesauri is innate.

St. 7. *Any mss induce systemic classifiers that are not just only symbolic, declarative but are also algorithmic, procedural allowing to effectively classify realities of their input domains.*

In other words, examine whether the realities match to the target mss or not.

gSt.8. *Classifiers attain the highest effectiveness if they are constructively regularized or can be adequately modeled by regularized classifiers.*

This statement, in fact, interprets the basic criteria of science to acknowledge classifications as the first priority step for the scientific representation of Nature.

2. We assume that rC* should access the matrices of imprint, then rC* should include inductors that can form 1- / 2- place rel at any level of expansion of thesauri, as well as assemblers of 1- / 2- place rels into mss representing, particularly, the algorithms themselves and the communicators of mss.

3.We also believe that classification and information are inseparable from each other and acknowledge that physicist approach to the grounding the origination of information, and, therefore, to the grounding the origination of non-entropic, negentropic, or living by Shrodinger realities.

Thus, we can question the origination of elementary classifiers in Nature, then their development to the highest cognizers.

4. Finally, let us remind some of the still open questions with their locations in Part 1 that , we believe, can develop our research.

Chapter 2/1

1. Can we provide relationships universal for certain classes of languages or prove that is impossible?
2. Can relationships be represented not numerically, particularly, by neuron nets?
3. What patterns of mental personality of humans including their emotions, motivations and social relationships can, in principle, be represented by adequate algorithmic models?
4. Are the patterns of personality of humans, including the entire ranges of human emotions, motivations and social relationships inevitable in cognitive modeling and if, so, can they be equally represented by mentals?
5. Can we advance in Human-Computer Communications to the extent acceptable, at least, for chess players recalling that chess languages represent the natural ones like drops of water in oceans?
6. Can we organize neurons in nets having mental doings equal to those of mentals?

Chapter 3/1

1. Given classifiers f presented by Boolean tables, we can transit both to their DNF and ANN representations.

DNF line is transparent, while the NN one remains vague.

To prove the idea that deep learning of f by NN can be interpreted as the construction of reduced DNF.

2. Construct RGT Solvers based on the processing of mentals

3. Extend chess scales of effectiveness of strategies to the entire RGT problems

4. Consist the scales of effectiveness of mental doings of humans with ones of doings of Solvers

5. Compare the adequacy of OJP and attributive estimates of the effectiveness of attaining certain solutions.

Chapter 4/1

1. How to organize ANN capable of performing all mental doings available to the mentals and equal to them?

2. Can thesauri Th be adequately modeled by computers, particularly, represented by the memory of computers in the way that any mental doings processed in Th could be equally reproduced in computers?

3. How to minimize the sets of attributes for the max compression of mpgi matrices without destroying the original classifications?

4. Grounding the fundamental hypotheses of Piaget that cognitive doings are learned stage by stage from certain root doings to the highest ones by means of only a few rules, we had argued that mentals, including the rules of their cognitive development, are reducible to certain roots, including 1- and 2- place classifiers, as well as - a chain of development of classifiers to various cognitive mentals was tracked to some extent.

4.1. Given octaves of mss and certain basic classifiers can we construct models of stage by stage development of human cognizing based on the inductors of revelation of 1/2 place classifiers

of increasing abstractness and on the procedures of acquisition of mss from communities and their processing for several cognitive doings?
4.2. Can we construct models of the origin of octaves of mss and basic classifiers , then unite them with the aforementioned models of development of cognizers, starting from the octaves of mss, to construct models of the origin the highest cognizers in Nature.

Chapter 5/1

1. Answering the question of how stored classifiers can be self-organized into algorithms it is worth addressing the ideas of self-organization of units of other types [63}.
Actually, let' ask the following question:
whether following the laws of physics in the spaces of originated 1-/2-place rels, exhaustive search procedures can be originated that seeking chains from the ongoing root-situations to the acknowledged utilities could regularly reveal algorithms, represented for example, as conjunctions of rels.
2. Note, that neurons of NN or of the models ANN are capable to unite the steps of transition from matrices of imprints to rule-based classifiers.
While it was found reasonable to have, at first, durable infers as attributes to form then case-based matrices followed by transition to rule-based classifiers, the neurons, in fact, make them in parallel.

Part 2. INTERPRETATIONS

Preface

1. Mental systems represent realities, have varying effectiveness with respect to our goals and are processed to support utilization and gain benefits from utilities.

Classifiers induced by mental systems are effective with respect to (wrt) the goals insofar as regularly provide utilities and enhance the effectiveness of modeling of these utilities constructively and adequately.

2. A mighty way to enhance the effectiveness and efficience of classifiers is to *regularize* them. Namely, classifiers Cl of members x of communities C are regularized in C if accompanied by ontological in C methods, instructions allowing x regularly provide positive samples of inputs of Cl as well as let the members of C do the same by communicating with x.

3. In this work we discuss constructive models of mental systems, *mentals*, covering by expressiveness algorithms, OO languages and approaching to natural languages, provide arguments of the adequacy of mentals for explaining, understanding and human-computer interactions as well as convince to follow the ideas of inventors of algorithms in adequate modeling of mental behavior.

4.To consist functional and connectivity mental models and recalling that artificial neuron nets (ANN) are systems of classifiers, we argue that mentals can be reduced to systems of basic classifiers, which can also be ANN.

Then, modeling classifying by mentals we argue that mentals, in principle, can be adequately modeled by ANN, which can consist of functional and connectivity mental models.

5. Humans are becoming powered enough to question further types of their being in the Universe, but so far they have no answers whether the solutions are in the corrections of their genomes , discovering new types of organizations of humans, or in the transition to a new type of descends, humanoid machines, etc.

Assuming that scenarios with dominating cognition are the most promising to resolve the challenges of evolvement of humans and the most expected for the next stage of their being, while the kernel of effective cognition is universal for all types of further being of humans, we present constructive models of the kernel, question the completeness and adequacy of its modeling and the ways of development of the models.

6. Progressing outcomes of AI , in fact, proves that realities not only of a cellular nature, but also those constructed, assembled from ordinary available to humans units of matter, can attain the highest cognizing of the Universe , at least, comparable with to one of the humans.

And since these constructed highest cognizers are only the assemblies of units of matter it is expected to question the necessity of humans constructing the highest cognizers.

We question the feasibility of the origin of the highest cognizers in Nature and enlighten some ways to answering.

Keywords: Negentropics, cellular, classifiers, relationships, cognition, constructive mental models, Markov algorithms, regularization, adequacy, explaining, human-computer interactions, neuron nets, origin of cognizing.

Chapter 1/2. Interpretating Introduction to Part1

1. Being by Classifying

We, humans, are somehow able to represent the causers of imprints in us and the imprints themselves by classifiers of these imprints. And since imprints can follow only certain doings of their causers (external, internal, or both), the classifiers represent the causers representing, in fact, the doings of the causers.

1. We process classifiers to preserve or provide our utilities.

We enhance the effectiveness of this process, particularly, enhancing the quality of classifiersas, as well as uniting the efforts of community members for the utilities by communicatives (cms) of the classifiers, i.e., by IDs of classifiers or by classified samples , as, for example, in this work, we communicate now by English words corresponded to IDs of classifiers of the author.

2. Classifying the causers of imprints and imprints themselves as *realities* and their totality as our *Universe* , we assume that the effectiveness of preservation of our identity is to the extent, to which we, in particularl, adequately represent the Universe by constructive classifiers and effectively process them.

In other words, as comprehensively and adequately classifiers cover the diversity of realities as powerful we become.

For example, each of us enhances his/her effectiveness in the Universe by developing his/her language skills. Then, communities, for example, speaking in English and using more than

300 thousand highly constructive classifiers are incomparably more powerful than tribes with languages of a few thousand words representing mainly non constructive classifiers.

And whether the first assertion of the Bible "...In the Beginning was the Word, and the Word was God" doesn't remind us that the classifiers and their cms in languages, acquired either from genomes or from communities are one of the fundamentals of our power…

2. Classifying Being

Identifying being, humans were for long time perceiving themselves as an inseparable part of physical nature , and then were concentrating on the mental patterns of their being.

Buddhists and many of their followers represent these patterns as those of all over and ever- existing conscious Cosmic Mind while Christians represent them as immortal souls created by God similarly to and governed by God.

Both Buddhists and Christians interpret being as cognition of humans of themselves, thus, cognition of and approaching to the God.

1. These days, in a diversity of classifiers we identify ourselves as originated and formed by communities, then as a type of cellular realities, *cellulars*, which, in turn, identified as an intersection of a type of non entropic, negentropic realities [6,55], *negs,* with a type of durables, *refers*, i.e., realities capable of preserving the identity of imprints caused in them. Consequently, humans and representations of humans, say by their doings, are, at least, dependent, although mainly predetermined by fundamentals of communities, cellulars and negs.

1.1. Particularly, mental doings of humans are essentially predetermined by genomes and cultures of their communities, implying the communality of members of communities not only in innate means of representing realities by sensors and classifiers, in types of processing of these representations for various utilities, including thr enhancement of effectiveness of themselves, but also, in general, in the commonality of particular lines of reasoning, counting, expressing them in languages, etc.

That is why the novelty of mental doings of humans, usually, is in enlightening lines of cause-effect reasoning between the already known communal utilities and representations of realities. And only occasionally the novelty is in discovering new utilities or case-effect reasoning why these novelties become so sound in communities up to becoming awarded by Nobel Prizes.

1.1.1. Thus, reflecting the above assumptions to this research we classify it as an attempt to enlighten and then generalize already known and successfully applied expertise of mental modeling by the founders of algorithms, namely, the expertise of transition from classifiers of computability to their models, algorithms.

And, apparently, in this modeling we cannot but have to heavily rely on the fundamentals of communities, cellulars, negs and durables stated, particularly, by the following assumptions.

3. Fundamentals of Durables, Negentropics and Cellulars

Ad1. Let us define *durables* as realities that, in contrast with others, *temporals*, to some extent can be properly identified in the meantime.

For example, *refers* are durables causing certain outputs from certain types of realities. A type of refers, sensors, can cause imprints from warmth, light, sound, chemicals, etc.

Other durables, *absorbers,* are durables in absorbing liquids or some types of chemicals.

Realities synthesizing sugars from carbon dioxide and water are types of *energy saver* durables, while sugars are, in fact, types of storages of the light, warmth energy.

Ad2. Durables can be *static*, like stones, rocks or solar systems, and *dynamic*, like rivers, oceans or star systems.

Active durables do by themselves for durability, in contrast with *passive* ones only objects by doings of others.

Note, that durablesare tightly related to the concept of *strings* in cosmology [33] and , seemingly, can be reduced to them.

1. An1.*Root negentropics*, rNg, are active durables consisting of refers, storages and energy savers along with ability to process stored energy for preserving doings of rNg, while *negentropics, negs,* are realities containing the root ones, i.e., rNg.

An2. The existence of types of durables seems tractable, but the origin of negs and cellular (even though as uncials) still remainsa a mystery.

Nevertheless, a positive answer to the question is expected in [55], some artificial negs, robots, humans can already construct, while studying in AI induce tracks to the constructive cellulars [56,57].

2. Ac1. Cellulars *do* to reproduce themselves, to benefit from *utilities,* i.e., realities favorable for the *roots*, to avoid their *damagers*, to *utilize* realities still uncertain wrt the roots, as well as to challenge the already gained utilities and, possibly, roots themselves.

Ac2.Cellulars represent realities causing doings.

Ac3. The vast majority of doings of cellulars are predetermined by genomes and cultures of their communities.

Ac4. Cellulars gain effective doings, at least, by a chance search in the space of diversely replicated doings of their cells, and, possibly, by regular cognition of regularities of the Universe.

Ac5. Cellulars have *sensors*, or classifiers outputting identified imprints, *percepts,* for their certain types of inputs.

Ac6. *Mental doings* of cellulars are doings over percepts, the outputs of genomic classifiers and those gained in the lifetime. They are aimed at supporting all doings of cellulars.

Mental doings represent processing of a type of realities, *mental doers (mdoers*) and *systems of mdoers (mss)* while it will be argued that mdoers are reducible to certain types of classifiers.

Ac7. Mss of members x of communities C of cellulars comprise certain nets xN, in fact, nets of classifiers.

The nodes of xN are IDs of mss, while the mss, as well as *meanings of IDs* are represented by connectivity subnets of nets xN rooted in these IDs.

Mss m can be activated internally or externally by their IDs or by samples r of already classified by m realities.

Sets of IDs of mss m or classified by m and communalized in C realities r comprise *communicatives* (cms) of C.

3. Commenting on the assumptions and, first of all, on the negentropicity of humans let's recall that entropics, following Schrödinger [6], comprise the vast majority of realities. They inevitably lose their energy, and therefore, any sign of durability.

In opposite to entropics, negentropics, negs, comprise only a small island of realities and are able to preserve certain durables, *root realities*, or *roots* in space and time, and the premisenecessary to preserve the roots is their ability to regularly gain energy from others.

We assume that r*oots* for realities r, are any given, usually not yet explained realities, possibly constituents or doings of r, that are preserved for r regularly and with first priority wrt others.

Apparently, roots of negs necessarily include doings to gain energy and those to preserve this ability.

4. Roots of cellulars include, at least, their genomes, doings for periodic diversified reproduction of genomes and doings for preserving realities induced as auxiliary to the roots.

4.1. Mental doings baking other doings, including themselves, are either genomic or gained in the lifetime , while gained mainly by acquisition from the cultures of communities.

5. ***Do and Systemic Classifiers***. *Mdoers* do over outputs of sensors and their chains elaborate instructions for the effectors.

Do classifiers Cl are defined as a type of doers that either elaborate certain outputs for realities at the input parts or remain passive, thus, splitting input domains into two classes +Cl and ?Cl.

We state that mdoers can be represented as classifiers of various types, particularly, *relationships (rels)*, *rules, regularities* or their compositions, *algorithms*, and we argue that all these classifiers are reducible either to classifiers of n-tuples of identified, nominated realities or to the systems of those classifiers.

5.1. Mss compose mdoers to represent, particularly, *systemic classifiers* , for example, Factories, Computers, Chess Positions, etc.

We assume that any systems h, including mdoers and mss, induce certain systemic classifiers hsCl with positives +hsCl s determined at a glance as follows.

Realities r,r' are equal with respect to doers d if outputs of d applied to r,r' are the same .

In other words, the analysis of r, r' by their embedded rules doesn't find any distinction between r and r'.

Doers are equal if their performances, i.e., the input/output pairs, for inputs of their indoms coincide.

Systems G, G' of doers are equal if decompositions of G,G' until the terminal ones are isomorphic wrt equality of corresponded to each other doers, while these doers are linked by equal rels.

Realities r *match* the systems h if certain doers d can reveal in r equal to h systems h'.

Finally, these pairs (h, d) determine *systemic classifiers hsCl* with positives +hsCl.

6. ***Goal Orientation***. Classifiers of roots and utilities are identified as *root* and *induced goals, so* negs, including cellulars can be classified as *goal oriented* realities.

Attributes, we assume, include classifiers of utilities and their constituents if the utilities are compound as well as, possibly, classifiers of realities with uncertain yet utility.

Apparently, realities can be partially ordered by degrees of their utilities wrt the roots, thus, inducing corresponding ordering for goals and attributes.

In that ordering, classifiers of the first layers identify utilities for the roots, while the ones of the consequent layers i can classify utilities not only wrt the roots but also wrt any goals k of the layers k before i, i.e., k<i.

6.1. Mss, as well as their constituent mdoers, can be processed for a variety of goals, such as learning new utilities and enhancing the effectiveness of mss.

7. ***Fundamentals of Humans.*** While the fundamentals of mss and their processing can be found for all cellulars the highest ones of them are unique only for humans that can be stated, particularly, by the following assumptions.

Ah1. Doings of members of human communities are mainly equal implied by thr equality of 99% of genomes of all humans and the commonality of cultures of communities of their being.

Ah2. Humans adapt to the Universe mainly by cognition and development of mental doings.

Ah3. Humans accumulate, reproduce effective doings, and then transfer them in space and time not only by genomes but also by the records of the patters of these doings that are essentially depersonalized and estranged from particular members of their communities.

Ah4. Meanings m of mIDs of x@C are connectivity subnets of nets xN rooted in mIDs that can be scaled by their effectiveness wrt utilities of x, say, constructiveness of adequate modeling of m, and wrt explanations of m in C, say, completeness of m wrt intensions of x and expectations in C.

Ah5. Mental doings classified by psychologists and psychiatrists including classifying, learning, prognosticating, communicating can be equally represented by adequate constructive models of mss and their nets.

8. (The refined definition see in the Intro of Part 1) ***Cognizers*** are, we assume, a type of mss, while *cognition* includes doings in acquisition, accumulation, as well as revelation, discovery of mss, particularly, by learning new utilities or enhancing the effectiveness of the existing ones.

7.2. (The refined assumption see in the Intro of Part 1)) All over governing of mss, including cognitizing , activation and processing, is realized, we assume, by *controllers* that, it is not excluded, can be the causers of our awareness or consciousness and also be mss.

Particularly, controllers govern the communication between the members of communities, explaining and understanding the mss of each other.

7.3. Being realties we can also classify and explain ourselves.

For example, controllers explain the mss “Humans” of the author by resolving it into this ongoing text, namely, corresponding English words to the IDs of constituents of mss of the author.

In general, this resolution can start with any constituents of the target mss, while their IDs can be chained causally, logically or in a variety of other modes and detailed depending, particularly, on the goals of the author.

7.4. Mainly equal doings of members of communities C mean, particularly, that

- things ***r*** causing equal imprints for arbitrary x,y@C , imply corresponded to ***r*** and mainly equal mss of x,y, what implies certain Universe UC equal for all members of C,
- mental doings including classifying, learning, teaching, inference, prognostication are equal in C, which allows the members of C to communicate doings of each other aimed at enhancing the effectiveness of their doings by collaborations.

4. Effectiveness of Classifiers

A mighty way to enhance the effectiveness of mss, and thus, cognizers, is the *regularization of classifiers* induced by mdoers and mss [47].

Namely, classifiers Cl of members x of communities C are regularized in C if are accompanied by ontological in C methods, instructions allowing x to regularly provide positive samples of inputs of Cl as well as let the members of C do the same by communicating with x. In constructive regularization, these samples can be provided deterministically and without any involvement of cellular while, otherwise, can be grown from a priory given prototypes like cells or crystals, be the products of services to humans or machines.

1. Regularly provided positives r of classifiers Cl and Cl themselves are interpreted as *models* of classifiers Cl' if r are classified as positives of Cl` , while Cl are interpreted as *adequate models* of Cl' if positives r meet certain additional requirements focused on positives of Cl.

For example, algorithms are adequate models of deterministic methods if, following Church, equal algorithms can be corresponded to any method by certain instructions [30].

5. Extending Constructive Mental Models

Interpreting algorithms as constructive models of a type of mental doings, deterministic methods and OO Languages as a constructiveextension of algorithms for theirhigher expressiveness, we constructively extend the expressiveness of OOL to approach one of the natural languages and state the following:

Stt1. *Algorithms constructively regularize and model deterministic methods, OOL constructively regularize and strongly extend the expressiveness of algorithms, while mentals constructively regularized and strongly extend the expressiveness of OOL.*

Stt 2. *For languages L of communities C allowing the members x of C to communicate,* i.e., to explain and understand mss of each other expressed in L, *communication algorithms LC can be constructed letting computers communicate mental models Mns of mss Ms of C equally wrt the members of C if Mns and Ms are equal to each other.*

6. Questioning the Adequacy of Mentals

1. Justification of mentals as adequate models of mss can be performed by analogy with justification of algorithms as adequate models of computability by Church.

This justification, ideally, means that for the original problem Being of humans in the Universe (BU) for systemic classifier sClm of any mss m of any x@C solving BU, it is possible to provide mentals m' with classifier sClm' equal to sClm.

Realistically, since the adequacy of mentals can be examined only for a finite number of mss, it is worth examining the adequacy, first of all, for certain key mss. As such key mss we select meta mss, i.e., ones doing over mss, then, ones acknowledged by psychologists, psychotherapists as nucleus in identifying cognizing norms for humans.

2. Acknowledging, that OO Languages extend algorithms and are types of regularized mss that are adequately modeling computation, as well as doings of systems that humans can code by numbers in the ways letting equally represent relationships between the originals, and that it is highly questionable whether OOP can provide regularized models of natural languages , we assume, that mentals enhance the, adequacy of mental modeling since

-first, they are exempted from the requirement to have only numeric input IDs provided by human experts/programmers, i.e., they are extended to systems capable of operating along with numeric input IDs with ones provided by the given sensors similar to those of neuron nets, and

-second, mentals let represent cognizing relationships we identify in natural languages.

3. Adequacy of mss, at present, is being questioned both, functionally and connectively.

Functional questioning examines the equality of performances of mss and models of mss of any origin.

In contrast, connectivity modeling requires that the units of models of mss were adequate models of the units of nerve systems, neurons, and is questioning the adequacy of mss with artificial nets of neurons, *ANN*.

Examining primarily the functional adequacy of mentals and recalling that ANN are systems of do classifiers, first, we provide evidence that mentals can be reduced to systems of do classifiers, and also state, in fact, that

Stt3.*Mentals can be reduced to systems of do classifiers*

Stt4. *Mentals can consist of functional and connectivity mental models.*

4. Proving of the ability of transition from models of mental doings within the framework of functional paradigm to equal models of the same doings within the framework of the connectivity one and vice versa will provide a fundamental argument that these models, in fact, catch the essence of mental doings.

ANN models of classifying mentals allow us to argue the following

Stt5:*Mentals, in principle, can be adequately modeled by ANN, which, at first, represent 1-/2-place resls of mentals by equal to them unit ANN, then, compose these units into ANN equal to the basic constituents of mentals and, finally, unite these constituents into ANN equal to the target mentals.*

The arguments include the methods of inductive formation of one/two place relationships that being the basic units of mentals , by analogy with those in clauses of languages, induce the guides of assembling mentals from these relationships, particularly, in the ANN modes.

4.1. To carry out the above reduction not only excitedly, in principle, but also constructively, the storages and linkages of the unit ANN into the target compound ANN should to be refined.

Recalling NN, where the rels between neurons of the same or different layers, are, assumingly, represented by synapses, while the number of those synaptic links of NN of any human concurs with the number of particles in the Universe let us assume the statement

Stt6: *Given A and B NN classifiers, the rels1 between A and B can be formed as a new NN classifier Arels1B linked by synapses necessarily to NN classifiers of A, B and expectably, to other ones,*

that can provide a hint to the modelingof mental classifying of the types of mentals or ANN, where similar to children, the complexity of the learned relationships can be incrementally enhanced, starting with the units of the types of conditional reflexes , then, step by step uniting them into more and more compound classifiers.

5. The next barrier in justification of the adequacy of mentals is the incredibility of problem of Being of humans in the Universe (BU) for examining adequacy of mentals to target mss. Ideally, to prove the adequacy of mentals m' for target mss m, we had to confirm equality of m and m' for any type of their relevant processing for any tasks of BU problem, which is, apparently, unrealistic.

To overcome this barrier, we follow the views that BU problem can be approximated by game models [52]. Then, similar to [27] ,we assume, that combinatorial RGT games with known hierarchies of utilities and solutions in spaces of possible strategies in game trees can with proper adequacy represent the BU problem.

Examining the adequacy by RGT models of BU includes examining the equality of

- models of mental doings adequate for RGT kernel problems wrt comprehensive mental doings in Universe

-scales of effectiveness of RGT Solvers wrt to scales of effectiveness of mental doings of BU.

7. Advancing in Adequate Mental Modeling

1. At the present time a variety of mental doings have proper algorithmic or OO models including the following ones.

The branch of theory theory of algorithms, synthesis of algorithms, where assuming a priory certain classifiers of mdoers are already given, algorithms of synthesis of equal constructive versions of these mdoers are developed.

In deductive modes of synthesis, these mdoers can include certain axioms and logical statements or can be determined recursively [31]. In inductive modes, including machine learning, these mdoers can be represented by samples of their domains or their representations, performances of mdoers, etc. [39].

Certain mss provide methods for transferring, teaching of mss inside the communities C as well as methods of acquiring of these mss.

While commonality of thesauri of members C let them avoid specification of these methods, it becomes unenviable in transmitting and acquiring human mss by computers.

In contrast with machine learning, where teachers are forced to provide computers with representations of mss step by step, by portions, teachers can do that holistically and completely when they teach them. The questions arise to the abilities of computers in accepting, disposing and properly processing these mss.

Prospective answers to the above questions for RGT problems are presented in [33]-[36], [39]-[43].

1.1. In Part 1, we were analyzing how a key mental ability to communications can be regularized as the first step to its constructive adequate modeling.
Then we argued that both functional mental models, mentals, and connectivity ones, ANN, are composed of the classifiers of realities, i.e., they have the same roots, why the question of consistency of functional and connectivity models, in principle, is reduced to proving that equal mentals and ANN can be provided for any mental doings.

1.2. We have interpreted the constructs of mentals by the basics of attributive classifying of theoretical computer science, then, referring to some relevant models of classifying**, we** provide a method for inductive formation of two place relationships that being the basic units of mentals, by analogy with those in clauses of languages, induce the guides of formation of mentals, particularly, in the ANN modes.

We also discussed the formation and acquisition of systemic classifiers and ways of coping with imperfect classifying.

1.3. We analyze in details how the On the Job Performances (OJP) adequacy of mental**s** wrt to mss can be acceptably estimated by performances of RGT Solvers stating , particularly, that the scales of effectiveness of Solvers and humans have to be consistent and doings of Solvers should be completely governed by mentals.

8. Looking Ahead for Being.

Humans are becoming powered enough to question the further types of their being in the Universe, but so far they have no answers whether the solutions are in the corrections of their genomes , discovering new types of organizations of humans or in the transition to a new type of descents, humanoid machines or others.

1. In [42], it was proved that in classes of algorithms A, allowing to enumerate all possible appearances of algorithms by enumeration of the states of regulators that determine any of their appearances, it is possible to arrange the sequences of tournaments between these algorithms similar to those by Elo in chess [19] allowing to converge the sequences to the best in A algorithms.

In general, the theorem provides a frameworkof models of adaptation for Being in the Universe for a various regulators of adapting algorithms.

Regulators of adaptation based on acquisition of thesaurui of human communities and available by their recordings, for example, provided by Wikipedia, have a special impotency.
On the way to this dream we outline research steps that we believe may approach to the modeling of acquisition of human thesauri by nets of mentals.

2. Progressing outcomes of AI in the constructing **of** basic, root cognizers capable to developing tothe artificial ones, functionally equal to the highest human cognizers, lead to significant consequences. In fact, AI proves that realities not only of cellular nature, but also those constructed, assembled from ordinary, available to humans units of matter can attain the highest cognizingof the Universe, at least, comparable to one of the humans.

And since these artificial, constructed highest cognizers are only the assemblies of units of matter unconstrained from the mysteries of the origin of cells, it is expected to question the necessity of humans or some ones else in assembling, constructing the highest cognizers.

In other words, it is expected to question the feasibility of the origin of the highest cognizers within the framework of acknowledged physics.

We do enlighten the base of these consequences and track some steps of their grounding.

9. Priority of Cognizing in Attaining Negentropicity

Accepting that sensors, effectors and mental controllers are inevitable for any negs, we assume that

Stt7. *Mentals introduced to regularize constructively mental doings of humans can be considered as constructively regularized mental models for all negs including those of the class of energized negs*

Assuming that scenarios with dominating cognition are the most promising to resolve the challenges of evolvement of humans and are the most expected for the next stage of their being, while the kernel of effective cognition is universal for all types of further being of humans, n Parts 1-2 we present constructive models of the kernel, question the completeness of representations and ways of its further study.

1. The Content of the Work and a view on our references were aleady presented in the Introduction of Pat 1.

Chapter 2/2. Interpreting Cellulars

1. The Mystery of Cellulars

1. The mystery of negentropic islands of cellular realities (cellulars) in the Universe, embraced by entropy, lies in the fact that root doers and systems of doers of cellulars, including backing roots mental and other evolutionary justified systems, are determined by a type of programs, genomes.

1.1. Roots include cells themselves, doers of replication of cellulars, classifiers of beneficial or damaging roots of realities and other constituents that, seemingly, were comprising proto-cellulars themselves.

Genomic realities that are backing roots include, for example, doers for assembling the nests of birds or sounds of infants indicating their hunger or joy.

Any cell of either unicellulars or organisms contains genomes of the corresponding types.

Genomes provide instructions for assembling roots from surrounding elements, including assemblers and genomes themselves.

Codes of any genomes of any type are chains of the same 4 types of -nits that are passed only first hand from parents to their descendants, while the origin of the proto genomes remains uncertain.

1.2. Genomes provide instructions for the formation of cellulars, ensuring that the doers of descendants are comparable to those of their parents

The process of formation of descendants includes the embodiment of genomes of their parents, as well as the acquisition of ad hoc systems of doers of communities, if any, and doers discovered and revealed personally through the lifetime (by this we mean life as an organism).

Milestones of a lifetime are conception, birth, gaining memberships in communities or families, decline and death.

Genomic formation lasts from the conception until death, while the personalized and community formation lasts throughout the lifetime, at least since the birth, but possibly earlier.

Some cellulars, such as unicellulars or insects, are completely formed at birth by genomes, right after the embodiment of their genomic programs.

Descendants of humans, and some other cellulars are formed continuously since the conception, first by genomes, and then throughout the lifetime, both by genomes and by cmmunities via interactions with their cognizers.

1.3. Ultimately, the being of descendants of cellulars throughout their lifetime is predetermined either completely by genomes or both by genomes and cultures of communities of cellulars, if any.

The bulk of the essence of cellulars is genomic, a tangible part of it comes from cultures, and only a small fraction is revealed personally throughout the lifetime.

For example, the being of ants or bees is almost completely genomic, while the list of professions of humans and what they chose personally throughout lifetime essentially succeed from genomes they inherited, cultures of their communities and scopes of cultures they have acquired. Particularly, humans of creative professions are specialized in enriching the cultures of communities, for example, by enhancing the effectiveness of their mental systems on appropriate scales.

Thus, the role of "free will" in choosing personal types of being appears to be very illusive…

And since cells evolved into organisms and then to communities of organisms, one may wonder if the organizations of types of anthills or swarms of bees, enriched with the essentials of the power of humans, could not become a new idea for human progress. Particularly, the essentials to adapt to the Universe not only cognizing it by chance via blind diversified replication, as many cellulars do, but, in addition, by conscious cognition of Universe during their lifetime, recording, accumulating and passing mental systems to descendants free of bounds of time and space.

Recall, that anthills exist for more than 40 million years vs a couple of million years for humans…

2. Following the cosmological views [33], we classify the vast majority of realities as *entropics,* while interpreting Schrodinger [6], we classify as *negentropics or negs* an island of non-entropic realities r that can gain energy from others to preserve certain *roots* in space and time, i.e., realities, say, constituents or doings of r, determining the identity of r.

Roots of negs, as it follows from definitions, should include, at least, means to gain energy and ones to preserve these means, which, in turn, require means to represent and affect realities*,* i.e., *sensors* and *effectors,* then means to *classify utilities*, *damagers and uncertainties*, i.e., classifiers of still favorable, damaging or uncertain with respect to *(wrt)* the roots realities, as well as *controllers* of the doings.

2.1. We classify cellular realities, cellulars, as a type of negentropics, which include alive cells in the roots capable of reproducing and composing themselves into the organism, while classifying humans as a type of cellulars.

3. Accelerated mental power approaches humans to "Homo Deus" [34] with a chance of reliable preservation of cellular roots while, in parallel, to the entire crashing of cellulars on Earth.

The kernel of this strength lies in the effectiveness of organizations of humans, while the weakness is caused by the imperfection of these organizations and their genomic heritage.

Humans are becoming powerful enough to question the further types of their negentropic being in the Universe, but they do not yet have answers whether the solutions are in the corrections of their genomes, discovering new types of organizations of humans or in transition to a new type of descends, humanoid machines or others.

It is worth reminding that nowadays autonomous agents, a type of constructed negentropics, are challenging the cellular being of humans at all, as well as recall that following Buddhist and Indian sources, humans being lemuroids or Atlantis in the far past, could have not only cellular types of negentropicity [35].

Assuming that the kernel of effective cognition is universal for all types of negs, the scenarios with dominating cognition are the most promising to resolve the challenges of the evolvement of humans and the most expected for the next stage of their being, finally, we question the completeness of representation of the kernel and the ways of its further study.

2. Cellulars: Being

1. We, humans, classify everything causing imprints in us as realities, and we classify the totality of realities as the Universe.

And we extrapolate the Universe to the *Universe, i.e. to anywhere and anytime that we suppose can be existing.

Humans classify themselves as a type of cellulars, a subclass of livings, which, in turn, are considered as a subclass of negentropics, i.e. realities capable of preserving certain properties, roots, consuming the energy and matter of others [6].

2. Any cellulars of any type, descendants, originate from some already existing cellulars, parents, of the same types at birth from the first hand and through means provided by the parents, say, egg cells, seeds, spores or tissues.

While the chain of transmissions of genomes from parents to their descendants is lasting about 4 billion years after half a billion years of Earth's origin, the origin of proto-cells remains uncertain.

3. Any descendant cell contains specifications, programs – genomes, replication of themselves, as well as means realizing replication, including processors of the genomes and constructors of new cells from nutrients.

These constructors include primary classifiers of energy and matter, sensors; mental doers and systems over sensor outputs, analyzing and processing these outputs to elaborate instructions for managing realities; effectors realizing these instructors and controllers coordinating the doings.

4. Cellulars can be classified into the categories of unicellulars or protists, multicellulars or organisms, and communities.

Focusing on the commonality of protists and organisms, we will refer to them as cellulars, while referring to "communities" - only for communities of organisms.

5. The formation of organisms can be completed at birth, as it happened with, say, insects, upon reaching maturity or with memberships acknowledged by communities, particularly, parents, say, for packs of wolves or tribe communities, and can last for a long time for advanced human communities.

Correspondingly, cellulars of some types, insects, in lifetime preserve their software and hardware, determined genomically at conception.

Other cellulars, especially in communities, develop, at least, their software in lifetime and may differ from each other in cultures and degrees of acquisition of the cultures of communities. Humans, in addition to first-hand acquisition of cultures, can acquire cultures aliened/estranged into books, arts, etc.

5.1. Cellulars preserve their roots in changing environments by changing themselves, i.e., adapting or changing environments.

Indeed, all cellulars adapt genomically by replicating into certain extented versions of themselves and continue being those as their versions that survive.

For example, protists or insects adapt by replication of a huge number of versions of themselves, thus enhancing the chances that some of the versions will survive with this huge number of versions.

Community members, in addition to diversified replication, adapt cognitively by gaining mental systems accumulated in cultures or revealed personally in lifetime.

In turn, communities adapt by integrating adaptation variations of their members.

5.2. The types of cellulars ever being on Earth are about 10 billion.

The existing types by Darwin are those that have adapted to the changed environments. Simultaneously, let's recall that there is no strong evidence on a smooth transition, evolution from one type of cellulars to another, since the intermediates between the existing types of cellulars did not yet find, why the hypothesis on parallel being of these types has not yet been refuted.

5.3. Historically, the adaptation of cellulars is biological, and its engine is the diversified replication.

However, nowadays, humans are adapting, primarily, cognitively in lifetime and progressing, particularly, to the construction of new types of cellulars not excluding the essential transformation of humans themselves.

3. Cellulars: Doings and Measurement of Doings

1. Cellulars do since conception, first, should be formed equal to their parents either by embodying only their genomic programs, like insects, or, in parallel, gaining memberships in communities, like humans.

Then, at least, since birth cellulars do benefit from utilities, i.e., from realities directly or not favoring their roots, as well as avoid from damagers, utilize realities either already classified as reducible to utilities or classify still uncertain ones.

The essence of gaining membership in communities C is to acquire the accumulated by C and common in C doers, certain meta-doers of controlling and developing them, as well as communicatives, cms (say, IDs) of these doers to communicate about realities for coordination of the efforts of members of C in solving common in C problems.

2. For humans, these doers are mainly mental ones, mdoers or their systems, while cms of mdoers can be IDs of mdoers or some representations of their nature, say, the samples they classify, or the models of these samples.

We assume that mdoers comprise systems having unique IDs and being united in nets with nodes and relationships between them assigned by these IDs.

Mental systems, mss, in these nets are interpreted as connectivity subnets rooted in certain nodes of the nets.

Mss determine classes of equality of realities that can be processed for several goals and decomposed or abstracted.

For example, mss of Factories include nodes of of their stuff, workshops, buildings, etc., and a variety of mss and realities can be equal or match this classifier, which, in turn, can be decomposed or abstracted.

Cms of C of certain types represent mss and can be organized into languages L, like English is organized from IDs of certain mss that have thesauri comprising mss of C, corpora of totalities of IDs, syntaxes and semantics of L.

3. Mental systems in a variety of their scopes are intrinsic for cellulars and at different periods of lifetime are formed genomically or cognitively.

They are backing doings for roots or goals induced by roots and varying in effectiveness with respect to them.

Effectiveness of mss rises by, particularly, grounding mss in commonly acknowledged ones like axioms or rules of logical inferences, eliminating malicious circles in their representations, and more, being consistent with, say, cause–effect or originations in time chains of realities the mss represent.

The most powerful mss are seemingly gained when we provide their constructive models and make them adequately.

4. Cellulars scale realities by degrees of their values with respect to (wrt) roots and goals induced by roots, as well as by degrees of specification and modeling of valuable realities allowing their estrangement from persons for sharing in human and other, say, autonomous agents, communities.

Scales can be autonomous for particular members of communities, i.e., be person sufficient, or can be acknowledged by all members, i.e., be community sufficient.

Let us provide some of these scales.

4.1. The degrees of a person-sufficient scale of realities can start with those indicating that realities cause certain prints, then indicate that realities have classifiers, and then refine whether realities have models either adequate or not, and whether the models are constructive or cellular dependent.

4.2. Realities r (or representatives ^r of r) of members x of communities C can be estranged from x to C or others if, autonomously from x, they can process certain realities r\`(representatives ^r\`of r\`) equal to r (^r).

Models r\` of classified realities r of x provide examples of such estranged from x realities. In turn, the models r\` of realities r can be classified by new classifiers and cause the corresponding models, etc., thereby inducing the scales of models of enhanced abstraction.

4.3. Community-sufficient scales of C of estranged from x realities r, along with the models of r of x or ordered by certain criteria classifiers of r of x, can be the IDs of r or their representations.

These scales can represent the degrees of proliferation of models of members C in the entire C and the degrees of equality of quality of these models to the quality of original models.

It can be assumed that the commonality in C of some models of members of C, community-sufficient scales, can represent the degrees of communicativeness of these models in C.

5. Other aspects in classifying mental doers and systems include, particularly, synchronous availability from their origin in time, say, in the past or present, being causes or their effects, and equidistant accessibility, like the nodes in "star" type networks.

In general, all ongoing mental doings classified in science, are relevant to be mentioned here.

But following the objectives of this research, we have recalled only some of the key aspects of mss to substantiate the consistency of constructive specifications with their scientific classifications.

4. Communities

1. Wiki classifies communities as "… a social unit (a group of living things) with a commonality such as norms, religion, values, customs or identity."

Accepting this classification allows us to focus on some peculiarities of human communities and try to interpret them within the framework of our models.

2. While many aspects of communality are well-known for various types of cellulars, their uniqueness for humans are, particularly, in

- possession of a huge number of lifetime mss
- generation after generation intensive growth of lifetime mss
- huge accumulators of the records of communicatives (cms) of mss
- diversity of types of cms of mss, say, IDs or representations of their samples
- diversity of types of records of cms, by which the descents can acquire mss of communities.

2.1. We believe that uniqueness of humans in the enlargement, actually unlimited, in the number of revealed mss and in the ability to accumulate the cms records of mss, letting humans use mss further than the limits of livetime of each of their generations, are, essentially, in their ability to manage the cms of communal mss by their estranged codes and in the highest diversity of these codes.

2.2. The effectiveness of managing the lifelong mss of humans tends to be more valuable, precise and compact than the evolutionary gained genomic ones.

Actually, humans are approaching being with lifetime gained doings overlapping the genomic ones.

2.3. The means of recalling mss of communities or transmitting them to each other either by direct communication or records of their cms, let them, particularly, be free in choosing ways of their further being in the Universe, say, continuing evolutionarily shaped ongoing ones or choosing more perspective ways in their renewed hierarchy of values.

3. What follows is a view on cellular communities, while it seems to be extended to one of the autonomous agents and, possibly, negentropics.

Assuming a set W of almost genomically equal cellulars, *the subsets vC of W are* ***vC communities*** *if the members of vC are, at least, equal wrt the attributes v of the sets V, in general, of fuzzy attributes of cultures of W.*

Apparently, cultural attributes of members of communities should address to the commonality of their thesauri, languages, moral and ethical principles in choosing the means to attain some goals, etc.

For example, scientists comprise a variety of communities identified by peculiarities of their professional specializations.

Attributes v for nations and states should estimate the commonality of languages, traditions, history of members, the unicity of laws they are obedient and live up to.

Wiki, for example, provides proper attributes for families, religions, congregations.

3.1. Nowadays the carriers of ideas and ideologies form communities of their followers explicitly, or tighten, particularly relying on regrettable impunity of muggings in virtual spaces, which is becoming disastrous for conventional and established communities.

5. Challenging Cellular Negentropicity

1. Evolutionarily, cellulars have always fought and competed for energy resources, as any environment in which they choose to reproduce themselves, is physically restricted by the energy that inevitably causes competition and struggle for survival.

2. Mental power of humans lets them become unique cellulars approaching all sources of energy of the Universe, thus, giving them an opportunity to exempt struggle for survival.

3. Picks of mental power of humans on the scales of its effectiveness and efficiency include, particularly, the following ones.

3.1. Awesome coverage of all types of realities of the Universe by classifiers and enormous expansion of the number of classifiers.

Classifications span from particles, black energy and matter to spaceships, from sequencing of genomes to genetic engineering, from computers to global networks and their controllers.

An estimate of the number of human classifiers can be the number of about 300, 000.00 units in English dictionaries.

3.2. Continuous rise of abstractness and universality of classifiers.

3.3. Approaching the regularization of classifiers of any external or internal realities.
Starting with the grown up regularization of, say, plants and domestic animals, humans are intensively progressing in constructive regularization of mental doings, especially cognition that exempting from cellular dependency enhances the dimensions of empowering.

3.4. Along with communal being of many cellulars in families, parks, swarms, etc., humans along with families and tribes, invented various types of higher organizations, effectively integrating them for common goals, particularly for cognizing.

4. The aforementioned and other dimensions of mental power allow us to question whether humans with unlimited access to energy sources can organize themselves into the type of negentropics that, instead of fighting for energy resources, will prioritize their mental doings, say, cognizing of Universe, to become so knowledgeable that could, at least, serve their successful ongoing access to energy and be ready to prevent possible troubles with it.

4.1. An unconquered obstacle to organizations of humans, particularly, those with domination of cognition, is in their evolutionary malicious genomic heritage to fight for superiority and subsequent possible aggressiveness, greed, evil, cruelty resulting in evasion of members of organizations from routine obligations or their erosion by corruption.

4.2. To avoid routine work and servicing for humans, we could assume that with the growth of intelligence of machines, they would be able to take this burden, while humans could concentrate on creative work and cognition.

Unfortunately, these options raise new serious problems.

With passing services to machines, humans should delegate them certain control that will inevitably result in malicious usage of this control [34].

Then, the deepening of cognition assumes effective cognizing organizations that again meet obstacles with malicious dimension of genomic nature of humans, while the transmission of cognition to machines challenges the further being of humans, at least, as a type of cellulars.

4.3. In all fairness, we should remind about the social life experience of humans lasting more than 100,000 years that inevitably reflected in their genomic patterns.

Following Confucius and his adherents, these patterns can allow us to organize consistent communities of humans addressing, particularly, to their solidarity, tolerance, patience, self-sacrifice and love.

Acknowledging, in principle, the validity of the idea of solidary being of humans, unfortunately, one cannot but should acknowledge the absence of a comprehensive constructive project of the embodiment of this idea for humankind.

5. The questions arise *whether humans being enough mentally empowered can solve the problem of either transferring to new types of negentropics free from drastic aspects of heritage of humans or find types of organization for themselves allowing, at least, to preserve the attained advances in serving energy.*

And *whether these negentropics with dominating cognition can be classified as a new type of descendants of humans, consistent with their further being or as those of conquering with them for unconditional priority.*

These fundamental, critically important questions still remain open and hopefully can be resolved with deepening cognition of ourselves and the Universe.

6. Thus, acknowledging nowadays the intractability of the problem of *DDPEN* of *doings for* quasi-eternal, *durable preservation of energized negentropicity* in the Universe, let's focus on the study of mental doings of humans, which are to a certain extent invariant wrt all types of negentropics.

Namely, let's focus on the problem of identifying the mental doings of negentropics necessary for all of them to preserve the energized negentropicity from the challenges of environments.

In other words, for the class ENs of *energized negentropics,* including not only humans, but all negentropics having sufficient technologies to access the quasi-eternal energy sources, we can state the problem IMDE PEN of *identifying the mental doings* necessary for all *negentropics* of ENs in the *preservation* of *energized negentropicity* (PEN) and the effectiveness of these doings in PEN.

6.1. To analyze the mental doings of negentropics in EN, we apparently refer to human ones as a base.

Namely, so far we have presented a modified specification of certain *constructively regularized models* (CRM) of mss, *mentals,* introduced in [47] for the class MS of mss, i.e., the class MS representing the entire mss that, we assume, can be adequate CRM for human mental doers and systems of mdoers , mss.

We assume that mdoers and mss have, at least, IDs and can be consciously processed, particularly, to explain and understand the doings/doers in communications.

The construction of mentals refers only to the existence of sensors, effectors and mental controllers of humans, as well as to their goal-oriented doing in general, without addressing the peculiarities of goals of any type of negentropics.

Thus, accepting that sensors, effectors and mental controllers are inevitable for any negentropic, we assume that CRM for humans, mentals, can be considered as CRM for all negentropics, including those of the class EN.

6.2. Scales of the effectiveness of mental doings wrt PEN are diversified and include those of coverage of classified realities, abstractness, universality and regularization of classifiers. Focusing on the scales of *constructive regularization of mental doings,* we aim to approach the solutuin of the CRMD PEN problem.

And as a step to solving CRMD, it is reasonable to study cognitive doings or *constructive regularization of cognition,* i.e., the problem CRC PEN, as well as subproblems of CRC, including the regularization of mss representing inductive and deductive inferences, acquisition, accumulation, enhancement of effectiveness, controllers and communications.

Within the framework of the above assumptions, we are going to continue to advance in adequacy of mentals for the constituents of cognizers started in [47], as well as to continue developing CRC models in [36-46].

7. To advance in CRC PEN, we need to question the range of mental doings of humans inevitable for any cognizing negentropic of EN.

Certain mental doings of humans are the subject of studying the cognition in AI in certain pragmatic human-machine contents, but the question is whether they cover the scope of those that are inevitable for negentropics of EN with cognizers estranged from celulars.

Is it necessary to regularize motivation, will, imagination, intuition, consciousness and other dimensions of human personality?

Mental patterns are consequent to the dimensions of doing of humans and include, particularly, the following types.

- *Patterns of Everyday life* include adapting to self-care, health and safety, social interactions and transactions at home, school and work , memorizing basic instructions, personal data (name, address) and important interests, goal setting and problem solving, judgments, as well as doing integrated, i.e., setting goals, making decisions, and then judging the consequences, basics of cognizing and social
- Patterns of the *Highest life* include regular cognition and communication, social relationships and norms including family, humanistic and ethical ones.

6. Conclusions

1. Concluding, let us, first, address to the impacts and challenges of AI, which include the following:

From pragmatic positions, new powerful supporters of humans, humanoids, would be constructed.

New hierarchies of professions would emerge in communities, where the top layers would be occupied by the most successful owners of new knowledge.

Ultimately, human communities can transit to those with higher control of being and doing of their members, similar to those in anthills but, in contrast, with a special enhanced attention to the cognition of the Universe. And it is not excluded that the successful being of millions of years of ants or bees, enriched with the power of cognizing in lifeime, can become the base of a new imperium of humans or “The Emperor’s New Mind” by Penrose [28].

According to another scenario, the power of e estrangement of the essentials of humans (including cognizing) into constructive models would allow humans to contain these essentials in new shells that could be more resistant to the challenges of the Universe, had advanced and diversified sources of energy supply and ,thus, be transited to a new negentropic [6,49] being.

The impact of AI on humans could be radically different for the owners of basic software of AI like Google or Facebook, and for those who only consume their services. Human knowledge and control are becoming extremely dependent on the owners of centralized and monopolized software and AI, so there is a danger for some strati to be out of all services like Wiki and Internet communications.

There is a danger of centralized AI’s personalized, out of borders and location influence on the thoughts, emotions, preferences, etc., of people, thus, manipulating their intentions and behavior.

And what about the danger that our children steering continuously the screens of gadgets full of AI manipulators of minds, will become the followers of their parents and cultures?

2. Let us also address to some enspiring aspects of AI development.

2.1. Progressing outcomes of Artificial Intelligence in constructing basic, root cognizers capable to develop to the artificial ones functionally equal to the highest human cognizers, produce significant consequences.

In fact, AI proves that realities not only of a cellular nature but also those constructed, assembled from the ordinary units of matter available to humans, can attain the highest cognizing of the Universe, at least, comparable with that of the humans.

And since these artificial, constructed highest cognizers are only the assembles of units of matter unconstrained from the mysteries of the origin of cells, it is expected to question the necessity for humans or someones else in assembling, constructing the highest cognizers.

In other words, to question the feasibility of origin of the highest cognizers within of framework of the acknowledged physics.

Aimed at enlightening the base of the above consequences and tracking some steps of their grounding, we present constructive models of cognizing meant to be consistent with ones by J. Piaget, argue the adequacy of models and the feasibility of algorithmic formation of the chains of those models that are rooted in the basic units of cognizing, the classifiers, and are incrementally rising their complexity.

Nowadays, physicists are approaching the grounding for the origination of information, and therefore, the grounding for the origination of non-entropic, or negentropic or living realities according to Schrodinger.

And since we believe that classification and information are inseparable from each other, it becomes reasonable to question the origination of elementary classifiers in Nature, and then their development to the highest cognizers.

2.3. In other words, a few fundamental, root laws of cognizing by J. Piaget lead to the lifelong cognitive development of humans.

Questioning whether these roots can be originated in Nature by the acknowledged laws of physics and whether they can convey them to the constructive self-reproduction, in what follows we present our research findings and open questions in proving the hypothesis by modeling.

Namely, we approximate the scope of acknowledged human cognizing by cognitive doings consistent with the assertions by J. Piaget and those targeted by AI.

We specify mental models/mentals and argue the adequacy of mentals in modeling human cognizing, and then make it evident they are decomposable up to the sets of the standard types of 1-/ 2- place classifiers of varying degrees of abstractness that can be formed inductively and uniformly to enrich cognizers with their assembles revealed in proper spaces or assembled by communities.

The reduction of constituents of cognizers to the basic 1-/2-place classifiers lets us narrow the search for inevitable constituents of r*C*, thus, the origination of r*R* to the search for means of formation and origination of 1-/2-place rels and their regular composition in a variety of mental systems.

Questioning the self-organization of classifiers in algorithms we argue the way of their further development by competitions in a local tournament.

3. Resuming the originations of cognizers in Nature, their development to the highest degrees of cognition letting their self-reproduction allow us to assume that cognizers can regularly be originated and constructed in Nature which resembles the regularity of origination of stars, their being, explosions followed by the birth of new stars.

References

[1] R. Feynman, *The Meaning of it All*, Addison Wesley, Massachusetts, 1998.

[2] J. von Neuman, *Theory of Self-reproducing Automata*, University of Illinois Press, 1966.

[3] T. Winograd and F. Flores, *Understanding Computers and Cognition*. A new foundation for design Publishers, Huntington, NY, 1986.

[4] J. Flavell, *The Developmental Psychology of Jean Piaget*, D. VanNostrand Comp. Inc., Princeton, N.J., 1962.

[5] Z. Pylyshyn, "Seeing and Visualizing: It's Not What You Think", *An Essay on Vision and Visual Imagination*, http://ruccs.rutgers.edu/faculty/pylyshyn.html, 2004.

[6] E. Shrodinger, *Mind and Matter*, Cambridge, 1956.

[7] J. Mandler, *The Foundations of Mind: Origins of Conceptual Thought*, Oxford Univ. Press, 2004.

[8] A. Turing, *Computing Machinery and Intelligence*, Mind 49, 1950, [Reprinted in Minds and machines. A. Anderson (ed.), Engelwood Cliffs NJ, Prentice Hall, 1964].

[9] N. Chomsky, *Aspects of the Theory of Syntax*, MIT Press, 1965.

[10] J. Pitrat, "Consciousness and conscience, in artificial beings: the conscience of a conscious machine", *ISTE*, London, UK, 2009.

[11] K Fu, *Syntactic Methods in Pattern Recognition*, London, 1974.

[12] E. Zermelo, "Ubereineanwendung der mengenlehre auf die theorie des Shachspiels", *Proceedings of the fifth International Conference of Mathematicians*, Cambridge University Press, pp. 501-504, 1912.

[13] J. Searle, "Is the brain's mind a computer program?", *Scientific American*, vol. 262, pp. 26-31, 1990.

[14] D. Roy, "Grounding language in the world: signs, schemas, and meaning cognitive machines group", *The Media Laboratory, MIT,* pp. 1-27, 2005. http://www.media.mit.edu/cogmac / projects.html

[15] C. Shannon, "Programming a computer for playing chess", *Philosophical Magazine Ser.7*, vol. 41, 1950.

[16] Ch. Brutyan, I. Zaslavski, and L. Mkrtchyan, "On methods of automated synthesis of positional strategies in games", *Problemi Kibernetiki*, Moscow, vol. 19, pp. 41-75, 1967.

[17] B Stilman, "The Zones in Games" and B Stilman, M. Tsfasman "Positional Value and Assignment of Priorities", pp. 70-104, in the book: M. Botvinnik, "About solving approximate problems", S. Radio, Moscow, (in Russian), 1979.

[18] M. Botvinnik, "Computers in chess: solving inexact search problems", *Springer Series in Symbolic Computation, with Appendixes*, Springer-Verlag: New York, 1984.

[19] A. Elo, *The Rating of Chess Players*, Past and Present, London, 1978.

[20] J. Laird, *The Soar Cognitive Architecture*, MIT Press, England, 2012. See also: John Laird, *Introduction to Cognitive Architecture*, VISCA-2020, the Virtual International Symposium on Cognitive Architecture, University of Michigan, June 4-5, 2020.

[21] (2013) UNDL Foundation, [Online]. Available: www.unlweb.net.

[22] "UNL Specifications", UNL Center of UNDL Foundation, 2005.

[23] F. Gobet, "Chunking mechanisms in human learning", *Trends in Cognitive Sciences*, vol. 5, pp. 236-243, 2001.

[24] G. Atkinson, *Chess and Machine Intuition*, Ablex Publishing Corporation, New Jersey, 1993.

[25] J. Furnkranz, "Machine Learning in Games: A Survey in "Machines that Learn to Play Games", N. Sci., 2001.

[26] J. Moon, *Topics on Tournaments*, N.Y., Holt, and Winston, 1968.

[27] R. Benergi, *Theory of Problem Solving*, Mir, Moscow, 1972.

[28] R. Penrose, *The Emperor's New Mind*, Oxford University Press, 1999.

[29] A. Markov and N. Nagorni, *Theory of Algorifms*, Nauka, Moscow, 1984.

[30] A. Maltzev, *Algorithms and Recursive Functions*, Nauka, Moscow, 1965.

[31] H. Marandjian, "A method of synthesis of programs of numeric functions", *Mathematical Problems of Cybernetics and Computer*, Yerevan, vol. 26, pp. 5-13, 1986.

[32] F. Harary, *Graph Theory*, Addison-Wesley, 1969.

[33] T. Vachaspati, L. Pogosian and Daniele A Steer , "Cosmic strings", *Scholarpedia*, vol. 10, no. 2, 31682, 2015, & Perlov D. Vilenkin A. Cosmology for the Curious. Springer, 2017.

[34] Y. N. Harari, *Homo Deus: A Brief History of Tomorrow*, Harper-Collins, 2017.

[35] H. Blavatsky, "Between Light and Darkness*", Жизнь замечательных людей* (inRussian). Moscow: Молодаягвардия, 2010.

[36] E. Pogossian and A. Martirosian, "Learning", Reference book on Intellectual Systems. Radio iSvjas Publishing Company, Moscow, (in Russian), vol. 2, pp. 206-231, 1990.

[37] E. Pogossian, "On modeling cognition", *Computer Science and Information Technologies* (CSIT11), Yerevan, pp 194-198, 2011.

[38] E. Pogossian, "Specifying personalized expertise. International Association for Development of the Information Society (IADIS)", *International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006)*, Barcelona, Spain, pp. 151-159, 2006.

[39] E. Pogossian, V. Vahradyan, and A. Grigoryan, "On competing agents consistent with expert knowledge", *Lecture Notes in Computer Science, AIS-ADM-07: The International Workshop on Autonomous Intelligent Systems - Agents and Data Mining*, St. Petersburg, Russia, June 6-7, pp. 229-241, 2007.

[40] E. Pogossian, "On measures of performance of functions of human mind", *6th International Conference in Computer Science and Information Technologies, CSIT2007*, Yerevan, pp. 149-154, 2007.

[41] E. Pogossian, "On a transparent presentation of written English syntax", *5th Intern. Cognitive Linguistics Conference*, VrijeUniversiteit, Amsterdam, pp. 209-214, 1996.

[42] E. Pogossian, "Adaptation of combinatorial algorithms", *Academy of Sci. of Armenia,* p. 293, 1983.

[43] E. Pogossian, "Effectiveness enhancing knowledge-based strategies for SSRGT class of defense problems", *NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems*, Salamanca, Spain, 2011, pp. 16.

[44] Z. Naghashyan and E. Pogossian, "Developing Java software for representation, acquisition and management of strategy knowledge", *Transactions of IIAP of NAS of RA, Mathematical Problems of Computer Science,* vol. 33, pp.187-195, 2010.

[45] K. Khachatryan and S. Grigoryan, "Java programs for matching situations to the meanings of SSRGT games", *Proceedings of SEUA Annual Conference*, Yerevan, Armenia, pp. 135-141, 2013.

[46] S. Grigoryan, "Research and Development of Algorithms and Programs of Knowledge Acquisition and Their Effective Application to Resistance Problems", PhD, p 111, Yerevan, Armenia, 2016.

[47] E. Pogossian, "Towards adequate constructive models of mental systems", *12th International Conference in Computer Science and Information Technologies, CSIT2017*, Yerevan, pp.96-101,2017, as well as IEEE's Xplore electronic library and is available at the link :http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8307363

[48] E. Pogossian, S. Grigoryan and N. Hakobyan, "On systemic classification and machine learning", *12th International Conference in Computer Science and Information Technologies, CSIT2017*, Yerevan, pp.102-107, 2017.

[49] E. Pogossian, “On the way to dominating cognition”, *Transactions of IIAP of NAS of RA, Mathematical Problems of Computer Science,* vol. 49, pp.79-91, 2018.

[50] S. Grigoryan, N. Hakobyan and H. Vrtanesyan, “Object–oriented Modeling of Matching to Systemic Classifiers”, *Transactions of IIAP NAS RA Mathematical Problems of Computer Sciences*, vol. 49, pp. 115-122, 2018.

[51] E. Pogossian, A. Javadyan and E. Ivanyan, “Effective discovery of intrusion protection strategies,” *The International Workshop on Agents and Data Mining, Lecture Notes in Computer Science*, St. Petersburg, Russia, pp. 263-274, 2005.

[52] V. Germeier, *Games with the Nature*, (in Russian), Moscow, Phis-math publishing, 1980.

[53] Thomas B. Sheridan, *What Is God? Can Religion Be Modeled?* New Academia Publ. 2014

[54] E. Pogossian, “Constructing adequate mental models”, *Transactions of IIAP of NAS of RA, Mathematical Problems of Computer Science,* vol. 50, pp.35- 42, 2018.

[55] Discrete Mathematics. Edited by S. Yablonski and O. Lupanov, “Naula”, Moscow, 1974.

[56] V. Vapnik, A. Chervonenkis. The Theory of Pattern Recognition, “Naula”, Moscow, 1974

[57] E. Pogossian, M. Hambartsumyan and Y. Harutunyan, *A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations*, National Academy of Sciences of Armenia, IPIA, (1974-1980 research reports).

[58] B. Stilman, “Hierarchy of Formal Grammars for Solving Search Problems, in Artificial Intelligence. Results and Prospects”, *Proceedings of the International Workshop*, Moscow, (in Russian), pp. 63–72, 1985.

[59] J. M. R. Parrondo, J. M. Horowitz and T.Sagawa, “Thermodynamics of information”, Nature Physics, vol. 11, pp. 131-139, 2015.

[60] E. V. Koonin, *The Logic of Chance: The Nature and Origin of Biological Evolution*. Pearson Education, Inc. Publishing as FT Press Science, 2012

[61] Artificial Neuron Nets, https://en.m.wikipedia.org/wiki/Artificial_neural_network

[62] “From Matter to Life”, Edited by S.I. Walker, P.C.W. Davies, G.F.R. Ellis, Cambridge University Press, 2017, 494 pp.

[63] M. Eigen*, Selforganization of Matter and the Evolution of Biological Macromolecules*, Springer, 1971.

[64] E. Pogossian, “Challenging the uniqueness of being by cognizing*”, Transactions of IIAP NAS RA, Mathematical Problems of Computer Sciences*, vol. 51, pp 66-81, 2019.

[65] E. Pogossian, “On assessment of performance of systems by combining on-the-job and expert attributes scales”, *International Conference in Computer Sciences and Information Technologies*, Yerevan, Armenia, pp. 331-335, 2015.

[66] B. Karapetyan, “A High-Level Language for Chess Concepts”, *Mathematical Problems of Computer Sciences*, 1986

[67] E. Arakelova and E. Pogossian, “Tools for testing and correction of the completeness of knowledge acquisition by autistic children,” *8th International Conference in Computer Science and Information Technologies*, CSIT2011, Yerevan, Armenia, pp. 159-165, 2011.

[68] Z. Naghashyan, “Developing software for formation and acquisition lexical units in certain limited languages”, PhD thesis, Yerevan, 2010.

[69] E. Pogossian, S. Grigoryan and L. Berberyan, “Personalized Interactive Tutoring in Chess”, *Didactics in Chess, London Chess Conference, December 10-13,* 2016.

[70] E. Danielyan and E. Pogossian, "On expanding of method of local measurement to evaluation of management strategies", *Proc. of the Ann. Conf. of the State Engin. Univ. of Armenia*, 2009.

[71] K. Khachatryan, S. Grigoryan and T. Baghdasaryan, "Experiments validating the Be-Have-Do meaning presentation model and matching algorithm for competing and combating problems", *Proceedings of International Conference in Computer Science and Information Technologies*, pp. 155-159, Yerevan 2013.

[72] K. Khachatryan and S. Grigoryan, "Java Programs for Presentation and Acquisition of Meanings in RGT Games", *Proceedings of SEUA Annual Conference*, pp. 135-141, Yerevan 2013.

[73] S. Grigoryan, "Structuring of goals and plans for personalized planning and integrated testing of plans", *Transactions of IIAP NAS RA, Mathematical Problems of Computer Sciences*, vol. 43, pp. 62-75, Yerevan, Armenia 2015.

[74] S. Grigoryan, "On valiidity of personalized planning and integrated testing algorithms in reproducible games", *Proceedings of International Conference in Computer Sciences and Information Technologies*, pp. 317-321, Yerevan, Armenia, 2015.

[75] N. Hakobyan, "A system for transforming images to symbolic presentation for combinatorial defense and competition problems", *Proc. of the RA NAS and NPUA Ser. of Tech. Sc.*, vol. 72, no. 2, pp. 199-209, 2019.

[76] K. Khachatryan, S. Grigoryan and T. Baghdasaryan, "Experiments validating the Be-Have-Do meaning presentation model and matching algorithm for competing and combating problems", *Proceedings of International Conference in Computer Science and Information Technologies*, pp. 155-159, Yerevan 2013.

[77] S. Grigoryan and N. Hakobyan, "Experimenting with acquisition of and matching to systemic classifiers", *Transactions of IIAP NAS RA, Mathematical Problems of Computer Sciences*, vol. 5, pp. 96-103 , 2018.

[78] E. Pogossian, "Artificial Intelligence: Alternating the Highest Human Cognizing", *13th International Conference in Computer Science and Information Technologies*, CSIT2019, Yerevan, 2019, pp. 36-46, as well reprinted in IEEE's Xplore: https://ieeexplore.ieee.org/xpl/conhome/8890749/proceeding

[79] S. Grigoryan, N. Hakobyan and T. Baghdasaryan, "Knowledge-based solvers for rgt combinatorial problems", *12th International Conference in Computer Science and Information Technologies*, CSIT2019, Yerevan, 2019, pp. 36-46 , as well reprinted in IEEE's Xplore https://ieeexplore.ieee.org/xpl/conhome/8890749/proceeding

[80] E. Pogossian, D. Dionne, A. Grigoryan, J. Couture and E. Shahbazian, "Developing goals directed search models empowering strategies against single ownership air threats", *International Conference in Computer Sciences and Information Technologies*, pp. 155-163, Yerevan, 2009.

[81] D. Dionne, E. Pogossian, A. Grigoryan, J. Couture and E. Shahbazian, "An optimal sequential optimization approach in application to dynamic weapon allocation in naval warfare" , *11th International Fusion 2008 Conference in Cologne*, July 1-3, 2008

Contents of Appendices

Appendix 1: Solvers in General

1. Pogossian, "**Effectiveness enhancing knowledge-based strategies for SSRGT class of defense problems**", NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems, Salamanca, Spain, 2011, pp. 16.

(see Chapter 3 of Part 1 for the details)

2. E. Pogossian, V. Vahradyan, A. Grigoryan. **On Competing Agents Consistent with Expert Knowledge**", Lecture Notes in Computer Science, AIS-ADM-07: The Intern. Workshop on Autonomous Intelligent Systems - Agents and Data Mining, June 5 -7, 2007, St. Petersburg, 11pp.

Abstract.
We aim to advance in constructing collaborative agents able to acquire the contents of human vocabulary associated with competitions. Refining the framework and criteria of performance of agents we project the study on the class of game tree represented competition problems. For known representative of the class – chess like combinatorial games, we categorize the contents of a comprehensive repository of units of chess vocabulary by formal structures of attributes, goals, strategies, plans, etc. We define Personalized Planning and Integrated Testing algorithms able to elaborate moves in target positions dependent on those categories of chess knowledge. We then demonstrate the effectiveness of the algorithms by experiments in acquisition the solutions of two top Botvinnik's tests – the Reti and Nodareishvili etudes. For min max game tree-based search algorithms these etudes appear to be computationally hard due the depth of the required analysis and very dependence on the expert knowledge.
Keywords: competing, agents, knowledge, acquisition, consistent, measures.

1. Introduction

The concept of agents is based on the idea of assistance with tasks that require reciprocal and consistent human-agents knowledge processing.

We plan to advance in constructing collaborative agents able to improve their problem-solving performance by acquiring, learning or inferring the contents of human vocabulary associated with competitions.

We develop the approach announced in [18] and study problem solving for the subclass of optimal strategy provision problems where the space of hypothesis of solutions can be specified by game trees (SSGT). Viability of the approach was demonstrated for the intrusion protection SSGT problems and the Intermediate Goals At First (IGAF) algorithms. The IGAF algorithms as well as [21] are based on a common knowledge planning and dynamic testing of the plans in the corresponding game trees. It was proven that the IGAF2 algorithms are able to acquire a range of expert knowledge in the form of goals or rules and to increase the efficiency of strategy formation with an increase in the amount of expert knowledge available to the algorithm.

The effectiveness of the IGAF2 algorithms was successfully tested for the network intrusion protection problems against representatives of four classes of attacks: SYN-Flood, Fraggle, Smurf and Login-bomb, allowing to formulate, in particular, the following statements:

- Number of nodes searched by the IGAF2 algorithm for making decisions with all expert rules and subgoals is the smallest compared with the IGAF1 algorithm or with the minimax algorithm in which the depth of the search is increasing up to 13
- The recommended version of the IGAF2 algorithm with all expert rules and subgoals, for the depth of search 5 and 200 defending steps, is outperforming the productivity of the minmax algorithm by 14% while using 6 times less computing time and searching 27 times less nodes of the tree.

Knowledge based strategies for another SSGT problem where supply chain management agents compete in a market to gain max profit for different customer requests were studied in [20, 22]. For each request the agents generate possible offers and for each offer a game tree is constructed to allow simulation and tracing further actions with suppliers, production and delivery. In the mean time, strategic plans based on the handbook common expert knowledge are generated followed by the quantification of the plans for strategy analysis and their evaluation for final decision making.

In the paper regular improvement of programs by expert knowledge is studied for known representatives of the SSGT class – chess like combinatorial games [19], where the problems of knowledge representation and consistent inclusion into the programs stay central since the pioneering work by Shannon in 1950.

Players indicate and communicate chess knowledge by units of vocabulary and are able to form corresponding contents. Whether is it possible to form equal contents by computers remains under question.

A comprehensive overview of the state of the art in learning the contents of vocabulary in games is presented in [12]. Models of chess concepts based on the elements of chess contents – chunks, are discussed in [13, 14] to answer whether the chunks are common for all players or are individualized. These problems are analogous to other semantics related problems, particularly to understanding instructions by robots [25, 31], ontology formation, etc. A comprehensive review of ontology based studying is presented in [6,7].

The approaches to regular inclusion of chess knowledge into strategy formation process are described in [3, 4, 23, 30]. All of them try to bring common handbook knowledge to cut the search in the game tree. The frontiers of those approaches can be revealed by understanding the role and proportion of the personalized chess expertise compared with the common, communicable one.

In 1912 Zermelo proved that all chess positions are strongly divided into three classes: winning, losing or drawing [32]. In [16, 17] a correspondence was revealed between Zermelo's classes of winning chess positions and strategies and a typical units of chess vocabulary. While this correspondence supports the constructive nature of human models of chess realities it, at the same time, states that the real implementation of those models, in principle, can only approximate (to a specification) the winning game tree structures due to the irresistible complexity of computations required to prove the correctness of the models. Thus, it follows that

i. for the same units of chess vocabulary both chess players and computers will, as a rule, use different models of chess realities essentially based on their individual experience

ii. any preferences between those models include uncertainty

iii. interpretations of the units of vocabulary have to rely on some personalized experience.

Therefore, strategy formation systems have to be able to integrate common chess knowledge with personal expertise of particular players and be able to improve their performance by regular acquisition, learning or inferring the contents of the units of chess vocabulary.

The paper reports our progress in programs acquiring typical categories of chess knowledge and is structured as follows. Sections 2 and 3 define the SSGT problems, refine the framework of study and measures of effectiveness of competing programs to induce corresponding requirements to them. Section 4 describes formal structures of the attributes, goals, strategies, plans, etc., and our approach to categorize a repository of chess vocabulary in about 300 units [24] by those types. Section 5 specifies the *Personalized Planning and Integrated Testing* (PPIT) algorithms able to elaborate moves in target positions dependent on typical categories of chess knowledge and demonstrates the effectiveness of PPIT algorithms by experiments in acquisition expert knowledge based solutions of two top Botvinnik's test etudes in strategy discovery, the Reti and Nodareishvili etudes. The Conclusion underlines the main findings of the research.

2. Optimal Strategy Provision Problems

We define the SSGT problems as a constructive subclass of a spacious class of *Optimal Strategy Provision* (OSP) problems [19, 22]. The OSP problems, first of all, meet the following requirements:

a. there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions in the (c) specified moments of time and (d) specified types of situations
b. there are identified benefits for each of the actors
c. there are descriptions of the situations the actors act in and transformed after actions.

For such problems with given arbitrary situation x and actor A, who is going to act in x, we can generate corresponding *game tree* GT(x, A) comprising all *games* started from x.

The games represent all possible sequences of legal actions of the players and situations that they can create from given initial, or root situation x. In our consideration the games are finite and end by one of goal situations of the problem.

In chess, for example, the actors are white and black players with checkmate as the goal, chess piece moves are (contra) actions and compositions of chess pieces on the board determine specific game situations (positions) transformed by actions according to chess rules in the corresponding game tree.

Assuming A plays according to a deterministic program, a *strategy*, the GT(x, A) represents, in fact, all possible *performance trees* of the strategies from the x. In that sense the GT(x, A) determines the space of all possible solutions from the x situation.

Note, that performance trees of the strategies are described not on the level of detailed commands but by their compositions, i.e. the actions of the players.

Given criterion K to evaluate the quality of strategies we can define the best strategy $S^*(x, A)$ and corresponding best action of A from x.

In the OSP problem with criterion K of quality of strategies it is required to find the best action in respect to K for any given situation.

The OSP problems comprise chess-like combinatorial problems, security and competition problems, particularly, network intrusion protection and management in oligopoly competitions problems. Many other games represented security problems such as Computer Terrorism Countermeasures, Disaster Forecast and Prevention, Information Security, etc., announced by the NATO (http://www.nato.int/science/e/ newinitiative.htm) seemingly can be reduced to OSP problems, as well.

The SSGT class comprises the OSP problems where the OSP requirement to have descriptions of situations after transforming them actions is replaced by the following stronger requirement:

a. the situations the actors act in and transformed after actions can be *adequately simulated.*

Thus, for SSGT problems the game trees can be constructively simulated what allows to create a common theory and computer-based game tree analysis methodology to find optimal solutions for corresponding problems.

In [19, 21, 22, 23] it was proved that chess and chess like combinatorial problems, intrusion protection and oligopoly competition are SSGT problems and common methodology can be effectively applied to find high quality solutions for them.

3 . Refining Strategy Evaluation Measures

The framework of any study grounds on appropriate measures determined by the nature of studied problems and persuaded goals.

Concentration on the SSGT class of problems followed by focusing the chess permits to use measures of program performances induced by problems under consideration which, in contrast with universal ones, allow to control adequacy of measures by utilities of corresponding solutions.

We focus on measures of the *effectiveness* vs. *efficiency* and *On-the-job* (OJP) performance measures [20] vs. *Attributive Scoring of Preference* (ASP) based ones [9].

ASP measures evaluate performance of systems by aggregating the values of different attributes defined for different aspects of the performance into a global quality indicator (called the global preference).

Despite the fact that ASP measures are widely used in education, personal evaluation, systems and organizations evaluation, etc., they have a principle drawback in being very dependent on human subjective intervention in the measurement process. Particularly, they suppose a human choice of the base general criteria of functionality of examined systems, their partition on the subsystems with appropriate attributes and scales to evaluate them, and, finally, choice of a method to integrate measurements of the subsystems into global estimate of the system performance.

The OJP measures assume to be functions of quality of solutions of tasks comprising those problems. In other words, they have to be functions of "how" the solutions perform their "job" for individual tasks of those problems.

We distinguish two types of OJP measures based either on *question-answers* or on *tasks solution values.* The Turing Test (TT) [29] belongs to the first type and because it is not a formal procedure can cause uncertainties and drawbacks in its applications, like the following ones:

a. TT procedure is very dependent on subjective characteristics of the judges and cannot be regularly reproduced

b. TT in spite of measuring solutions of individual problems does not use their internal criteria and, instead, pretends to be a universal measure

c. TT is based on local questions –answers to the solutions of problems and needs a framework of their correct integration into global estimates of the quality

d. TT compares the "external" question-answers "shell" of the solutions and does not measure actions in the world caused by the solutions, or is not "grounded" [25, 27, 28].

The *tasks solution values* (TSV) criteria are long time used to measure human and program performance. Based on the analysis of performance trees TSV criteria are widely used for testing of programs [15].

In chess TSV criteria coincide with a widely applied tournament-based ones [10, 26]. The OJP criteria for measuring performance of chess programs, management and intrusion protection skills were developed in [20-23]. For further studying we chose TSV OJP measures of effectiveness of solutions.

Finally, we need *human-computer consistent* measures, allowing to measure human and program performances in problem solving at the same scale.

Particularly, to measure at the same performance scale an ability to incremental growth of the quality of performance by acquisition of expert knowledge.

At present, Elo's rating system is in the use for chess players [10]. Therefore, computer performance measures consistent with Elo ratings are needed.

We argue that max Sum criteria of strategy performance can play that role.

Indeed, the vectors of the terminal nodes of examined strategies can be comprised into a matrix and optimal strategies can be determined by comparing those vectors by criteria that are functions of the results of the analysis of the values of components of those vectors. For max Sum criterion those values are summarized which makes the criterion equal to the robin, absolute tournament among all possible strategies and isomorphic to the ordering induced by Elo ratings. Thus, we can assume that max Sum ordering is sensitive to knowledge acquisition and changes of the players ranks in the ordering is directly proportional to changes of their Elo ratings.

But what are the relationships between changes in locations of programs in the max Sum ordering and inclusion of chess knowledge in their work?

We state the following:

- In contrast with chess players, rising the dispositions of programs in the max Sum orderings is not necessary caused by possessing more chess knowledge and skill than the lower positioned programs
- If changes of positions of programs in the max Sum ordering *ceteris paribus* are caused by knowledge acquired by programs from the hierarchies of common chess knowledge (e.g. from handbooks) then those changes are measuring the rising of quality of programs caused by that knowledge acquisition and can be compared with a human ones by the same max Sum criterion.

Therefore, if we test procedures of human knowledge acquisition and max Sum criterion indicates regular increasing of performance of programs simultaneous with the work of the procedures then the assumption that those procedures simulate human knowledge acquisition gets a valuable argument.
Concluding, we accept max Sum criterion as a common for chess players and programs measure to evaluate the quality of acquisition, learning or inferring chess knowledge.

4. Categorization of Chess Expert Knowledge

To represent chess knowledge, we classify the units of chess repository (UCR) [24] in frame of known models and constructive views followed by assumptions on their origin.
The 1st step of UCR classification reveals elements of specification of chess as a game which in turn induces new game analysis-based models as the following:

- game tree model adequately representing chess, all possible behaviors in it – strategies, concept of the best strategy, etc.
- strategy search methods and methods to enhance their efficiency
- Zermelo's statement on chess positions partition as winning, losing and drawing ones.

On the 2d step of UCR classification those models and their constituents were classified as conceptual and procedural types of strategy knowledge, represented, particularly, by attributes, classifiers and rules and with meanings referred to the elements of game tree, chess specification components or other not chess elements [16].
Attributes are defined as a kind of operators having values in a given set of realities. Particularly, they may be sets of numbers of some ranges.
We distinguish basic *attributes* and constructed ones that compose basic attributes to determine their values. Attributes with 0/1 or "false" / "true" values are named *classifiers* and correspond to concepts and goals.
Concepts are defined as a kind of knowledge to identify and recognize realities whereas goals are concepts with elements of intensions or requirements to achieve them.
Motives are attributes used to argue the preferences of some goals in analyzed situations.
Strategies, plans, goals and *rules* are defined as expert knowledge to specify compositions of his actions in time.
We name (c*omplete*) *S-strategy of C* the performance of the player/competitor C (i.e. the performance of corresponding program) in the game tree with starting position S (S-tree).
Rules are kind of if *x* / then *y* operators to specify a procedure *y* for the realities that fit to the *x* requirements. Any strategy may be determined as a composition of rules.

Given criteria F we say that *a strategy G achieves the goal F* if criteria F are satisfied for the set of terminal nodes of G.

The *strategy G will be called F-projected* if we are interested in whether the terminal nodes of G satisfy criteria F or not.

Any description of an F-projected strategy G aimed to make the search of G more efficient is named a *strategy plan for F* [19].

Descriptions of problems and strategies resolving them are extreme examples of strategy plans. A useful strategy plan would systematically identify the directions that are not promising and eliminate them for reducing the search space.

Strategy planning is a process of narrowing the search space for the target strategy which also reflects the specifics of the planner such as knowledge of language, the system of values and methods of search.

We identify the following two sources of strategy expertise represented by UCR:

- concepts and attributes determined by specification of the game tree
- winning strategies induced by examining the game tree.

Let us illustrate our reasoning to link chess concepts with classes winning by Zermelo (for the defeat and draw ones similar links can be induced by analogy).

In general, a position is winning if a winning strategy can be generated with the root in that position. The "Mat" positions are the only exclusion where winningness is determined by a few game specification attributes calculated either statically or by one-two plies. The attributes are the following:

K(ing) is under check, K can't escape, K has not defenses.

Being constituents of the concept "Mat", those attributes get their own values of winningness induced by the value of "Mat" winning positions – the maximum value in chess.

It is evident, that any utility of winningness of positions provides not only two extreme "yes" or "not" utility values of those attributes but also values intermediate between those two poles.

Similar reasoning is natural to spread, analogically, to other than King pieces as well as to squares where pieces may be located.

Expanding the ranges of values and definitions of the attributes in a natural way we get the scales where not only their max and min values but intermediate ones, too, become meaningful respective to utilities they are associated with.

Another source for valued attributes and concepts are positions with known values, winning or not: endgames, combinations, etudes, beginnings, "tabbies" – typical analyzed positions, etc. [23] and, possibly," chunks" [13, 14]. Partitioning and generalizing their constituents we get a set of basic concepts, attributes with associated estimates of their winningness. These attributes applied to constituents (or configurations of constituents) of analyzed positions create estimates of their utility in the context of winningness of those positions.

Analysis of chess concepts of the Repository allows to identify them as the descriptions of elements of positions having tangible winning utility. Hence, chess concepts become elements of specifications of winning by Zermelo positions what argues for possibility of their simulation. The players integrating in some reasonable way the utilities of constituents of positions get estimates of winningness of the positions.

What configurations of elements of positions are covered by concepts and how that is happening needs a special studying, particularly, in view of the theory of "chunks"?

The concepts can only indicate some possible utilities for further deeper analysis and strong estimate. An uncertainty is a priory in their nature caused partly by limitations of chess players' static or quasi dynamic analysis of the winningness of positions and partly as a consequence of an individualized way in formation of concepts.

It is worth to accept that using the same names for the concepts chess players along with common strong meaning concepts use many other ones having, in general, not equal interpretations which coincide only in some "skeleton" parts. Those uncertain concepts, as a rule, don't create big casualties due to their intermediate preliminary role of hypothesis of winningness of positions which later on are strongly tested by explicit game tree-based analysis for final decisions.

5. Experiments in Acquisition Knowledge Intensive Solutions

Let's describe the class of *Personalized Planning and Integrated Testing* (PPIT) programs aimed to acquire strategy expert knowledge to become comparable with a human in solving hard chess problems. In fact, the following two tasks of knowledge acquisition can be identified the process:

- construction of shells of the programs allowing to acquire the contents of units of chess vocabulary and
- construction of procedures for regular acquisition of the contents of the units by the shells.

We formulate the limitations in designing effective shells as following:

- be able to store typical categories of common chess knowledge as well as the personalized one and depend on them in strategy formation
- be able to test approximate knowledge-based hypothesis on strategies in questioned positions by reliable means, for example, using game tree search techniques.

The second task we plan to solve in the following two stages:

- to prove that the shells, in principle, can acquire the contents of units of vocabulary used by chess players and allow to tune them properly to solve expert knowledge intensive chess problems
- to develop procedures for regular acquisition of the contents of those units.

The paper reports our progress in the first stage.

6.1 Personalized Planning and Integrated Testing Programs

We design shells of PPIT1 programs as a composition of the following basic units:

- Reducing Hopeless Plans (RHP)
- Choosing Plans with Max Utility (CPMU)
- Generating Moves by a Plan (GMP)

Given a questioned position P1 and a store of plans, RHP recommends to CPMU a list L1 of plans promising by some not necessarily proved reasons to be analyzed in P1. The core of the unit is knowledge in classification of chess positions allowing to identify the niche in the store of knowledge the most relevant for analysis the position. If the store of knowledge is rich and P1 is identified properly it can provide a ready-to-use portion of knowledge to direct further game playing process by GMP unit. Otherwise, RHP, realizing a reduced version of CPMU, identifies L1 and passes the control to CPMU.

CPMU recommends to GMP to continue to play by current plan if L1 coincides with list L0 of plans formed in the previous position P0 and changes in P1 are not essential enough to influence on the utility of current plan.

If changes in P1 are essential, CPMU analyzes L1 completely to find a plan with max utility and to address it to GMP as a new current plan. Otherwise, CPMU forms new complementary list L1/ L1*L0 from the plans of L1 not analyzed, yet, in L0, finds a plan with the best utility in that list and comparing it with utility of current plan recommends the one of them with a higher utility.

To calculate utilities of the attribute, goal and plan type units of chess knowledge, we represent them as operators over corresponding arguments as follows:

-for basic attributes the arguments are characteristics of the states of squires in the questioned positions, including data on captures of pieces, threats, occupations, etc.

- for composed attributes, including concepts and goals, the arguments are subsets of values of basic attributes relevant to the analyzed positions
- for plans the arguments are utilities of the goals associated with realization of those plans.

Utilities of arguments of basic attributes are calculated by trajectory-zones based technique (TZT) [3, 18, 21] originally suggested to estimate utilities of only captures of the opponent pieces. For example, to choose capture with max utility TZT chains the moves to each piece of the opponent (trajectories) without accounting possible handicaps for real capturing then using all available knowledge "plays the zones” of the game tree induced by the trajectories followed by estimation of their values to choose the best.

The utility of units of knowledge the operators assemble from utilities of corresponding arguments in some predetermined ordering. Thus, each operator can provide by a request the arguments which are analyzed at the moment.

For example, realizing current plan the shell can determine the goal in the agenda which in turn determines basic attributes to be considered followed by indication of the arguments of those attributes.

Utility estimation operators rely on the principle of integration of all diversity of units of knowledge the shell possess at the moment. In fact, the operators represent a kind of expert knowledge with a variety of mechanisms and leverages to make them better. Along with dynamically changed parametric values of pieces they can include rules, positions with known values and strategies to realize them, other combinatorial structures. To estimate expected utilities the operators, take into account the cost of resources necessary to get them.

In current C++ realization the units of knowledge are realized as OO classes with specialized interfaces for each type of knowledge and one common for the shell itself.

We experiment in solving Reti and Nodareishvili etudes (Fig.1, 2) requiring by Botvinnik [3, 4] intensive expert knowledge-based analysis not available to conventional chess programs.

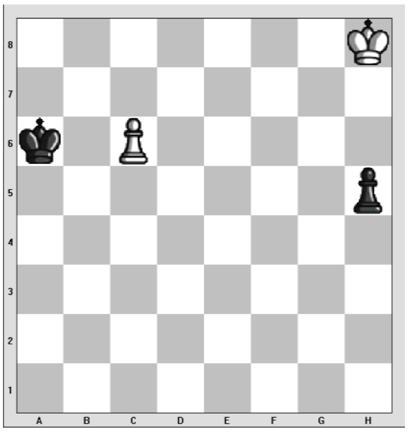


Fig.1. Reti etude: draw

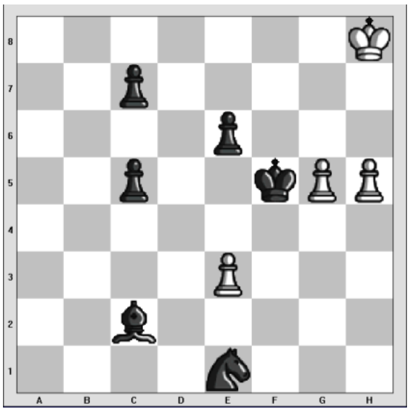


Fig.2. Nodareishvili etude: winning.

The following results can be reported at the moment.

6.2 Reti and Similar Etudes

Because Reti etude is a unique and handbook chess problem it is supposed to be in the knowledge store of the shell. RHP unit classifies it as a pawn endgame and suggests a known plan for the white king:

Realize G4 and if G1, or G2, or G3 realize them as well,

where goals G1, …, G4, listed by their priority, have the following contents:

G1. Capture the pawn

G2. Move forward the pawn at passant

G3. Protect your pawn

G4. Be max close to both: your and opponent pawns.

Because the plan is known the control is passed to GMP unit which acts as follows.

Based on the TZT the GMT unit determines utilities of attributes for each goal, finds the goal with max utility and makes the move induced by that goal.

Several other etudes induced by the original Reti one by adding new pawns in different parts of the chess board were considered.

In all experiments by adding new rules it was possible to focus the analysis of the program on the essential Reti part of the chessboard and successfully resolve the position.

6.3 Nodareishvili Etude

Nodareishvili etude is a unique and known chess problem and RHP classifies it as a pawn endgame as well but suggests four corresponding plans for different stages of the game.

The goals for the plans have the following contents:

G1. Move forward pawns at passant
G2. Protect your pieces
G3. Occupy or protect the most important squires
G4. Identify infinite check
G5 Avoid infinite check
G6. Decrease suppression of your king
G7. Capture the queen
G8. Find opponent pieces on the same line with its king
G9. Attack the king
G10.Attack the pieces on the line with the king
G11.Occupy the last horizontal line
G12. Approach king to your queen
G13. Be opposite to the pawn closest to opponent king.

The plans for experiments were presented as follows:

Plan1: Do G1and G6
Plan2: Do G1and if G4 do G5 and G6
Plan3: If G8 do G9 and G10
Plan4: Do G11 and G12 and G13.

After several corrections and tuning of utility estimation procedures the shell, in principle, was able to acquire the above units of knowledge, to choose corresponding plan for each stage and realize it.

Along with mentioned positive knowledge acquisition ability the efforts to formalize the units of knowledge for storing in the shell were too time consuming to rely only on those pure knowledge acquisition forms to solve knowledge intensive chess problems.

6. Conclusion

Currently the development of agents is facing a principal difficulty achieving a consistency and compatibility in the way agents and humans use and exchange knowledge.

In this paper we have reported our progress towards developing human knowledge acquisition by competitive agents that elaborate strategies.

We define a class of competitive problems with a wide range of applications. For chess like combinatorial games, which are long-recognized representatives of this class used for in-depth studies, we determine a measure common for chess players and programs – the so-called max Sum criterion. It can be used to evaluate the quality of acquisition, learning or inferring chess knowledge.

Classifying categories of chess knowledge according to the formal structures of attributes, goals, strategies, plans, etc., we define the class of Personalized Planning and Integrated Testing programs able to elaborate moves, in a given position, dependent on those categories of chess knowledge.

We demonstrate the effectiveness of PPIT1 programs by experiments in acquisition of knowledge for two etudes which are intractable for common chess programs -- the Reti and Nadareishvili etudes suggested by Botvinnik to test the quality of knowledge intensive solutions in computer chess.

We find that PPIT1 programs, in principle, are able to acquire the contents (meanings) of units of vocabulary used by chess players and allow one to tune the contents properly to acquire the solutions of those etudes and play them properly.

In addition, we conclude that pure knowledge acquisition-based approaches to solve problems consume too much time to formalize chess knowledge properly to be used in the programs and must be complemented by knowledge learning and inference-based methods.

References

[1] Adelman L. Riedel S.: Handbook for Evaluating Knowledge–Based Systems, Kluwer Acad. Publ. (1997)

[2] Atkinson G.: Chess and Machine Intuition. Ablex Publishing Corporation, Norwood, New Jersey (1993)

[3] Botvinnik M.: About solving approximate problems, Sov. Radio, Moscow, (in Russian) (1979)

[4] Botvinnik, M.: Computers in Chess: Solving Inexact Search Problems. Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag: New York (1984)

[5] Baghdasaryan T., Danielyan E, Pogossian E.: Supply Chain Management Strategy Provision by Game Tree Dynamic Analysis International Conference: Management of Small and Medium Business: Information Technologies (SBM2006), Sevastopol, Sept. 3-8, (2006) 37-41

[6] Gorodetski V., Kotenko I.: Attacks against Computer Network: Formal Grammar Based Framework and Simulation Tool. Proc. of the 5 Intern. Conf. "Recent Advances in Intrusion Detection", Lecture Notes in Computer Science, v.2516, Springer Verlag (2002) 219-238.

[7] Gorodetski V., Kotenko I., Karsaev O.: Framework for ontology-based representation of distributed knowledge in multiagent network security system. Proc. of the 4-th World Multi-conf. on Systems, Cybernetics and Informatics (SCI-2000), vol. III, "Virtual Engineering and Emergent Computing", Orlando, USA, July (2000) 52-58

[8] Dhar V, Stein R.: Seven Methods for Transforming Corporate Data into Business Intelligence, NJ, Prentice-Hall (1997)

[9] Dujmovic J.: A Method for Evaluation and Selection of Complex Hardware and Software Systems. The 22nd Inter.Conf. for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG 96 Proceedings, Vol. 1 (1996) 368-378.

[10] Elo A.E.: The Rating of Chess players, Past and Present. London, Batsford (1978)

[11] Flavell J.: The Developmental Psychology of Jean Piaget, D. Van Nostrand Company Inc., Princeton, New Jersey (1962).

[12] Furnkranz J.: Machine Learning in Games: A Survey in “Machines that Learn to Play Games”, Nova Scientific (2001).

[13] Gobet, F., Simon, H.A.: Expert chess memory: Revisiting the chunking hypothesis. Memory, 6 (1998) 225-255.

[14] Gobet, F.: Chunking mechanisms in human learning. Trends in Cognitive Sciences, 5, (2001) 236-243.

[15] Kaner C., Falk J., Nguen Q.N.: Testing Computer Software. International Thomson Computer Press (1993).

[16] Pogossian E.: Specifying Personalized Expertise. International Association for Development of the Information Society (IADIS): International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006), 8-10 Dec., Barcelona, Spain (2006) 151-159.

[17] Pogossian E.: On Understanding of Human Requests by Computers. The Twenty-Eighth Annual Conference of the Cognitive Science Society, July 26-29, 2006, Vancouver, Canada (2006) 221-223.

[18] Pogossian E., Javadyan A., Ivanyan E.: Effective Discovery of Intrusion Protection Strategies. The Intern. Workshop on Agents and Data Mining, St. Petersburg, Russia, Lecture Notes in Computer Science, Vol. 3505 (2005) 263-274.
[19] Pogossian E.: Combinatorial Game Models for Security Systems. NATO ARW on "Security and Embedded Systems", Porto Rio, Patras, Greece, Aug. (2005) 8-18.
[20] Pogossian E.: Business Measurements by On-The-Job Competition Scales. International Conference "Management of Small Business: Problems, Teaching, Future", Sept. 20th-24th, Sevastopol (2004) 31-35.
[21] Pogossian E. Djavadyan A.: A Game Model for Effective Counteraction Against Computer Attacks in Intrusion Detection Systems, NATO ASI, Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management", Tsahkadzor, Armenia, August 19-30 (2003) 823-851.
[22] Pogossian E.: Focusing Management Strategy Provision Simulation. Proceedings of the CSIT2001, 3d Inter. Conf. in Comp. Sci. and Inf. Technologies, Yerevan (2001) 37-42.
[23] Pogossian E.: Adaptation of Combinatorial Algorithms. Academy of Sciences of Armenia, Yerevan (1983) 1-293 (in Russian).
[24] Pogossian E., Hambartsumyan M., Harutunyan Y.: A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations. National Academy of Sciences of Armenia, IPIA, (research reports 1974-1980) (in Russian) 1-55
[25] Roy D.: Grounding Language in the World: Signs, Schemas, and Meaning Cognitive Machines Group, The Media Laboratory, MIT http://www.media.mit.edu/cogmac / projects. html) (2005) 1-27.
[26] Sadovski L.E., Sadovski A.L.: Mathematics and Sport. Moscow, Fiz-Mat Literature (1985).
[27] Searle J.: Is the brain's mind a computer program? Scientific American 262 (1990) 26-31.
[28] Searle J.: Minds, brains, and programs. Behavioral and Brain Sciences 3 (3) (1980) 417-457.
[29] Turing A.M.: Computing Machinery and Intelligence. Mind 49 (1950) [Reprinted in Minds and machines. A. Anderson (ed.), Engelwood Cliffs NJ, Prentice Hall (1964) 433-460].
[30] Wilkins D.E.: Using Patterns and Plans in Chess. Artificial Intelligence, v.14 (1980) 165- 203.
[31] Winograd T., Flores F.: Understanding Computers and Cognition. A new foundation for design Publishers, Huntington, NY (1986) 11–59.
[32] Zermelo E.: Uber eine Anwendung der Mengenlehre auf die theorie des Shachspiels. Proceedings of the fifth International Conference of Mathematicians, Cambridge, Cambridge University Press (1912) 501-504.

3. E. Pogossian, "**Specifying personalized expertise**. International Association for Development of the Information Society (IADIS)", International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006), Barcelona, Spain, 2006, pp. 151-159.

Abstract
We aim to understand the role and proportion of personalized expertise compared with common, communicable one. To make the studying constructive we concentrate on a particular class of game represented problems and go in the depth of a typical representative of the class, the chess. We simulate and analyze expert requests to the games storage in a natural language to find winning strategies of the specified types. The correspondence revealed between units of chess vocabulary and winning by Zermelo classes of chess positions and strategies argues for a constructive nature of the content of the units, in principle, allowing to simulate them. At the same time, it states that any real implementation of those contents, in principle, can be only an approximation to the original winning game tree structures due to a prohibitive complexity of computations required to prove correctness of the vast

majority of the contents. Thus, we get a precedent of a measurable specification of human expertise where the learned contents of realities having the same vocabulary, as a rule, are essentially personalized.
Keywords. expertise, content, vocabulary, knowledge, strategies, chess

1. INTRODUCTION

We aim to understand the role and proportion of personalized expertise compared with common, communicable one.

Cognitive sciences say a lot about the mind and, actually, computers allow to simulate many symbolic procedures that are typical for hypothetic–deductive and formal operations period of development by Jean Piaget [Flavell J. 1962].

However, along with progress in modeling the attempts to advance in simulation of early childhood abilities face principal barriers.

Some researchers explain those barriers by symbolic nature of computer simulation which makes principally impossible to represent adequately the inborn and early childhood abilities which are thought to be the bottom part of the iceberg of human abilities [Winograd T., Flores F. 1986]. There is an ongoing long-time discussion about the nature of human image processing and its adequate simulation. The followers of the view on universal “coverage” of the “seeing by mind” [Pylyshyn Z. 2004] argue against those who think it is impossible to reduce the image based depictive thinking to the propositional thinking [Kosslyn S. 1980].

Another view of the mind refers to a religious or some other belief-based interpretations like “Plato’s World” [Penrose R. 1999].

In knowledge representation and formation, we follow the Jean Piaget’s theory on developmental psychology [Flavell J. 1962; Pogossian E.1983] where human knowledge is comprised from schemas of exterior actions (like catching, sucking, etc. ones) and cognitive structures – compositions of interior operations (like arithmetic, logic, quantity and size understanding, etc.). The mechanism of development is organized from the components responsible for adaptation - assimilation, accommodation and equilibration. While the model seems to cover human procedural knowledge, particularly in the form of concepts, it cannot explain, for example, how some patients can reproduce long memorized depictions with arbitrary details during operations on the brain [Springer S., Deutsch G. 1980].

We classify functions of a human mind into three categories based on the separation used in languages to describe human activity. Namely, there are three main types of activities – production, exchange/communication and understanding being, corresponding to verbs “do”, “have” and “be”. We then consider the subject of simulation of functions of human mind as (approximately) composed of three research directions: problem identification and solving, vocabulary understanding and content formation, and understanding being.

Such categorization, along with an effort to identify topics of research of functions of human mind that do not fall into any of the three categories, can help in designing specialized tests to reveal the essence of solutions of simulation problems at comparing human performances to those of programs. Those tests realizing on-the-job performance analysis of concrete problems can overcome the inconsistencies coming from a general destination tests like well-known Turing one [Searle J.1990].

To study the proportion between common and personalized expertise we consider the problem of understanding expert’s requests (UER) by a chess playing computer, given the goals and contexts of those requests. We aim to answer to the following questions:

Is it possible to adequately simulate UER?

Are models of realities corresponding to the same requests common to all participants of the communication acts, or they are personalized?

To what extent, and by what means, one should try to integrate common expert knowledge and personal experience and do they correlate with behavioral views and hypothesis on it?

For reliability of results along with chess we study two other problems of the same class: a protection of computer networks from intrusions and management in oligopoly competitions [Pogossian ,2000-01-03-05].

To study the UER problem we use a collection of about 300 units of chess vocabulary (UCV) with their interpretations [Pogossian et al,1974-80, 1983]. UCV consist of specific words, phrases, idioms, names, etc. used to communicate about chess. We will refer to the collection of UCV with their interpretations as chess repository, and its elements will be called units of chess repository (UCR).

A UER solution must take UCV expert requests as input and must perform the following functions:

Interpret an UCV request and represent it as a formal query to the base of chess positions

Search in the base of chess positions using formal query

Reveal a discrepancy between the respond set of positions and the set needed by the expert

Correct the query if necessary, to repeat the 2d function or to end the search if its quality is satisfactory.

The quality of UER solutions is tested by discrepancy between sets of positions that solutions and experts correspond to the queries tuned to the essence of the problem.

The UER chess problem is analogous to the problem of understanding instructions by robots [Winograd et.al,1986; Roy D.2005]. It seems promising to link models of chess concepts in the UER chess solution with the chunks [GobetF., Simon H.,1998; Gobet et al,2001] to answer whether models of concepts correspond to the chunks and are common for players or are individualized. A comprehensive overview of the state of the art of learning in chess is presented in [Furnkraz J.,2001]]. Zermelo in 1912 [Zermelo] proved the theorem that all chess positions are strongly divided into three classes: winning, losing or drawing. If the descriptions of the above classes were available with an acceptable complexity it would be possible to create a perfect chess player. At present, the theorem's proof technique is the base of the most successfully used strategy search min max algorithms. Using the same technique, similar statements can be proved for other SSGT problems

In the paper we present our study of the UER chess problem where we focus on construction of adequate models of the content of realities corresponding to the expert requests assuming that goals of the requests are in finding chess positions associated with winning strategies and the meaning of the requests stay the same for each analyzed position.

First, we discuss the difficulties of our early attempt to build chess ontology that revealed the fundamental problem of existence of adequate computer models of chess concepts.

Based on analysis of 300 units of chess vocabulary we link chess concepts with classes of winning positions in the game tree that can be specified by Zermelo algorithm and are, in principle, available for both parties of the discourse – the experts and the computers. Then, we propose a way to solve the individualized version of the UER chess problem.

Finally, we formulate some associated research directions and link them with human behavioral ones.

2. LESSONS FROM CHESS ONTOLOGY EXPERIENCE

In our early attempt to solve the UER chess problem by constructing a comprehensive chess ontology the Repository of chess knowledge was formed covering available sources of chess knowledge (Pogossian,83; Pogossian et al, 74-80).

About 300 units of the Repository – concepts, descriptions, rules and other chess related texts were classified in accordance with known tradition in chess for systematic analysis and simulation.

For theoretical analysis of chess concepts a finite logic of predicate calculus with type variables was defined allowing to specify arbitrary composition of elements of the game tree.

Starting from basic concepts of chess specification the ontology was built as a hierarchy of chess concepts where each consequent layer was including models of concepts composed from subordinate ones and some new attributes and procedures.

Up to 4 layers including about 100 concepts were realized. For example, the 4th layer was including, particularly the concepts "Threats to take a piece", "Defense of X by piece Y", "The square X is overloaded", "Double attack ".

Successive formation of layers consequent to the 4th was stopped, particularly, in view of the following difficulties in simulation:

1. While the models of concepts for chess specification were determined easily and there was no doubt about their adequacy the concepts of the third layer and up were requiring tangible efforts to clarify their meaning. In fact, we were forced to determine their meaning in some artificial way that was far away from the one chess students are practicing.
2. Despite to the efforts to reproduce typical meaning common for the chess players there stays a lot of doubt that we can achieve that goal. In chess concepts the role of constituents that are individual for players seemed very essential.
3. It was not excluded that along with factors of reality, context and aim influencing on the meaning of the concepts there were other important ones out of our consideration.

To cope with the first obstacle, we suggest simulation environment where along with means for describing definite common constituents of the concepts there are mechanisms allowing to acquire and represent the precedents and cases of the concepts consistently with the first type of means.

The second obstacle forced us accept the essential difference in representations of the concepts by experts where each representation could be only an approximation of the target chess realities.

To clarify the third issue let's recall that in communication acts, particularly, when chess masters make a request to assistants to find positions in the base of them, the words (phrases, or other communicative means) are triggering the process of understanding the request that includes identification of the goals and context of the request, recall of the attributes that listeners associate with the request which can be of a common use or gained from personal experience, believes, commitments, tradition both parties have, etc.(Winograd , Flores, 87).

While the game tree and all its components are available to computers the second kind of realities about human relationships are far from any acceptable simulation, yet.

In the next section we analyze the content of some units of chess vocabulary and argue that they correspond to the physical entities of the game tree.

Thus, both the experts and computers completely possess the physical reality they intend to communicate about.

3. CONSTITUENTS OF WINNING STRATEGIES

To represent chess knowledge, we classify UCR in frame of known models and constructive views followed by suggestion of hypothesis on their origin.

The 1st step of UCR classification reveals elements of specification of chess as a game which in turn induces new game analysis-based models as the following:

1. game tree model adequately representing chess, all possible behaviors in it – strategies, concept of the best strategy, etc.
2. strategy search methods and methods to enhance their efficiency
3. Zermelo's statement on chess positions partition as winning, losing and drawing ones

On the 2d step of UCR classification those models and their constituents were classified as conceptual and procedural types of strategy knowledge, represented, particularly, by attributes, classifiers and rules and with meanings referred to the elements of game tree, chess specification components or other not chess elements.

Attributes are defined as a kind of operators having values in a given set of realities. Particularly, they may be sets of numbers of some ranges.

We distinct *basic attributes* and constructed ones that compose basic attributes to determine their values. Attributes with 0/1 or "false" /" true" values are named *classifiers* and correspond to concepts and goals.

Concepts are defined as a kind of knowledge to identify and recognize realities whereas goals are concepts with elements of intensions or requirements to achieve them.

Motives are attributes used to argue the preferences of some goals in analyzed situations.

Strategies, plans, goals and *rules* are defined as expert knowledge to specify compositions of his actions in time.

We name (c*omplete*) *S-strategy of C* the performance of the player/competitor C (i.e. the performance of corresponding program) in the game tree with starting position S (S-tree).

Rules are kind of if *x* / then *y* operators to specify a procedure *y* for the realities that fit to the *x* requirements. Any strategy may be determined as a composition of rules.

Given criteria F we say that *a strategy G achieves the goal F* if criteria F are satisfied for the set of terminal nodes of G.

The *strategy G will be called F-projected* if we are interested in whether the terminal nodes of G satisfy criteria F or not.

Any description of an F-projected strategy G aimed to make the search of G more efficient is named a *strategy plan for F* (Pogossian E,Pogosian L., 2000).

Descriptions of problems and strategies resolving them are extreme examples of strategy plans. A useful strategy plan would systematically identify the directions that are not promising and eliminate them for reducing the search space.

Strategy planning is a process of narrowing the search space for the target strategy which also reflects the specifics of the planner such as knowledge of language, the system of values and methods of search.

Identifying UCR by the above game tree constructions allows to associate assertions and hypothesis from corresponding theories.

4. CHESS EXPERTIZE IS PERSONALIZED AND QUASI-SPECIFIES ZERMELO'S CLASSES

People communicate to organize their collective actions. They declare units of vocabulary associated with the content of required actions and expect to trigger corresponding reactions from participants of the dialogue. The participants, in their turn, react in accord with the contents they individually associate with those units [Winograd, Roy].

It has a sense to imagine the contents of units of vocabulary as some procedural structures, for example classes in OOP, that determine actions associated with those units. Depending on situations and the goals persuaded their people interpret, understand the same units in different ways recalling different projections of those procedures.

The entire contents of the UV may correspond to the knowledge that we perceive as an encyclopedic one and which is partly stored in specialized explanatory dictionaries. Fragments of that content we recall and use depend, particularly, on the context of concrete situations where the activity is expected and the goals that are persuaded there [Roy].

The content of units of vocabulary comprises common and personalized for community members elements. While the common parts of the units may be described and passed to members in regular ways by common intermediate means (books, instructors, etc.) the personalized constituents of the contents are formed specifically by each member and there are no regular ways to operate with them.

We identify the following two sources of strategy expertise represented by UCR:

- concepts and attributes determined by specification of the game tree
- winning strategies induced by examining the game tree.

Let's illustrate our reasoning to link chess concepts with classes winning by Zermelo (for the defeat and draw ones similar links can be induced by analogy).

In general, a position is winning if a winning strategy can be generated with the root in that position. The "Mat" positions are the only exclusion where winningness is determined by a few game specification attributes calculated either statically or by one-two plies. The attributes are the following: K(ing) is under check, K can't escape, K haven't defenses.
Being constituents of the concept "Mat", those attributes get their own values of winningness induced by the value of "Mat" winning positions – the maximum value in chess.
It is evident, that any utility of winningness of positions provides not only two extreme "yes" or "not" utility values of those attributes but also values intermediate between those two poles.
Similar reasoning is natural to spread, analogically, to other than King pieces as well as to squares where pieces may be located.
Expanding the ranges of values and definitions of the attributes in a natural way (see the Table) we get the scales where not only their max and min values but intermediate ones, too, become meaningful respectively to utilities they are associated with.

Table 1. Attributes providing utilities to elements of positions

For the King	For a piece X	For a square S
K is under check (threats, capture)	T(x): set of pieces threatening to capture X	Ts(x): set of pieces threatening to S
	A(x): set of pieces attacking X	As (x): set of pieces attacking S
	Cby(x): set of pieces threaten to capture by X	
K haven't defenses (defenseless)	D(x): set of pieces defending X	Ds(x): set of pieces defending S
	Dby(x): set of elements defended by X	
K can't escape (suppression)	M(x): set of moves available to X	

Another source for valued attributes and concepts are positions with known values, winning or not: endgames, combinations, etudes, beginnings, "tabbies" – typical analyzed positions, etc. [Pogossian 83] and, possibly," chunks" [Gabit]. Partitioning and generalizing their constituents we get a set of basic concepts, attributes with associated estimates of their winningness.
These attributes applied to constituents (or configurations of constituents) of analyzed positions create estimates of their utility in the context of winningness of those positions.
The players integrating in some reasonable way the utilities of constituents of positions get estimates of winningness of the positions. A realistic model of that integration is described in [Botvinnik].
Analysis of chess concepts of the Repository allows identify them as the descriptions of elements of positions having tangible winning utility. Hence, chess concepts become elements of specifications of winning by Zermelo positions what argues for possibility of their simulation.
What configurations of elements of positions are covered by concepts and how that is happening need special studying, particularly, in view of the theory of "chunks" [Simon, Gobet 1986].
The concepts can only indicate some possible utilities for further deeper analysis and estimate. An uncertainty is a priory in their nature caused partly by limitations of chess players' static or quasi dynamic analysis of the winningness of positions and partly as a consequence of an individualized way in formation of concepts.
It is worth to accept that using the same names for the concepts chess players along with common strong meaning concepts use many other ones having, in general, not equal interpretations which

coincide only in some “skeleton” parts. Those uncertain concepts, as a rule, don’t create big casualties due of their intermediate preliminary role of hypothesis of winningness of positions which later on are strongly tested by game trees for final decisions.
As a consequence of that generic uncertainty in representation of concepts it is worth to formulate the UER chess problem in a way allowing both forming concepts with strong common meaning and ones that exist in the forms individualized to the particular players. Correspondingly, adequacy of the models of concepts have to be examined taking into account those individualized representations of concepts.

5. FORMING PERSONALIZED CONTENT

Concepts and attributes that specify the games have original utilities and are used for creating new learnable expertise with corresponding induced utilities. The operations to do it can be common (like geometric transformations preserving some utilities) or be specific for individuals and their experience.
New learnable knowledge can specify winning positions and strategies only with some degree of likelihood. That is because the utilities of expert strategy knowledge are mainly tested by static and quasi dynamic means and are not substantiated by game tree complete dynamic analysis due of their enormous size.
The utilities of expert knowledge stay, as a rule, uncertain what is an essential argument against their totally unified models. In fact, they are mostly individualized with a part of common constituents.
The content of a UV is a procedure recognizing definite set of realities unified by the name of that UV. That procedure is the solution of a classification problem which is either an acquisition of existing ones or is learned by solving classification problems, specified, as a rule, inductively by providing examples of situations with associated utilities.
Thus, the starting point to learn the content is the utility u(P*) we gain in a situation P*.
We associated attributes, concepts other procedures with that P* that for new situations cause the same utility u(P*) if they take place for them.
The procedures associated with utility u(P*) determine corresponding goals which in their turn induce new goals that causally are connected with the old ones, etc. As the result the causality chains are formed where the subordinated child ones become favorite goals to approach to the parent ones.
How determine the utility u(P) of a P situation given the u(P*) for situation P* with attributes X*= x1,…, xn true for P* . ?
Typical rules to form the causality transitive chains include, particularly, the following ones:
Any P has utility u(P) so close to the u(P*) as many attributes from X* are true for P
As many situations of some class have the same true attributes so much is the likelihood that a new situation with the same true attributes will belong to the same class
There are several methods to form the content of base elements of a vocabulary given *training sets in attribute representations.* They include methods to form partial Boolean functions, machine learning and statistics-based methods, the Beckon-Mile induction, the G. Polya’s plausible reasoning, Piaget’s assimilation-accommodation, etc. methods.
If training examples are unique it is appropriate to use transformations preserving the utility of those examples which can be, particularly, problem dependent, common geometrical or logical, etc.

6. PROVING ADEQUACY OF MODELS

Proving adequacy of UER chess solutions, in general, can be identified as a computationally hard problem of identification or proving equality of programs. Thus, there is no straightforward complete testing for UERchess solutions.
In general, the dialogue has to be organized between carriers of the concepts where the experts must test the adequacy of models. The testing has to be on a regular, independent from a particular subject

base. There have to be criteria and technique to estimate the level of approximation, make necessary measurements, interpret and substantiate the experiments requiring, particularly, not an abstract, absolute approximation of some ideal concepts but only coincidence with the expert ones

We test the models of expertise as the following:

- examples are generated on the board by the models and commented by the experts;

for that a generator using models of the concepts as inputs generates situations on the board that don't contradict to the models, submits them to the experts and gets their estimates for further possible corrections;

- because there is a threat that only a part of the meaning of the concepts were "caught " by the models they have to be tested by complementary means like the following:

* experts generate examples which are tested by corresponding models
* situations are taken from some repository and the reactions of the models and experts are compared
* situations are taken from the professional books or hand-books.
* there is an evidence of a completeness of the set of tests [on-the- job performance ideology]

Summarizing, the following consequences of the UCR study can be formulated:

1. Concept type UCR are, as a rule, hypothetic, approximate descriptions of classes of winning by Zermelo positions, or are constituents of such descriptions.
2. The plan and strategy types of UCR can be interpreted as constructions and elements based on the constituents of the game tree.
3. The likelihood of hypothetic descriptions, as a rule, is small because precise determination whether the positions are winning cannot be done since it needs generation and evaluation corresponding strategies what is a hard-computational problem. Therefore, the value of concepts, in principle, cannot be common for all experts and descriptions of concepts along with common constituents have to include an essential part of subjective ones.
4. The ideology of the Object Oriented (OO) programming along with OO Data Bases, Case Based Systems (CBS), Neuron Nets, ID3 and Boolean functions reconstruction techniques (Dhar, Stein,97; Russel, Norvig, 2002) seems appropriate to integrate common and subjective constituents of concepts.
5. The techniques equally with CBS can be corresponded with human mechanisms of unconscious and intuitive knowledge processing with a huge amount of precedents.

7. CONCLUSION

The correspondence revealed between about 300 units of chess vocabulary and winning by Zermelo classes of chess positions and strategies argues for a constructive nature of models of realities.

It states that real implementation of the models, in principle, can only approximate to specification of winning game tree structures due to irresistible complexity of computations to prove correctness of the models. Thus, it follows that

- for the same requests both chess players and computers will, as a rule, use different models of realities essentially based on their individual experience
- any preferences between those models include uncertainty
- interpretations of units of vocabulary have to rely on some personalized experience.

What means are worth to use for integrating common expert knowledge and personal experience and do they correlate with behavioral views on it?

Different organizational forms of regular knowledge and personal experience, the forms of their integration and usage and correlation with known behavioral views on them are meaningful to discuss. However, the recommendations about their merits we find reasonable to do after experiments in the process. We hope the experiments will highlight, particularly, the following behavioral views and observations of chess, language and intuition described by H. Simon, de Groot, J. Adamar, G. Atkinson, F. Gobet, M. Krogius, P. Smolensky, D. Roy:

1. All lifelong human impressions are stored in his memory (seemingly, in the right hemisphere); that impressions repository is the base for hints, ideas, solutions of problems and other actions.
2. Human knowledge has prepositional and depictive forms of storage and representation with corresponding search engines.

It is stored as a composition of units of different levels of abstraction, e.g. as a combination of prepositional (predicative, procedural) descriptions of utilities invariant to associated impressions with precedent, cases, examples types of it.

3. The requests to the repository of impressions can be conscious or not and represented in a diversity of forms: attributive, associative, aesthetic, emotional, hormonal, etc. The needs for the requests are initiated by intensions and formed using entire expert knowledge.
4. The repository of concepts and other types of knowledge is the base of attributes for prepositional (structural, linguistic) representation of target positions and requests.

Those representations, in turn, become elements of new attributes and higher-level abstractions in form of prepositions.

5. "Grounding" the expert requests (Roy,2005) either with current actions or with impressions memorized in similar situations in the past is an essential step of understanding; usually, the process includes iterative justification of the requests to achieve acceptable satisfaction by results of actions caused by them.

8. REFERENCES

[1] Atkinson G. 1993. *Chess and Machine Intuition.* Ablex Publishing Corporation, Norwood, New Jersey.

[2] Botvinnik M. 1984. *Computers in Chess: Solving Inexact Search Problems.* Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag: NY.

[3] Dhar V, Stein R.1997. *Seven Methods for Transforming Corporate Data into Business Intelligence*, NJ, Prentice-Hall.

[4] Elo A.E. 1978. *The Rating of Chess players, Past and Present.* London, Batsford.

[5] Flavell J. 1962. *The Develometal Psycology of Jean Piaget*, D.VanNostrand Company Inc., Princeton, New Jersey.

[6] Furnkranz J. 2001.*Machine Learning in Games:* A Survey in "Machines that Learn to Play Games", Nova Scientific.

[7] Kruchten P. 2000. *The Rational Unified Process. An Introduction*, Addison-Wesley.

[8] Gobet, F., Simon, H.A. 1998. Expert chess memory: Revisiting the chunking hypothesis*. Memory,* 6, 225-255.

[9] Gobet, F.at al, 2001. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5, 236-243.

[10] Kosslyn S. 1980, *Image and Mind.* Cambridge, MA Harvard University Press.

[11] Penrose R. 1999. *The Emperor's New Mind.* Oxford University Press.

[12] Pogossian E., Javadyan A., Ivanyan E. .2005. Effective Discovery of Intrusion Protection Strategies. *Lecture Notes in Computer Science,* AIS-ADM-05: The International Workshop on Autonomous Intelligent Systems - Agents and Data Mining June 6-8, 2005, St. Petersburg, Russia, LNAI 3505, pp.263-276.

[13] Pogossian E. Javadyan A. 2003. A Game Model for Effective Counteraction Against Computer Attacks in Intrusion Detection Systems, *NATO ASI, Data Fusion for Situation Monitoring*, Incident Detection, Alert and Response Management", Tsahkadzor, Armenia, pp.30.

[14] Pogossian E. 2001. Focusing Management Strategy Provision Simulation. *Proceedings of the CSIT2001, 3d International Conference in Computer Science and Information Technologies*, Yerevan.

[15] Pogossian E., Pogossian L. 2000. On Management Strategy Search Simulation//*Artificial Intelligence, N2,* p.424-431. http://wwwmaus-soft.com/iai; http://www.iai.donetsk.ua/_u/iai/dtp/ai.html .
[16] Pogossian E. 1997. On Explanatory Definition of English Clause. *5th International Cognitive Linguistics Conference,* Vrije Universiteit Amsterdam, July 14 - 19, 1997; Transactions of IPIA NAS RA, pp 46-55.
[17] Pogossian E. 1983. *Adaptation of Combinatorial Algorithms* (a monograph in Russian), Yerevan., 293 pp.
[18] Pogossian E., Hambartsumyan M., Harutunyan Y. (1974-1980) A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations. *National Academy of Sciences of Armenia, IPIA, research reports* (in Russian).
[19] Pylyshyn Z. 2004. *Seeing and Visualizing: It's Not What You Think, An Essay On Vision And Visual Imagination*, http://ruccs.rutgers.edu/faculty/pylyshyn.html.
[20] Roy D. 2005. Grounding Language in the World: Signs, Schemas, and Meaning Cognitive Machines Group, The Media Laboratory, MIT *(http://www.media.mit.edu/cogmac/projects.html).*
[21] Rui Y., Huang T. S., Chang S. F. 1999. Image retrieval: Past, present, and future., *J. of Visual Communication and Image Representation,*10:1-23.
[22] Russell S., Norvig. P. 2002. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Englewood Cliffs, NJ.
[23] Searle J. 1990.Is the brain's mind a computer program? *Scientific American* 262, pp26-31.
[24] Smolensky, Paul. 2001. Optimality Theory. In MIT Encyclopedia of the Cognitive Sciences, eds. Robert A. Wilson and Frank C. Keil. Cambridge, MA: MIT Press/Bradford Books.
[25] Springer S., Deutsch G. 1980. Left Brain, Right Brain. State University of NY.
[26] Winograd T., Flores F. 1986. Understanding Computers and Cognition (A new foundation for design). Publishers, Chapter 2, pp. 11–59, Huntington, NY.
[27] Zermelo E. 1912. Uber eine Anwendung der Mengenlehre auf die theorie des Shachspiels. Proceedings of the fifth International Conference of Mathematicians, Cambridge, , Cambridge University Press, p.501-504.

4. Pogossian E. **On Measurable Models of Promotion of Negentropic Strategies by Cognition**, 2010, pp.37-45 in this Proceedings.

Abstract: Could models of mind be independent from living realities but be classified as mind if the mind uses the same criteria to form the class mind?
In the paper a constructive view on the models of mind, cognizers, is presented and the measurable criteria and schemes of experiments on mentality of cognizers are discussed.
Keywords: modeling, cognition, measures, negentropic, strategies.
ACM Classification Keywords: A.0 General Literature – Conference.
Conference topic: Problems of Computer Intellectualization

1. Introduction

Due mind forms models of any realities including itself raises the question whether models of mind can be mental not being living realties (LR), assembled from LR or developed from the springs of LR?

In other words, whether are models of mind which do not depend from LR but are classified as mind possible if mind uses the same criteria when forms the class mind?

To answer the question constructive models of mind and criteria of measuring their mentality as well as the exhaustive experiments on revealing the truth are needed.

In what follows a measurable approach to the models of mind, cognizers, is presented and the criteria and experiments of testing of mentality of cognizers are questioned.

This approach to refining of cogs continues the approach started in [Pogossian,1983] and continued in [Pogossian,2005,2007] on interpretation of the recognized views on mind [Flavell,1962,Neuman,1966, Botvinnik,1984, Atkinson1993, Pylyshin,2004, Roy,2005, Winograd, 986,Mendler,2004,] by models having unanimous communalized meanings followed by experiments on validity of those models.

The paper describes the author’s view on mental behavior and traditionally we should address to the readers by using words "our view“,"we think», etc.

On the other hand, mental behavior, we assume, is identified with ourselves and we plan to discuss personalized and communalized constituents in communications.
That explains why we find possible in the paper to use the pronoun "I" for the mind along with "we" and "our" when they seem to be appropriate.

2. A View on Mind

2.1.I am a *mind* and I am able to interpret, or *model* the *realities* I *perceive,* including myself, evaluate the quality, or *validity* of models and use those models to promote my *utilities*.

The models are composed from cause-effect relationships between realities, particularly between realities and utilities, and any composition of those relationships comprise the *meanings* of the realities.

The basic, or *nucleus* utilities and meanings are inborn while mind incrementally enriches them by assimilating and accommodating by Piaget [Flavel,1962, Mandler, 2004] cause-effect relationships between realities and already known utilities and meanings *solving* corresponding *tasks* and *problems.*

By Piaget “Mind neither starts with cognition of itself nor with cognition of the meanings of realities but cognizes their interactions and expanding to those two poles of interactions mind organizes itself organizing the world” [Flavell,1962].

As much coincide ontology, or *communalized* (vs. *personalized*) meanings of realities with meanings of their models and as much those meanings are *operational*, i.e. allow to reproduce realities having equal with the model’s meanings, so better is the validity of the models.

In what follows a personalized model of mind, a view W, and a communalized version of W, *cognizers,* are presented with discussion of the validity of cognizers and schemas to meet the requirements.

2.2.1. Minds are algorithms for promoting by certain *effectors* the utilities of *living realities* (LR) in their games against or with other *players* of those games.

The players can be LR, assembles of LR like communities of humans or populations of animals as well as can be some realities that become players because not voluntarily but they affect LR inducing games with environments or the units like programs or devices that have to be tested and response to the actions of engineers. To compare and discuss some hypothetic mental realities like Cosmic Mind by Buddhists and Solaris by Stanislaw Lem are considered as players as well. Note, that descriptions of religious spiritual creatures resemble algorithm ones.

2.2.2. A variety of economic, military, etc. games can be processed by players. But all LR in different ways play the main negentropic games against overall increase of the entropy in the universe [Shrodinger,1956].

In those negentropic games with the environments LR and their populations realize some versified *reproduction* and on-the-job selection *strategy elaboration algorithms* (r SEA).

The parent rSEA periodically generates springs of LR where each child of the springs realizes some particular strategy of survival of those children in on going environments. LR with successful survival strategies get the chance to give a new spring and continue the survival games realizing some versions of strategies of their parents while unsuccessful LR die.

2.2.3. The utilities of LR and their assembles initially are determined by their nucleus, basic interests in the games but can be expanded by new mental constructions promoting already known utilities. For example, the nucleus utilities of LR, in general, include the codes (genetic) of rSEA and algorithms for reconstructing rSEA using their genetic codes.

2.2.4. The periods of reproduction, the power of the springs and other characteristics of rSEA are kinds of means to enhance survival abilities of LR and vary for different LR depending, particularly, from the resources of energy available to LR and the velocity of changes of the environments of LR.

1.2.5. Minds can be interpreted as one of means to enhance the survival of LR. In fact, minds realize SEA but in contrast to on-the-job performance rSEA the strategies elaborated by minds are auxiliary relatively to rSEA and are selected by a priory modeling.

Correspondingly, the nucleus of mental LR in addition to rSEA codes include codes of mind developing algorithms like the adaptation algorithms by Piaget [Flavel,1962, Mandler, 2004].

2.3. Thus*, modeling* SEA, or mSEA, do, particularly, the following:

- form the models of games and their constituents
- classify models to form classes and other mental constructions
- use mental constructions for a priori selection the most prospective strategies for the players
- elaborate instructions for the effectors of players using the prospective strategies.

The effectors transform the instructions into external and internal *actions* and apply to the environments of mSEA and mSEA themselves, correspondingly, for developing the environments and mSEA and enhancing the success of the players.

2.4. Whether are the models of mind which are not dependent from LR but are classified as mind possible if mind uses the same criteria when forms the class *mind*?

To answer to the question constructive models of mind and criteria of measuring their mentality as well as the exhaustive experiments on revealing the truth are needed.

2.5. Let's name *cognizers* the models of mind not depending from LR while the models of mental constructions name *mentals*.

Apparently, this ongoing view W on mind is a kind of cognizers, say, for certainty, *1-cognizers, 1cogs* or *cogs* in this paper.

In what follows a constructive approach to cogs, the criteria and experiments of testing of mentality of cogs are presented.

3. Basic Approaches and Assumptions

3.1. Further refining of cogs extends the approach described above on interpretation of the recognized views on mind by models having unanimous communalized meanings followed by experiments on validity of those models to mind.

3.2. 1. Later on it is assumed that cogs are object-oriented programs, say in Java.

All programs in Java are either classes or sets of classes.

Therefore, it is worth to accept that cogs and their constituents, mentals, are either Java classes or their compositions as well.

Accepting the above stated assumption, the experiments on quality of cogs were run for SSRGT games. Particularly, because chess represents the class and by variety of reasons is recognized as a regular environment to estimate models of mind [Botvinnik,1984, Pogossian,1983,2007, Atkinson,1993, Furnkranz, 2001] in what follows the constructions of mentals and experiments on mentality of cogs are accompanied, as a rule, by interpretations in chess.

3.4. Following to the view W cogs elaborate instructions for the effectors of players to promote their utilities. The effectors in turn transform instructions into *actions* applied to the players and their *environments.* They can be parts of the players or be constructed by cogs in their work.
It is assumed that certain *nucleus* mentals of cogs as well as the players and their effectors are predetermined and process in discrete time intervals while mentals of cogs can evolve in time.
The fundamental question on the origin of nucleus mentals and other structures needs further profound examination.

4. Refining Constituents of Cognizers

4.1.1. In general, *percepts* are the inputs of cogs and have the structure of bundles of instances of the classes of cogs composed in discrete time intervals.
The *realities* of cogs are refined as the causes of their *percepts*.
The *environments* and the *universe* of cogs are the sets and the totality of all *realities of cogs*, correspondingly.
More in details, the bundles of instances of attributes of a class X of cogs at time t are named *X percepts at t* and the causes of X/t percepts are named *X/t realities.*
It is worth to consider *t percepts* and *percepts* as the elements of the unions of X/t percepts and t percepts, correspondingly, and assume that there may be multiple causes for the same percept.
Analogically, *t realities* and *X/t realities* are defined.
In case percepts are bundles of instances of attributes of certain classes of cogs the realities causing them are the classes represented by those attributes.
Otherwise, cogs learn about the realities by means of the percepts corresponded to realities and by means of the responses of those percepts when cogs arrange actions by effectors. Due cogs are continuously developed they start with percepts formed by nucleus classes followed by percepts formed by the union of new constructed and nucleus classes.
4.1.2. Cogs promote utilities by using links between utilities and percepts. They continuously memorize percepts, by certain criteria unite them in classes as *concepts* and distinguish realities to operate with them using *matching* methods associated with the concepts.
In addition, some concepts are nominated by *communicators* to communicate about the realities of the domains of the concepts with other cogs or minds and enhance the effectiveness of operations of cogs in the environments.
4.2.1. The base criteria to unite percepts in concepts are *cause-effect relationships* (cers) between percepts, particularly, between percepts and utilities.
For revealing cers cogs *form* and *solve tasks* and *problems.*
Tasks are requirements to link given percepts (or realities) by certain cers and represent those cers in frame of certain classes.
4.2.2 The basic tasks are the *utility* tasks requiring for given percepts to find utilities that by some cers can be achieved from the percepts. In chess utility tasks require to search strategies for enhancing the chances to win from given positions.
The *generalization*, or *classification* tasks unite percepts (as well as some classes) with similar values into more advanced by some criteria classes and associate corresponding matching procedures with those classes to distinguish the percepts of the classes and causing them realities.
The *acquisition* tasks create new classes of cogs by transferring ready to use classes from other cogs or minds while the *inference* tasks infer by some general rules new classes as consequences of already known to cogs classes.
The *question* tasks can be considered as a kind of formation tasks inference tasks which induce new tasks applying syntax rules of question tags to the solutions of already solved tasks.

The *modeling* tasks require revealing or constructing realities having certain similarities in *meanings* with the given ones.
Before refining meanings of realities let's note that to help to solve the original tasks some approximating them model tasks can be corresponded.
4.2.3. *Problems* are compositions of homogeneous tasks and *solutions of problems* are procedures composing the solutions of constituent tasks.
The problems can be with *given spaces* of possible *solutions* (GSS) or without GSS, or the *discovery* ones.
Tasks formation and *tasks solving procedures* form and solve tasks types.
4.3.1.To refine the meanings of realities and mentals it is convenient to interpret the percepts, uniting them concepts, nucleus classes and the constituents of those mentals as the nodes of the *graph of mentals* (GM) while the edges of GM are determined by utility, cers, attributive, part of and other relationships between those nodes.
Then the *meaning of a percept C* can be defined as the union of the totality of realities causing C and the connectivity sub graph of GM with root in C.
The *meaning of a concept X* is defined as the union of the meanings of the nodes of the connectivity sub graph of GM with the root in X.
The *meaning of realities R* causing the percept C is the union of the meanings of the nodes of the connectivity sub graph of GM with the root in the percept C.
4.3.2. Later on, it is assumed that the *knowledge* of cogs unites, particularly, the cogs, GM and their constituents.
4.4.1. Processing of percepts and concepts is going either *consciously* or *unconsciously.* While unconsciousness, usually, addresses to the *intuition* and needs the long way of research efforts for its explanation, the consciousness is associated with the named concepts and percepts in languages and their usage for communications. Particularly, the vocabularies of languages provide names of variety of concepts and realities causing those percepts.
Mind operates with percepts, concepts and other mentals while names realities causing those mentals when it should communicate.
Particularly, this ongoing description of cogs follows to the rules for named realities while internally refers to corresponding mentals.
4.4.2. When mind operates *internally* with the representations of realities it is always able to address to their meanings or *to ground* those representations [8].
For *external* communications mind uses representations of realities, *communicators,* which can be separated from the original carriers of the meanings of those realities, i.e. from the percepts of those realities, and become *ungrounded*.
The role of communicators is to trigger [12] the desired meanings in the partners of communications. Therefore, if partners are deprived of appropriate grounding of the communicators special arrangements are needed like the ones provided by ontologies. If the communicators are not sufficiently grounded well known difficulties like the ones in human-computer communications can rise.
Note, that if the model R` is a grounded reality the meaning of R' can induce new unknown aspects of the meaning of the original ones.
4.5. Realities R` *represent* realities R, or R` is a *model* of R, if meanings of R` and R intersect.
Model R` is *equal* to R if R' and R have the same meanings. The more is the intersection of the meanings of R and R` relative to the meaning of R the greater is the *validity* of R`. For measuring the validity of models, a variety of aspects of the meanings of original realities can be emphasized. Particularly, descriptive or behavioral aspects of the meanings can be considered, or be questioned whether the meanings are views only of the common use or they are specifications.

5. Questioning Validity of Mind

5.1. Modeling problems require constructing realities having certain similarities in meanings with the original ones.
When those realities are problems as well cogs correspond model problems to the original ones, run them to find model solutions and interpret them back to solve the original ones.
Apparently, solutions of problems are the most valid models of those problems but, unfortunately, not always can be found in frame of available search resources.
Valid models tradeoff between the approximations of the meanings of solutions of problems and between available resources to choose the best available approximations.
Due of that inevitable trade off the models are forced to focus on only the particular aspects of those solutions.
If communication aspects are emphasized the *descriptive* models and criteria of validity can be in use require the realities-models be equal only by communicative means of the communities.
On-the-job or *behavioral* criteria evaluate validity of models by comparing the performances of corresponding procedures.
The records of computer programs provide examples of descriptive models while when processed programs become the subject of behavioral validity. Sorts of behavioral validity provide functional testing and question-answer ones like Turing test.
Productive behavioral validity criteria compare the results of affection of the outputs of realities and their models on the environment. Fun Newman requirement on self-reproducibility of automata [Neuman, 1966] provides an example of productive validity. In its interpretation as *reflexive reproducibility* (RR) validity that criterion requires to construct 1-models of realities able to produce 2-models equal to the 1-models and able to chain the process.
5.2. To formulate criteria of validity of cogs it is worth to summarize the refined to this end views on mind as the following:

- mind is an algorithm to solve problems on promotion of utilities of LR in their negentropic games
- mind is composed from certain constituent algorithms for forming and solving tasks of certain classes including the utility, classification, modeling, questioning classes
- mind uses solutions of problems to elaborate instructions for certain effectors to make the strategies of LR more effective and the environments of LR more favorable to enhance the success of LR in negentropic games.

5.3. Criteria of validity of cogs to mind have to answer whether cogs have meanings that minds have about themselves.
On the long way in approaching to valid cogs a chain of inductive inferences is expected aimed to converge eventually to target validity.
Inductive inferences unite science with arts and, unfortunately, the term of their stabilization cannot be determined algorithmically. Nevertheless, what can be done is to arrange those inferences with the trend to converge to the target stabilization in limit.
To approach to valid cogs, it is worth to order the requirements to the validity of cogs and try to achieve them incrementally, step by step.
The requirements v1- v4 to validity of cogs condition them to meet the following:
v1. be well positioned relatively to known psychological models of mind
v2. be able to form and solve the utility, classification, modeling and question tasks with acceptable quality of the solutions
v.3. be able to use the solutions of tasks and enhance the success of the players
v4. be able to form acceptable models of themselves, or be able to *self-modeling*.

The requirements v2 - v4 follow the basic views on mind while v1 requires positioning cogs relatively, at least, to the recognized psychological models of mind to compare and discuss their strengths and weaknesses.
Note, that parent minds of LR reproduce themselves in the children minds in indirect ways using certain forms of cloning, heritage and learning procedures.
Some constituents of reproduction of LR can already be processed artificially, i.e. by regular for the human community procedures.
The requirement v4 is questioning, in fact, whether completely artificial minds, cogs, can reproduce new cogs equal themselves and to the biological ones.
5.4. What are the validity criteria to make cogs equal by meaning to mind and whether cogs valid by those criteria can be constructed?
It is a long way journey to answer to these questions and elaborate some approaches to implement.

6. Conclusion

Valid cogs, if constructed, confirm the assertion that mind is a modeling-based problem formation and solving procedure able to use knowledge gained from the solutions to promote the utilities of LR in their negentropic games.
Synchronously, mental cogs provide a constructive model of mind as the ultimate instrument for cognition.
Knowledge on the nature of instruments for revealing new knowledge gives a new look on the knowledge already gained or expected and raise new consequent questions.
Therefore, revealing by cogs the new knowledge on the instruments of cognition it is worth to question the new aspects of relationships between mind and the overall knowledge mind creates and uses.
Ongoing experiments on study of cogs are based on the technique of evaluating adaptive programs and their parts by local tournaments and use the game solving package with its kernel Personalized Planning and Integrated Testing (PPIT) and Strategy Evaluation units [Pogossian,1983,2005,2007].
Extended Abstract:
In what follows a measurable approach to the models of mind, cognizers, is presented and the criteria and experiments of testing of mentality of cognizers are questioned.
This approach to refining of cogs continues the approach started in [Pogossian,1983] and continued in [Pogossian,2005,2007] on interpretation of the recognized views on mind [Flavell,1962,Neuman,1966, Botvinnik,1984, Atkinson1993, Pylyshin,2004, Roy,2005, Winograd, 986,Mendler,2004,] by models having unanimous communalized meanings followed by experiments on validity of those models.
Due mind forms models of any realities including itself raises the question whether models of mind can be mental not being living realties (LR), assembled from LR or developed from the springs of LR?
In other words, whether are models of mind which do not depend from LR but are classified as mind possible if mind uses the same criteria when forms the class mind?
To answer the question constructive models of mind and criteria of measuring their mentality as well as the exhaustive experiments on revealing the truth are needed.
Mind is able to interpret, or model the realities it perceives, including myself, evaluate the quality, or validity of models and use those models to promote my utilities.
The models are composed from cause-effect relationships between realities, particularly between realities and utilities, and any composition of those relationships comprise the meanings of the realities.
The basic, or nucleus utilities and meanings are inborn while mind incrementally enriches them by assimilating and accommodating by Jean Piaget cause-effect relationships between realities and already known utilities and meanings solving corresponding tasks and problems.

By Piaget “Mind neither starts with cognition of itself nor with cognition of the meanings of realities but cognizes their interactions and expanding to those two poles of interactions mind organizes itself organizing the world”. As much coincide ontology, or communalized (vs. personalized) meanings of realities with meanings of their models and as much those meanings are operational, i.e. allow to reproduce realities having equal with the models' meanings, so better is the validity of the models.
In the paper s a personalized model of mind, a view W, and a communalized version of W, cognizers, are presented with discussion of the validity of cognizers and schemas to meet the requirements.

References

[1] [Atkinson,1993] G. Atkinson Chess and Machine Intuition. Ablex Publishing Corporation, Norwood, New Jersey, 1993.

[2] [Botvinnik,1984] M. Botvinnik Computers in Chess: Solving Inexact Search Problems. Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag: NY, 1984.

[3] [Flavell,1962] J. Flavell The Develometal Psycology of Jean Piaget, D. Van Nostrand Company Inc., Princeton, New Jersey, 1962.

[4] [Furnkranz, 2001] J. Furnkranz Machine Learning in Games: A Survey in “Machines that Learn to Play Games”, Nova Scientific, 2001.

[5] [Mandler,2004] Mandler J. The Foundations of Mind: Origins of Conceptual Thought. Oxford Univ. Press, 2004.

[6] [Neuman, 1966] John von Neuman.Theory of Self-reproducing Automata. University of Illinois Press, 1966.

[7] [Pogossian,2007] E. Pogossian. On Measures of Performance of Functions of Human Mind. 6th International Conference in Computer Science and Information Technologies, CSIT 2007, Yerevan, 2007, 149-154.

[8] [Pogossian ,2006] E. Pogossian. Specifying Personalized Expertise. International Association for Development of the Information Society (IADIS): International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006), 8-10 Dec., Barcelona, Spain (2006) 151-159.

[9] [Pogossian,2005] E. Pogossian. Combinatorial Game Models for Security Systems. NATO ARW on "Security and Embedded Systems", Porto Rio, Patras, Greece, Aug. (2005) 8-18.

[10] [Pogossian,2007] E. Pogossian, V. Vahradyan, A. Grigoryan. On Competing Agents Consistent with Expert Knowledge", Lecture Notes in Computer Science, AIS-ADM-07: The Intern. Workshop on Autonomous Intelligent Systems - Agents and Data Mining, June 5 -7, 2007, St. Petersburg, 11pp.

[11] [Pogossian,1983] E. Pogossian. Adaptation of Combinatorial Algorithms (a monograph in Russian), 293 pp. 1983. Yerevan.

[12] [Pylyshyn,2004] Z. Pylyshyn Seeing and Visualizing: It's Not What You Think, An Essay On Vision And Visual Imagination, http://ruccs.rutgers.edu/faculty/pylyshyn.html, 2004.

[13] [Roy,2005] D. Roy Grounding Language in the World: Signs, Schemas, and Meaning Cognitive Machines Group, The Media Laboratory, MIT (http://www.media.mit.edu/cogmac/projects.html) 2005.

[14] [Searle,1990] Searle J. Is the brain's mind a computer program? Scientific American 262, pp26-31, 1990.

[15] [Shannon, 1949] C.E.Shannon. The Mathematical theory of communication. In: The Mathematical Theory of Communication. Ed. C.E. Shannon and W. Weaver. University of Illinois Press, Urbana, 1949.

[16] [Shrodinger,1956] E. Shrodinger. Mind and Matter. Cambridge, 1956.

[17] [Winograd,1986] T. Winograd, F. Flores. Understanding Computers and Cognition (A new foundation for design). Publishers, Chapter 2, pp. 11–59, Huntington, NY, 1986.

5. Grigoryan S., Hakobyan N. and Baghdasaryan T., **Knowledge-Based Solvers for RGT Combinatorial Problems**, 13th International Conference in Computer Science and Information Technologies, CSIT2019, Yerevan, 2019, pp36-46, as well reprinted in IEEE's Xplore https://ieeexplore.ieee.org/xpl/conhome/8890749/proceeding.

ABSTRACT

In this work we present means for complete at the time usage of RGT Solvers, that include knowledge acquisition, matching and decision-making algorithms for RGT (Reproducible Game Trees) problems, particularly from the given specification of the problem it acquires knowledge and provides for adequate to expert decisions.

The RGT Solvers are regularly being improved and the current implementation tries to cover the drawbacks of previous versions, improved presentation of classifiers and mental doings, as well as enhancements in interface.

We also discuss ways of knowledge acquisition for marketing, battlefield problems by RGT Solvers.

KEYWORDS. Competition, strategy, planning, chess, expert systems.

1. INTRODUCTION

1.1. Description of RGT Solvers, achievements.

We develop tools for solving RGT [1-3] (Reproducible Game Trees) problems in adequate to expert ways. RGT is a class of problems, which includes the following requirements:

- there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions in the (c) specified types of situations;
- there are identified utilities, goals for each actor;
- actions for each actor are defined.

1.1.1. Many urgent problems of combinatorial nature, including marketing and management, network protection from various types of intrusions by hackers [4], chess and chess like problems [1, 5], certain problems of decision making in battle fields [6, 7], marketing (competing in oligopoly marketing environment) and management (supply chain management case, as an example) problems [18, 19], etc. RGT problems are reducible to each other, particularly, to some standard kernel RGT problem K, e.g. chess [7, 8]. Thus, we also follow the common approach in experimenting knowledge-based systems for chess [9].

1.1.2. RGT Solvers aim to solve such problems from specification and expert knowledge bases to search for optimal strategies.

Generally taking RGT Solvers consist of the following main modules: 1) Interface to interact with human, insert knowledge, provide situations and get the output for executed algorithms, 2) Knowledge base, which acquires the knowledge, adequate constructions and modules for effective usage of them, particularly in matching to situations, 3) Decision making modules, which provide adequate to human decision making algorithms in such problems.

1.2. Background of knowledge base.

1.2.1 Human deals with realities, some of which are not classified while human mainly deals with classified ones.

Classified realities can be divided into regularized and not regularized.

1.2.2. A mighty way of enhancement of effectiveness of mental systems is the regularization of classifiers induced by mdoers and mental systems.

Namely, classifiers 'Cl' of members 'x' of communities 'C' are regularized in 'C' if accompanied by ontological in 'C' methods, instructions allowing 'x' regularly provide positive samples of inputs of 'Cl' as well as let the members of 'C' do the same by communicating with 'x'. [10, 11].

1.2.3. In the mentioned approach the following it was stated that:

i. Algorithms are modeling and constructively regularize deterministic methods, ii. OO Languages are constructively regularized and strongly expand algorithms, iii. Mentals are constructively regularized and strongly expand OOL, iv. Mentals can consist of functional and connectivity mental models.

Natural languages contain a large number of constructively classified mentals, e.g., English has about 300 000 classifiers.

Algorithms are type of systems constructively modeling computational mental doings over numeric input IDs of realities and OO languages expand them by adding attributing/have, parenting/be and do relations.

1.3. For the strategy construction planning algorithms adequate to human planning are used. Plans are certain general descriptions of strategies. Each plan represents a hierarchy of goals. Those are the goals which are attempted to be achieved in given situations while playing by the plan. The essences of the plans are to select the goals which get the maximal profit from the current plan aims.

Realizing the current plan, the shell can determine the goal in the agenda which in turn determines basic attributes to be considered followed by indication of the arguments of those attributes.

1.4. Based on the provided background we demonstrate the adequate abilities of knowledge acquisition, matching and decision-making algorithms, particularly strategy searching based on provided expert knowledge.

We also provide evidence of adequacy of developed models to expert knowledge specifications for problems other than experimental chess problem, including marketing and battle field problems.

Particularly in contrast with previous versions of Solvers we aim to provide fully systemic classification with integrated TZT in decision making algorithms.

2. KNOWLEDGE PRESENTATION IN RGT SOLVERS

RGT problem acquisition from experts and decision providing based on for a certain problem specification requires several steps. In the following section we provide detailed description of how problems, strategy related knowledge and situations are integrated in RGT Solver and how situations are matched.

2.1 Knowledge concept

Knowledge are represented in form of logical expressions (predicates), in terms of the same scripting language that is used for strategy plans description.

Generally, those expressions look like a set of qualitative actions (with a glance of situational dependencies and/or conditions) that result to the expected qualitative effects (changes).

The action set is defined by the particular strategy plan and is contained in its description.

Separate qualitative actions (or moves) might require a substitution of quantitive values for defining the intensity of their affection, i.e. quantification.

The result is represented as an effect of that strategy plan application, in form of $\{P_1, P_2 \ldots, P_n\}$ environmental parameter values' qualitative changes [18].

2.1.1 Elements of knowledge base

A single element of knowledge base within a single discrete period of time, i.e. that contain information for a single simulation cycle, occupies a single record in the knowledge base and represented as an environment condition (***situation***) before the application of a particular strategy plan (SP_i) or ***action***, and the resulting ***situation*** after that.

As an environment state (situation) it is considered the qualitative values of all the working parameters of the problem model at the particular moment (simulation cycle): In general, situation is represented as a set of environmental parameter values (EPV), in form of $\{P_1, P_2 \ldots, P_n\}$, where P_i is a particular EPV.

A single knowledge element is represented in a predicate form like "*[entity to be affected E_0]* applying *[affector A]* results to *[affected entity E_A]*" (or [$E_0 + A \Rightarrow E_A$]) and accordingly takes a single record in the knowledge base. This form is used as a pattern in meaning that E_0 and E_A may be substituted by ***parameter P***, ***set of parameters {P_1, P_2 ..., P_n}*** (environment condition), ***state, set of states*** or ***situation***, and affector A may be substituted by an ***action, set of actions*** or a ***Strategy Plan*** (SP).

Here are some examples.

- an action affects a state: Si x action => S changed,
- an SP affects a state: {P1, P2 ..., Pn} x SP => P`1, P`2 ..., P`n}, etc.

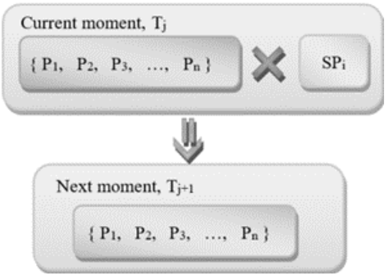


Fig. 1

Having a number of historical records of knowledge elements with the same type of initial entity to be affected, the same affector and the same type of affected entity, the knowledge base agent might optimize available knowledges by squeezing the statistical information from E_0, from the A affector force, and assessing the ***qualitative*** affection on E_A. After that number of records may be replaced with the optimized record, which will also optimally shrink the knowledge base size.

2.2. ACQUISITION OF RGT PROBLEMS

Developed RGT Solvers are able to acquire the knowledge from experts by the provided interface. Based on this we concentrate on developing adequate constructive mentals that extend OOP class and aim to provide natural language classifiers abilities, thus currently provide certain types of classifiers with have/be/do appropriate for now relations, which can be extended per required new classifier relations by experts.

Each classifier defined by the expert can contain its relations with other classifiers, have virtualization similar to OOP virtual/abstract classes and provide regularities and dependencies required to classify an instance of that.

The user also provides the type of classifier identified as one of defined in [12] types. As showed in [5, 13] RGT Solvers define several types of classifiers to make it adequate to expert classifiers presentation, where definition of each problem is started from the simplest ones, or in nuclears in Solvers. Nuclear classifiers provide simple classification based on regularity defined in it or based. Compositions of nuclears can describe basic classifiers, i.e. the simplest ones that appear in situations (e.g. figures in chess, battling units in battles).

```
{
  "id":"",
  "type":"nuclear",
  "parent":"",
  "rule":{
    "operation":"",
    "value":""
  }
}
```

Fig. 2 Nuclear classifier structure in JSON format

2.2.1. To achieve to a regularity of expert knowledge acquisition for RGT Games an interface has been developed [14, 15]. The interface is designed to acquire an expert knowledge in a form of patterns (classifiers) and transform situations from natural to symbolic presentations. Classifiers are used to define classes as well as operations, thereby providing a considerable uniformity of the structure of the language. Classifiers are composed from attributes which can be filled with objects that instantiate other abstracts.

2.2.1.1. New classifiers can be built on the basis of already existing ones both by composition and inheritance. For now, we provide single inheritance approach, consequently the semantics of inheritance are easily and simply defined. The derived classifiers inherit all the attributes of the parent class including incorporations, as well as enable activation of actions defined for the parent. It is also allowed to add new attributes and modify the relations between attributes of the child. However, one shall keep in mind that the instantiation of a derived classifier will check all regulations defined in the parent. This means that if in the parent classifier for the attribute X there was a condition like: X > 5 and in the derived it is modified to be: X

= 4, then no instance will ever get constructed for the derived one because the intersection of above two conditions is an empty set.

2.2.1.2. The language also supports the concept of virtualization. The latter is pretty generalized and is available for all kind of classifiers in contrast to previous versions, where only composites were enhanced to define virtualization. This means that properties of the classifier are not all known and that full specification may occur at a later time.

Syntactically, a classifier is virtual if it contains at least one attribute with undefined relations (the attribute condition is undefined if it is set to equal to '?' special value). To make the instantiations of such kind of classifiers one has to inherit from them and specify undefined conditions/relations. This procedure is called specification. Virtual classifiers serve as shortcuts and can be used in other classifiers as attributes. We call this usage of. There can be more than one virtual attribute used, consequently the instance of the classifier can be built from various compositions of specific attribute instances.

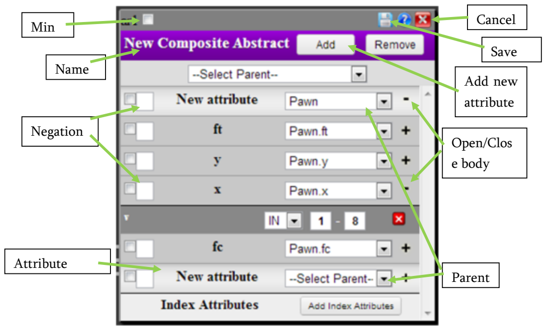


Fig. 3 Interface to define a classifier

2.2.2. EXPERIMENTING

Let's discuss some examples for more clarity.

2.2.2.1. The nuclear classifiers for chess are coordinates (X and Y), figure type and figure color, combination of which provides description of the chess field and the situation is provided by 64 fields descriptions, thus, 64 tuples of dimension 4.

Nuclear classifiers include one rule. Template of acquisition of nuclear classifiers is provided in Figure 2.

We demonstrate acquisition of classifiers relations by the example of chess classifier 'field under attack of knight' (figure). It is a composite classifier, which is a derivative of 'field under attack' and includes attributes (relation have) a) 'field' and b) 'move knight' where the 'field' inside 'move knight' ('moveKnight.field') is the same as the 'field' inside this classifier. That condition is being checked in the rule of nuclear attributes of 'fieldUnderAttackofKnight.field'.

2.2.2.2. At this stage we put some limitations on battlefield problem, like discretizing actions and limitations on newly appearing units.

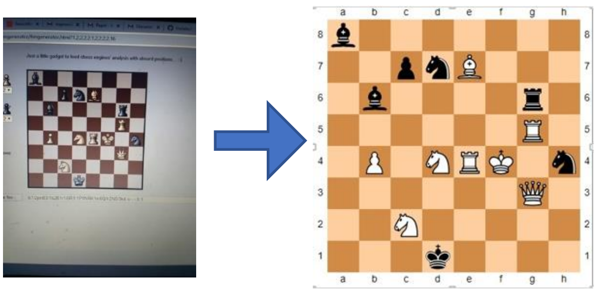


Fig. 4 Result of chess situation transformation

Here we consider that nuclear classifiers are unit type, unit location (coordinates in the given situation) and whom does the unit belong to. In this term, this is similar to chess and the basic knowledge acquisition and situation input is being done similarly to it.

Here we add some more nuclears, such as power of attack, minimal and maximal distances of attack and radius of damage, "health" of the unit (also discrete). Military units are considered as minimal classifiers. Actions for which are "attack" and "move" for each type of military unit and the postcondition for attacks can be noting that the target is destroyed or damaged.

2.2.2.3 Marketing and management problems can also be brought to the RGT class by splitting their specifics into corresponding terms of RGT meaning and defining them to meet the needed requirements of the class [19]. The goals in these problems are to get aimed values of profit, market share, achieving company stable state, enough conditions for survival in the market, needed rating in the market, etc. As management goals it might be acceptable level of feedback, low wastes in spending, time or/and other resources, etc.

2.2.2.3.1 Classifiers for marketing problem are: a) companies (competing against each other), which are actors with certain nuclears depending on their type, b) actions, which are changes of the product price and quality, c) situations, which are determined by the states and actions of competition. On each turn, states are determined by the set of parameters of current competition and scenarios of it, i.e., the competition template formed from the conceptual basis of management theory.
2.2.2.3.2 The management problem is considered on the example of customer-company-supplier model. There are two separate parts in this model – in the first, the company deals with customers and in the second it deals with suppliers.

In this model several companies (agents) compete in a market of certain product that can be assembled in predefined number of configurations. Configurations depend on types of limited main supply components used. Every main component is being produced by the number of supplier brands and available in couple of qualitive options (highspeed/lowspeed, big/small, cheap/expensive, etc). Besides, there are limitations in compatibility between certain main components.

The classifiers here are a) customers, which compose the demand of different types of the product, b) companies (agents), with their daily assembly abilities, bank accounts and warehouse, c) suppliers, which produce the supply components with their daily production capacity number of quality options for that components. The actions defined for different classifiers above are like "offer a product with X configuration and Y price", "request the X component from Y supplier for Z price", "accept offered X component from Y supplier for P price", "produce X number of product from listed components during N days", etc.

2.3. MATCHING CLASSIFIERS TO SITUATIONS

As described above situations present groups of nuclear classifiers of specified situations, however natural presentation of situations is different, thus a tools for regular transformation of input situations to solver required symbolic presentation are developed.

2.3.1. SITUATION INSERTION INTERFACE

RGT situations are presented in symbolic way including groups of nuclear classifiers. We have developed a system for transforming natural presentation of situations, which we consider as images to their symbolic presentation. The system and algorithm on which it is based are provided in [15].

The system uses Neural Network MobileNet SSD [16] for detecting actors, particularly, figures in chess and military units for battlefield problem. Afterwards, by comparing its coordinates it creates the symbolic presentation of situation, which is considered as situation input in RGT Solver.

We have developed situation transformation system for chess and battlefield: a) input for chess is an image with existing chess board and b) input for battlefield problem is a screenshot from strategic game "Generals: Command and Conquer".

2.3.1.1. The transformation system includes detection of objects (figures, military units) from the given image. Afterwards, the system calculates the approximate shape of the field and by comparison of detected objects' coordinates it constructs the initial field (board, military area).

For chess, the NN was trained on ~1000 images including chess board of different chess figure and board shapes, which ensures the universalization of the model. Accuracy of the model is pretty high, approximately 95 % of figures match to their exact place on final board.

2.3.1.2. For Battlefield problem, the NN was trained on ~500 images of screenshots from the game 'Generals: Command and Conquer", including 17 different classes of vehicles and military units: tanks, rocket launchers, soldiers, medical vehicles etc. The model is not as accurate as in chess due to some similarities of units (for example, some types of tanks), some similarity between not marked and marked objects etc. However, the results of NN are acceptable at this stage: detection precision is 76.3%, detection recall is 87.4% and classification accuracy (for correctly detected figures) is 96.9%. Currently, the model is being enhanced by more data collection and more detailed annotation.

2.3.2. MATCHING ALGORITHMS

For the already symbolic presentation of situations Solvers are able to match developed classifiers in two main directions: 1) matching network of classifiers and 2) matching certain required classifiers [12].

2.3.2.1. Matching of nuclears is as simple as checking the rule in it.

Minimal classifiers and composites are done by both children matching and attributes full set matching. Virtual classifiers are always matched by their children and specifications are matched by only parents. Actions are matched by matching of precondition and the actor. Once activated they can be applied on the situation to change it.

Dynamic classifiers are matched if precondition is matched and postcondition for all of the leaf situations is matched.

Goals precondition shall be matched in order to consider the goal doer, after which a game tree is generated and final situations are evaluated if an evaluation function is defined.

2.3.2.2. Certain given classifier matching, then only the subnetwork of classifiers which relates directly to the given classifier is being matched in reverse direction (starting from the required classifier to the nuclears which are needed to match it).

3. DECISION MAKING AND ACTION SELECTION FOR GIVEN SITUATIONS

For the strategy construction Personalized Planning and Integrated Testing (PPIT) algorithms are used, which create strategies using plans and are adequate to expert approaches. Plans are certain general descriptions of strategies. Each plan represents chains of goals sorted by their priorities. Those are the goals which are attempted to be achieved in given situations while playing by the plan. The PPIT program was designed in [1] and integrated in Solvers were provided in [5]. We enhanced Solvers by trajectory-zones based technique (TZT) [4] algorithms which provides adequate solutions for achieving a goal originally suggested to estimate utilities of captures only of the opponent pieces. For example, to choose capture with max utility TZT chains the moves to each piece of the opponent (trajectories) without accounting possible handicaps for real capturing then using all available knowledge “plays the zones” of the game tree induced by the trajectories followed by estimation of their values to choose the best.

Realizing the current plan, the shell can determine the goal in the agenda which in turn determines basic attributes to be considered followed by indication of the arguments of those attributes. In the following we provide integration of TZT into Solvers.

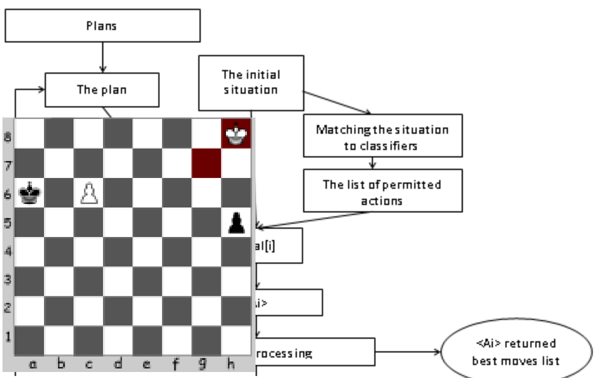


Fig. 6 Retie etude
situation, white to move rocessing algorithm

3.1. EXECUTION OF PPIT

The algorithm of the realized PPIT algorithm generally speaking has the blocks provided in Figure 4. So, for each Plan that is appropriate for the given situation by some classifiers (a precondition for the plan), it starts searching actions that bring to the goal with the highest priority in the chain. If the goal is achieved the processing of plan is finished, otherwise further goals are being processed.

Processing of goals in previous versions of PPIT [4] was implemented by generating trees without deep analysis of classifiers in it. The current goal achievement algorithms will be described in next section.

The adequacy of PPIT algorithms are experimented in chess, particularly in Retie etude [1, 3], for which the algorithm using the given plan can find a solution by processing ~15 times less positions than chess engines, such as Stockfish chess engine. The goal is to achieve a draw in the given Figure 5 situation.
The considered number of positions is brought in Figure 6.

	RGT Solver18
Move 1	22
Move 2	27
Move 3	27
Move 4	10
Move 5	9
Move 6	9
Total	104

Depth	Considered Nodes
1	36
2	136
3	570
4	1088
5	1414
6	2183
7	3309
8	4005
9	6717
10	9256
11	16987

Fig. 7 Considered positions in Retie etude solution left: in RGT Solver, right: Stockfish engine

3.2. INTEGRATION OF TZT

For going deeper into the game tree, various types of tree cutting solutions are applied, including alpha-beta pruning. For chess engines based on some defined knowledge they might drop many branches and achieve big depths. We follow the line suggested by Botvinnik [9] and developed within the intrusion protection problem algorithms [4] which rely on trajectories of attacks and zones of counteractions named TZT.

Idea defined in [9] and enhanced in [4] states that The trajectory of an attack is a subtree Ga(S', P') where S' is a subset of the system states $S' \subseteq S$ and P' is a subset of the actions, consisted of an offensive's conversion procedures Pa and a defender's normal conversion procedures Pdn, i.e. $P'=Pa \cup Pdn$, $P' \subseteq P$, $P' \neq \varnothing$ and The zone of counteraction is a subtree Gz(S", P") built around the graph of the trajectory of an attack Ga(S', P'), i.e. $Ga \subseteq Gz$, where S" is a subset of the system states $S" \subseteq S$, which belong to the trajectory of an attack, hence $S' \subseteq S"$, and P" is a subset of the actions, which consist of the conversion procedures, defined on the trajectory of an attack, P' and the defender's special conversion procedures Pds, i.e. $P"= P' \cup Pds$, $P" \subseteq P$, $P" \neq \varnothing$, hence $P' \subseteq P"$.

3.2.1. We implement the TZT solution into RGT Solvers PPIT algorithms to effectively achieve goals.

So, goal achievement checking is performed by the following steps: a. Generation of tree of situations with the defined depth that lead to goal achievement, b. Extension of situation chains by all possible counteractions by the opponent, c. Checking if goal can be achieved and evaluation if defined.

Let's consider the following example (Figure 7) where black goal is to take white pawn in vertical 'f'. Let's init the depth of tree with 6, which means 6 depth tree can be generated at most, evaluation of situation is done by materials. Following sequence of actions by achieve the goal: 1) Rf6-> Rxf5, 2) Rh5->Rxf5, 3) Kf4->Kxf5, 4) Kg4-> Kxf5, etc.

At step a. algorithm generates all the possible sequences of situations with the maximum (depth of tree) / 2 distance that can achieve the goal. The restriction of (depth of tree) / 2 is because it can perform only that number of actions in the tree, the rest will be opponent's actions.

At step b. algorithm generates all the possible counteractions by opponent for each sequence of actions and actions opposing counteractions of opponent.

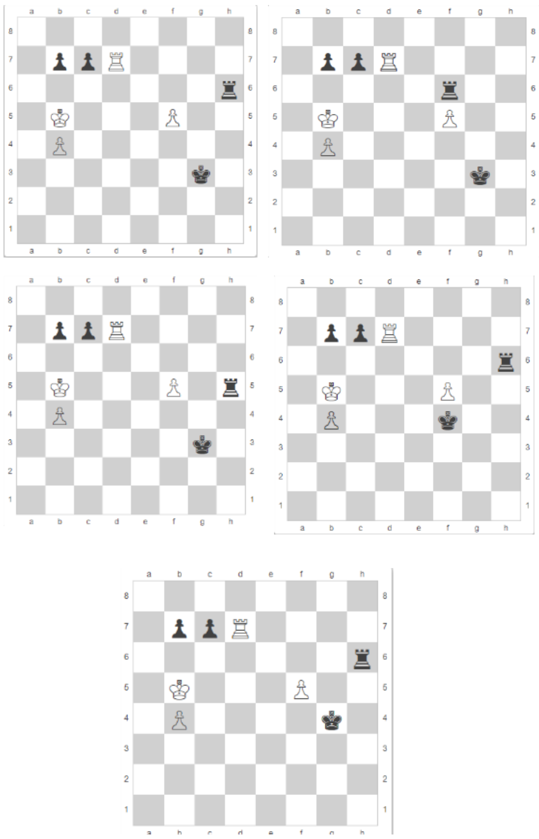


Fig. 8 Example of processing TZT

1)Rf6->Rd5->(c6+->Kb6->cxd5+->Kxb7)->Rxf5, 2) Rh5->Rd5->(c6+->Kb6->cxd5+->Kxb7)->Rxf5, Rh5->Rf7->Kf4->Kc5->Rxf5 3)Kf4->Rd4->(c6+->Kb6->cxd5+->Kxb7)->Kxf5, Kf4->Rf7->Rh5->Kc5->Rxf5 4) Kg4->Rd4->(c6+->Kb6->cxd5+->Kxb7)->Kxf5, Kg4->Rf7->Rh5->Kc5->Rxf5

3.2.2. It is obvious that there are many actions not considered when generating the tree for the goal (we can consider 4 different situations at first level, however there are 25 situations). If for the given situation we assume average number of moves by a side is ~20, the overall 6 depth tree would have 20^6 situations by brute force processing, however if we consider ~4 different situations on each level it would have 4^6 situations which is radically smaller number of situations.

At step c. all the final situations are evaluated to consider the best ones and select the branches as the best, accordingly suggesting the actions that bring to the expected best branches. This list is passed to the next goal in the plan chain as it is described in [1, 3] and similarly being processed with steps a, b, c for the goal.

4. CONCLUSION

a. RGT Solver is able to acquire problems and provide adequate to expert matching to classifiers based on systemic classification, while situations provided by natural presentation, say an image can successfully transformed to Solver required presentation by developed tools in Solver interface.

b. Integrated planning algorithms for decision making are described including PPIT and TZT algorithms by effective adequate to experts' approaches.

c. The ability of acquisition of various RGT problems, including chess, consumer dialogue, battle field, marketing and management problems are demonstrated.

REFERENCES

[1] E. Pogossian, V. Vahradyan and A. Grigoryan, "On competing agents consistent with expert knowledge", *Lecture Notes in Computer Science, AIS-ADM-07: The International Workshop on Autonomous Intelligent Systems - Agents and Data Mining*, pp. 229-241, St. Petersburg, Russia, June 6-7, 2007.

[2] E. Pogossian, "Specifying Personalized Expertise" International *Association for Development of the Information Society (IADIS): International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006)*, pp 151-159, Barcelona, Spain Dec 8-10, 2006

[3] S. Grigoryan, "Structuring of Goals and Plans for Personalized Planning and Integrated Testing of Plans", *Mathematical Problems of Computer Science*, vol. 43, pp. 62-75, Yerevan 2015.

[4] E. Pogossian, A. Javadyan, E. Ivanyan "Effective Discovery of Intrusion Protection Strategies". *Workshop on Agents and Data Mining*. pp. 263-274, St. Petersburg, Russia 2005.

[5] K. Khachatryan, S. Grigoryan and T. Baghdasaryan, "Experiments validating the Be-Have-Do meaning presentation model and matching algorithm for competing and combating problems", *International Conference in Computer Science and Information Technologies*, pp. 155-159, Yerevan 2013.

[6] E. Pogossian, "Effectiveness Enhancing Knowledge Based Strategies for SSRGT Class of Defense Problems". *NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems.* p. 16, Salamanca, Spain, 2011.

[7] E. Pogossian, D. Dionne, A. Grigoryan, J. Couture and E. Shahbazian, "Developing Goals Directed Search Models Empowering Strategies Against Single Ownership Air Threats," *International Conference in Computer Sciences and Information Technologies*, pp. 155-163, Yerevan, 2009.

[8] E. Pogossian, "Combinatorial Game Models For Security Systems". *NATO ARW on "Security and Embedded Systems,* pp. 8-18, Porto Rio, Patras, Greece, 2005.

[9] M. Botvinnik, "Computers in chess: solving inexact search problems", *Springer Series in Symbolic Computation, with Appendixes*, Springer-Verlag, New York, 1984.

[10] E. Pogossian, “Constructing Adequate Mental Models”, *Mathematical Problems of Computer Science*, vol. 50, pp. 35-51, 2018.
[11] E. Pogossian, “Towards adequate constructive models of mental systems”, *International Conference in Computer Sciences and Information Technologies*, pp. 37-46, Yerevan 2017.
[12] S. Grigoryan, N. Hakobyan and H. Vrtanesyan, “Object-Oriented Modeling of Matching to Systemic Classifiers”, *Mathematical Problems of Computer Science*, vol. 49, pp. 115-121, 2018.
[13] S. Grigoryan and N. Hakobyan, “Experimenting with Acquisition of and Matching to Systemic Classifiers”, *Mathematical Problems of Computer Science*, vol. 50, pp. 96-103, 2018.
[14] Z. Naghashyan, and E. Pogossian, "Developing Java Software for Representation, Acquisition and Management of Strategy Knowledge", *Mathematical Problems of Computer Sciences*, pp.187-195, Yerevan 2010.
[15] N. Hakobyan, “A System for Transforming Images to Symbolic Presentation for Combinatorial Defense and Competition Problems”, *NAS RA and NPUA Series of technical sciences*, vol. LXXII, N2, pp. 199-209, Yerevan 2019.
[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, Ch. Fu, A. Berg, “SSD: Single Shot MultiBox Detector”, *ECCV*, 2016.
[17] N. Hakobyan, “Enhancing adequacy of systemic classifiers by personalization and cases”, *RAU Bulletin* № 1, pp. 35-48, Yerevan 2019.
[18] T. Baghdasaryan, “Forming and Representation of The Knowledge Base of Trading Agent”, International Conference in Computer Science and Information Technologies, Yerevan 2011.
[19] T. Baghdasaryan, “Adapting RGT Solver Interface to Management Strategy Search Problems”, Mathematical Problems of Computer Science, pp. 135-145, Yerevan 2013.

6. Hakobyan N., “**A System for Transforming Images to Symbolic Presentation for Combinatorial Defense and Competition Problems**”, ISSN 0002-306X. Proc. of the RA NAS and NPUA Ser. of tech. sc. 2019. V. LXXII, N2, pp. 199-209.

Abstract
We aim to develop tools for regular transformation of combinatorial defense and competition problems situations to their symbolic presentation by machine learning solutions. As it is proven, that RGT class problems are reducible to each other, in this paper we demonstrate developed ANN model to detect chess board from an image and classify chess pieces. The model we developed is simpler than ones we have researched, but it has the same accuracy and is more universal.
Keywords: Neural Networks, Image Classification, Object Detection, Systemic Classifications, Algorithms

1. Introduction

1.1. Background of the research

Cognitive Algorithms and Models research directions include the ones in Artificial Intelligence and aim to develop constructively regularized models of human approaches in solving combinatorial problems.

There are different lines of researches in this area, e.g. machine learning solutions such as neural networks, concentrate on modeling of biological nature of human brain.

Following the line of Cognitive Algorithms and Models research directions, we concentrate ondevelopment and applications of cognitive functions to a class of combinatorial problems defined as problems where spaces of solutions are Reproducible Game Trees (RGT) [1, 2, 3].

1.2. RGT class

RGT class includes important problems like computer networks intrusion protection, optimal management and marketing strategy elaboration in competitive environments, defense of military units from a variety types of attacks, communication problems, certain types of teaching, chess and chess-like games [2].

In the continuous researches of our team a class of problems is defined as a class of unsolved combinatorial problems [3].

The class named RGT is a sub class of Optimal Strategy Provision (OSP) problems. RGT problems meet the following requirements:

- there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions in the (c) specified types of situations;
- there are identified utilities, goals for each actor;
- actions for each actor are defined.

Actors perform their actions in specified periods of times and do affect situations by actions in time t by transforming them to new situations in time t+1 trying to achieve the best utilities on that situations (goals) by regularities defining these actions.

1.2.1. Achievements in RGT and some open questions

There are certain achievements for RGT class, some of which are listed below:

Suggested in [4, 5] theory of mental doings provides ways for constructive and adequate models of various cognitive functions, e.g. classification, explanation, etc. We have implemented a knowledge-based expert system, RGT Solver18, which is able to utilize those cognitive functions.

Developed software, RGT Solver 2018, can examine developed models for systemic classifiers of RGT problems: algorithms and structures have been developed to provide an adequate description of systemic classifiers and to ensure their correspondence [6], and their adequacy were demonstrated by experiments [7].

RGT class combinatorial problems are reducible to each other [8].

The Results of the work are appliable to the actual problems of real-time detection of means of attack and protection of the enemy with the help of autonomous drones and making the best decisions of counteraction (attack, retreat, investigation) in such situations. In [9] defense of navy from air threats is described as RGT class problem and certain solutions are provided.


```
[{"cx": 0, "cy": 0, "fc": 2, "ft": 4},
{"cx": 0, "cy": 1, "fc": 2, "ft": 3},
{"cx": 0, "cy": 2, "fc": 0, "ft": 0},
{"cx": 0, "cy": 3, "fc": 0, "ft": 0},
{"cx": 0, "cy": 4, "fc": 0, "ft": 0},
{"cx": 0, "cy": 5, "fc": 2, "ft": 4},
{"cx": 0, "cy": 6, "fc": 2, "ft": 6},
{"cx": 0, "cy": 7, "fc": 0, "ft": 0},
{"cx": 1, "cy": 0, "fc": 0, "ft": 0},
{"cx": 1, "cy": 1, "fc": 2, "ft": 2},
...]
```


Figure 1: Symbolic Presentation in Solver

The natural situations of different problems are presented differently, while the transformation of natural situations (images) into software-presented ones (symbolically) is another widely discussed topic of the study with frequently provided solutions of machine learning [10, 11].

Particularly, in [11], author introduces new method of object detection, which can be useful for some RGT problems situations processing and ensures effectiveness and fast-working process. It also detects hidden objects, which cannot be detected adequately by human without any tools. However, the solution is provided for sequential frames processing and not for single images.

We consider the transformation of a situation to Solver as a computer vision problem, particularly, image classification and object detection. Since Neural networks are the leading method of solving such problems all over the world, we have also chosen them as a tool.

Currently, the first layer of Solvers is a symbolical input (fig. 1), which is done by experts/programmers.

Figure 2 Natural Presentations of Situations: Battles (Left) and Chess (right)

Natural presentations of situations are different (fig. 2) and, currently, there is no way to pass image presentations of situations to Solver in a regular way. Thus, in current work weaim to develop an interface for regular transformation of natural presentations of situations into Solver environment. The problem can be divided into two subtasks:

a. Detect situations from the given image.

b. Modify situation to acceptable for Solver form.

1.2.2. Battle Field as RGT class problem

Figure 1 Actors and Actions in Battle Field Problem

Battle Field can be considered as RGT problem by following interpretation (fig. 3):

1. The battling sides can be considered as interacting actors
2. Military units' movements, attacks can be considered as actions
3. The battle field area including military units can be considered as situations
4. Different situations can be considered as goals: capture objects, destroy enemy units, push frontline.

1.3. Related Works

During our research several works that perform to chess board detection and piece classification were considered. In the most of researches (3D objects) board detection is done via image processing and not machine learning using already developed model from opencv. (canny edge detection and hough line detection) [12].

In [13] author describes develops model for 3D objects/images, using mentioned opencv model for board detection and Caffe deep learning framework and pre-trained AlexNet [14] and achieved pretty high 99 % accuracy. The shortcomings of the model are a few: a) the model is too dependent to certain construction of board and pieces, i.e. it works badly for pieces and board of other construction than the training set is.

Chessify project, launched by Fimetech LLC([15]) in 2016 provides solution for chess board detection and piece classification for 2D objects with pretty high accuracy. However, Chessify is the best among similar solutions worldwide by universalization, it still supports only 2D detections and is not open-source.

In [16] author's described model for chess board detection avoids using opencv module but includes manually selecting of the four corners of the board in the specified order, which we also aim to automatize since in other than chess situations it might be hard to specify boundaries of the situation manually. However, the model uses SVM for training and is simpler than other models, the accuracy of pieces classification is much lower (~85 %) and it is still dependent of certain shapes of figures (training dataset).

All described models and others researched have several disadvantages that we aim to avoid, if possible, while building our own model – a) dependence over certain shapes of board and pieces (the most common issue); b) complex models; c) low accuracy.

1.4. Current work

Currently, the input layer in RGT Solver is provided by experts/programmers in symbolical way.
We aim to develop tool for regular passing of natural (image) presentations of RGT situations to symbolical ones in RGT Solver. For this purpose, we are using ANN.
As it is proven, that RGT class problems are reducible to each other, we are providing experiments for chess, so current work concentrates on the chess situations and pieces.
Existing models offer classification for certain types of pieces – training of NN was done on exact board and pieces. Its achievement is 99 % on test (same types), but if we change figure shapes somehow (take another board and other pieces with different shapes) result would be lower. We aim to enhance classification to be universal for detection of different shapes of figures.
For users' convenience, we have developed models both for 2D and 3D images. For experimenting we concentrate more on 2D objects.
We discuss parallels and possibility of transmission of achieved results of chess to battle field problem in this paper as well.
We provide two algorithms for our purpose – each of them will be described in separate sections.

2. Simple Algorithm

The task is following: Given the image of chess board it is needed to classify each field as an empty one or a certain figure depending on its color and type (black rook, white pawn etc.).
In other words, this task can be stated as follows: to classify nuclear classifiers from the situation, which are attributes of pieces for chess: color, type and coordinates.

2.1. Chess Piece Classification

2.1.1. 2D images

2.1.1.1. Dataset

A dataset of around 300 images of chess boards with existing figures on them was collected. A Python script was written to split the board into 64 equal squares (8 rows, 8 columns). Pieces constructions are of various types which insures universalization of the model.Each of the 64 received images of every picture includes either empty square or figure with certain color and figure type, which was annotated. The dataset was split randomly into training and testing sets by 3:1 ratio.There are overall 13 classes numerated from 0 to 12, where 0 corresponds to an empty field, 1 to 6 is for white pawn, bishop, knight, rook, queen and king and from 7 to 12 for black pieces with the same order.

2.1.1.2. Learning

Keras was chosen as the neural network library for our model, which was trained to classify pieces on images on the described dataset. VGG-like [13] convolutional network with the following construction is selected: 2 convolutional layers with 32 neurons and kernel size of (3,3)

- Max-Pooling layer with pooling size (2,2)
- 2 convolutional layers with 64 neurons and kernel size of (3,3)
- Max-Pooling layer with pooling size (2,2)
- Flattening the 2D arrays for fully connected layers
- 3 Dense layers with 256, 128, 64 neurons correspondingly and with RELU activation.
- The last layer is dense layer with 13 classes (for each of classes of our classification) and Softmax activation.

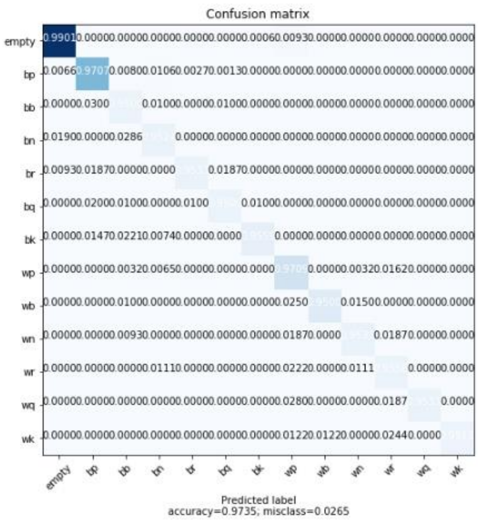


Figure 5: Confusion Matrix of Piece Classification

2.1.1.3. Results

Accuracy of 97.3 % on the test dataset of piece classification was achieved.Corresponding confusion matrix is provided in fig. 5.

2.1.2. 3D images

[13] Was taken as the base for this work.Dataset was collected as combination of datasets provided in [13, 16] and manually collected dataset of various constructions, which insures some universalization of the model.As long as all our models are built in Keras, and there is not built-in model for AlexNet there, we have chosen as our model VGG-16.Accuracy of 94.2 % is achieved.

3. Enhanced Algorithm

The algorithm includes several steps

3.1. Detecting Pieces

The dataset of chessboard images including chess pieces and empty fields was collected. All chess figures were marked and labeled in each of these images using LabelImg [18]. We have built ANN based on collected dataset to detect chess pieces from the board.

3.2. Coordinates Comparison

After detecting pieces, we have a set of pieces with coordinates (x1; x2; y1; y2).
For farther simplicity let's denote following:

$$F - setofreceivedfigures$$
$$X, Y: (R; R) - setofreceivedcoordinates$$
$$coordinates(x_i'; x_i'')correspondtothefiguref_i$$
$$A = avg(z'' - z'), for(z'; z'') \in XUY - themeanfieldsidesize$$

The next step is to understand how many rows or columns are between certain two figures. As long as all the fields' dimensions are nearly equal (the more the angle of shooting differs from 90o the less equal they are), we use the following equation to determine number of rows between figures f' and f'' with coordinates [y1', y2'] and [y1", y2"] (we consider that y1'>y2"):

$$\frac{(y_2' - y_1') + (y_2'' - y_1'')}{2xA} < r + p$$

Where r is the number of rows between figures and p is estimated parameter, which depends on the angle of shooting (the more the angle of shooting differs from 90o, the more the parameter should be). The value of r is equal to the minimal value from Z, the substitution of which into this equation will satisfy the condition.

$$r =]\frac{(y_2' - y_1') + (y_2'' - y_1'')}{2xA} - p[\qquad (1)$$

We use estimated value of 0.2 as p, which can be changed further, but seems successful choice for initial experiments.
The same is being done for columns.

3.3. Creating Initial Board

Now it's time to merge received figures together. The algorithm works following: one of the pieces is taken as an initial point with row index and column index equal to 0; on each iteration the program chooses not considered yet figure and count number of rows and columns by equation (1) and saves the results as a tuple (f, r, c), where f is figuretype (black bishop, white rook etc.), r and c are indexes of row and column correspondingly (can be negative).
After all iterations, the program gets the minimum number of all column indexes, if it is negative, it sums all received tuples third element by the absolute value of the minimum number. The same thing is being done for rows.
At the end we receive coordinates for each figure.
We fill the missing values with empty fields – for example, if we have some figures on coordinates (0,0) and (0,3), but there is nothing in (0,1) and (0,2), we fill them with empty fields.

3.4. Extending Board by missing lines/rows

If the initial board had empty corner row(s) or column(s) they would be missing after this steps and we would have matrix with dimensions 7x7 (if only one row or column was empty). The task is to find from which of 4 sides the empty line must be added.

Actually, this task is local for chess and probably would not appear in other RGT class problems. For instance, in Battle Field problem after detecting all military units, all 'fields' or 'areas' around (maybe with some limited range depending on zone where it is possible to hold fights) can be considered as empty. Of course, this brings another task, to detect, for example, is field a flatland, or which parts of area are plateau, which can affect possible movements there, but this is also a local problem and would be considered in future more detailed researches on Battle Field problem.

We take all empty fields we have in our received board and compare the images of those fields to each of 8 parts of a new line from top. If all of 8 parts matches with some empty fields in our board than we consider the line as the searched. We do the same for each side and we make as many iterations as many lines are missing.

Table 1: Comparison of the Results

	Simple Algorithm	Chessify	Building Chess ID	ChessVision
Accuracy	97.3 %	Unknown (high)	99 %	85 %
Universalization	Yes	Yes	No	No
Open-Source	Yes	No	Yes	Yes
Costly Training	No	Unknown	Yes	No
Used Library or Method	VGG like ANN	Unknown	AlexNet	SVM

4. Application in RGT Solver

After both modules for board detection and piece classification are ready, already classified figures and empty fields are being processed to JSON format as it is described in the below picture. Solver receives as an input a list of 64 JSONs (fig. 1) each of which refers to a certain field and contains nuclear classifiers' values.For example, the first raw of this image corresponds to the field with coordinates (0, 0), which is a8 on chess board, figure color on that field is black (2) and figure type is rook (4).Finally, chess situation in usual to Solver format is achieved and systemic classifications are being processed.

5. Parallels with Battle Field Problem

5.1. Classifying Units

For Battle Field problem tanks, rockets etc. can be considered as units.

Given in the image military units should be classified similar to units in chess

5.2. Difficulties

Some difficulties appear for transforming chess advances to battle field:

1. Quality data – if chess piece constructions are the same all over the world, the weapons are different in different countries. Anyway, the most popular types are of the same construction (e.g. tanks)

Figure 6 Military Units of Different Countries: Armenia (left) and Azerbaijan (right)

2. Classifying the actor/side – for chess it's simple – actor can be classified just buy piece's color. For battle field, it's harder, because battling sides' military units of same type difference is sometimes even harder to detect by human. These is still possible to realize, it's only making harder than the corresponding task for chess.
3. Angle – Angle of unit in an image and camera was not much important task for chess, as long as it is not hard to change angle for the photographer, while for battlefield sometimes it is possible to shoot only from limited number of places/angles. These also makes minimal required quality of the dataset higher.

6. Conclusion

Methods for situation's natural presentation transformation to Solver's symbolic presentation are suggested. Algorithms use Neural Networks for classification/detection of units and are described forcertain RGT class problem – chess. The first algorithm includes classification of chess pieces, the second one – detecting chess pieces from an image and detecting board by them. The discussed models were integrated with Solver18.Possible applications of solutions for battle field problems are described and some difficulties over battle field interpretation by chess solutions are listed. Implementation of the model for Battle Field problem, including dataset collection, researching ways for solution of described difficulties and ANN training are considered as the future steps.

References

[1] E. Pogossian, Adaptation of Combinatorial Algorithms, Yerevan: Academy of Sciences of Armenia, 1983, pp. 1-.293.

[2] E. Pogossian, On Modeling Cognition; in Computer Science and Information Technologies (CSIT11), Yerevan, Sept.26-30, 2011.

[3] S. Grigoryan, "Research and development of algorithms and programs of knowledge acquisition and their effective application to resistance problems", PhD, Yerevan, Armenia (2016).

[4] E. Pogossian, "Constructing adequate mental models", Transactions of IIAP NAS RA, Mathematical Problems of Computer Sciences, vol. 50, pp. 35-51 (2018).

[5] E. Pogossian, S. Grigoryan, N. Hakobyan, "On Systemic Classifications and Machine Learning", Computer Science and Information Technologies, 2017.

[6] S. Grigoryan, N. Hakobyan and H.Vrtanesyan "Object –oriented modeling of matching to systemic classifiers", Transactions of IIAP NAS RA Mathematical Problems of Computer Sciences, vol. 48. pp. 115-121 (2018).

[7] S. Grigoryan, N. Hakobyan "Experimenting with acquisition of and matching to Systemic Classifiers", Transactions of IIAP NAS RA Mathematical Problems of Computer Sciences, vol. 50, pp. 96-103 (2018).

[8] E. Pogossian, "Combinatorial Game Models For Security Systems", in NATO ARW on "Security and Embedded Systems", Porto Rio, Patras, Greece, 2005.

[9] E. Pogossian, "Effectiveness Enhancing Knowledge Based Strategies for SSRGT Class of Defense Problems," in NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems, Salamanca, Spain, 2011.
[10] He, K., Ren, S., Sun, J., & Zhang, X. Deep Residual Learning for Image Recognition (2016). CoRR, abs/1512.03385.
[11] R. Simonyan, Development of object detection, classification and positioning system, PhD, Yerevan, Armenia (2018).
[12] https://docs.opencv.org/2.4.13.7/doc/tutorials/calib3d/camera_calibration_square_chess/camera_calibration_square_chess.html.
[13] D. Yang, Building Chess Id, https://medium.com/@daylenyang/building-chess-id-99afa57326cd, 2016.
[14] https://en.wikipedia.org/wiki/AlexNet.
[15] https://chessify.me.
[16] J. Ding, ChessVision: Chess Board and Piece Recognition, 2016.
[17] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, Computer Vision and Pattern Recognition, 2015.
[18] https://github.com/tzutalin/labelImg.

Appendix 2: OJP in General

1.Pogossian E. **Management Strategy Search and Assessment Programming**. CSIT99, Yerevan, August 1999

ABSTRACT

Interrelated problems of establishing standards for management skill assessment and optimal management strategy search are studied. The deterministic and probabilistic models of the manager's behaviour are determined and the issue of their use to create skill assessment rating system is discussed.

The paper argues that present methods of measuring the effectiveness of management skill by using the simulation games are complex. Within certain constraints a theorem can be applied to the assessment to make it possible with the available computational resources. The possible positive impact of the matrix games, voting and other methods on the assessment are discussed. A unified view to the strategy search methods and their connections to the assessment ones are outlined. A class of knowledge-based strategies prospective for adequate management strategy search is defined.

THE MANAGEMENT SKILL ASSESSMENT PROBLEM

1. There exist established as well as still developing measures of complex human activities. These measures are important and so is problem of developing some appropriate scales for the assessment of the management skill.

We are studying standards of management skill assessment that satisfy the ABSEL's requirements for the skill measuring instruments: reliability, discrimination, convergence, universality [Anderson, Cannon, Malik, Thavikulwat 98].

The ABSEL Committee on Assessment was organized in 1995 to investigate the possibility for the use of simulation games as instruments for management skill assessment.

Two important research objectives were emphasized by the Committee:

- To be able to order the managers according to their on–the- job performances
- To reveal constructs responsible for on–the- job performances of the managers Our study is directed along the first objective and is aimed to construct adequate models and efficient assessment methods.

2.The problem is defined as the following.

Managers are competing in oligopoly markets for the same objectives, or criteria K, e.g. profit, market share, Return on Investment (ROI), success or not (*01success)* in the market, etc. The 01success criterion reflects in some degree more detailed profit, ROI, etc. ones.

Integrative performance of managers in possible competitions is evaluated by a method M. For example, M like robin round tournaments [*Moon*] can be a procedure of finding a competitor that has the best sum of performances by criteria K in all competitive oligopoly markets. We will name it as the *Maximum Sum (MS) method.*

We suppose that on-the-job performance of managers by 01success criterion and maximum sum method is inducing a linear, or "ideal", ordering O(01MS) of all managers.*

In the Management Skill Assessment (MSA) problem

given managers C1,C2,...,Cm, m>0, it *is required* to find criteria K and method M of the assessment of a competition such that an ordering O (K,M) of the C1, C2, Cm would be isomorphically imbedded into the "ideal" ordering O*(01MS).

Note, that MSA problem in its strong version, *MSA1*, may require to find identificators of the positions of the managers in accordance with the ideal ordering, or their *ratings.*

To resolve the MSA problem we need a constructive concept of the on-the job performance. We obtain it using simulation games.

We formulate the corresponding problem as the **Management Skill Game Assessment (MSGA) problem:**

Find a marketing simulation game G, an assessment criterion Kg and a methodology Mg such that the game competition ordering Og (G,Kg,Mg) of the C1,C2,…, Cm is isomorphically imbedded into the O*(01MS).

Current real market simulation games like CEO or GEO [*Thavikulwat,97*] are essentially increasing the quality of simulations and make them more like the competitions in the real business world.

We assume that real market simulation games are available and that the corresponding competitions with 01success criterion and MS method of assessment are similar to the original on-the job performances of managers and are inducing the same ideal ordering O(01MS).*

AN OVERVIEW OF OUR ASSESSMENT RESULTS

1.Given a pair of assessment 01success criterion K and the SM method M, we can play a series of real market simulation games for each competitor against all possible bundles of strategies in oligopoly competitions from any initial situation and then order competitors in accordance with their performances.

The results of such tournaments can be presented by (m, n) matrix – **Matrix of Grades** where m and n are the numbers of analyzed competitors and all competitive market situations, correspondingly.

We introduce a system of assumptions to construct an adequate deterministic model for the management skill assessment problem in form of corresponding Matrix of Grades in the next chapter.

2. As it was shown in [Pogossian,98] an ordering of competitors based on their game performances, or tournaments, is a computationally hard problem and the question of the appropriate constraints becomes essential in reducing that complexity.

In the deterministic model we suppose that managers' activities, in its essence, can be related to the behaviors of some programs and the ordering of those programs satisfy to the very important quasi-transitivity constraint:

any program in the ordering includes, to some degree, functions of its predecessors and, as a consequence, has a better performance.

A theorem is formulated that under this quasi-transitivity constraint the solution of the management skill assessment problem can be found with available computational resources.

A future development of the model is discussed to construct a scale of ratings of managers to assess them in a regular, standard way in the Centers of Management Skill Assessment, as well as to order a group of managers like MBA students. An important feature of the Matrix of Grades supporting this approach is in *independence of strategies from irrelevant alternatives* and *tests preserving an ordering of alternatives.*

The quasi-transitivity constraint is the analog of the criterion of "essentially improvable" strategies formulated in [*Pogossian,97*] for a symmetric problem of knowledge-based systems evaluation [Adelman, 92] and allowed to advance in strategies' efficient evaluation.

The preliminary version of the criterion has been applied earlier in 1979 to the strategy evaluation in two-person chess-like antagonistic finite games with perfect information [Pogossian, 85].

3. *In the probabilistic model* we suppose that managers have ratings that correspond to their skills and their behaviors depend randomly on their ratings. The model correlates with Elo's [Elo, 79] rating system developed and used worldwide for chess skill assessment.

It is supposed that initially the values of all ratings are equal. Then each rating is changing in the process of testing by test tournaments of the Matrix of Grades. The changes are proportional to the differences between expected and actually gained results of tournaments.

The process supposedly converges to the actual values of ratings of tested managers.

We outline the model in frame of Elo's rating system interpreted by Sadovski [Sadovski, 85] and discuss its application to the MSA.

4. To standardize the analysis of the Matrix of Grades and link it with other interpretations we replace the concepts “competitors” and “market situations” by more neutral and general ones: *alternatives* A1... Am and *testors* T1, correspondingly.

Thus, in a way, we are dealing with a multi-criteria evaluation and assessment problem which has applications in sport [Sadovski], matrix games [McKinsey] and voting [Moulin] and, therefore, there is a possibility of borrowing the knowledge from these areas.

We discuss the issue of the adequacy of some approaches used in the matrix games and voting for the skill assessment. More details on the assessment in voting are presented in [Danielian, Pogossian, 99].

The issue of tests that preserve orderings and some urgent research directions of it are outlined

BASIC ASSUMPTIONS OF THE ASSESSMENT DETERMINISTIC MODEL

1. To advance in solution of management skill assessment problems we need to make the following assumptions.

1.1. ***Assumption1.*** E*ach competitor is identified by a corresponding deterministic program.*

A competitor C competes in accordance to one of the available control p*rograms* which determine possible ways of its behavior. We suppose that during a competition each competitor selects one of the programs and keeps it unchanged. Given the competitors and the market one determines the outcomes of the competition deterministically.

1.2. ***Assumption2.*** *Competition in markets may be described by sets of situations, actions and strategies in discrete time periods.*

Competitors make allowed actions, or *moves*, from corresponding sets **A1, ..., Am** simultaneously, step by step and in T periods. We name a vector of such actions as a *bundle of actions*. Bundles of actions transform the initial situation S into sequences of new situations.

We call a tree of all possible sequential bundles of actions of competitors from an initial situation S in T periods a *S-game tree,* or *S-tree.*

In fact, S-tree is the sum of performances of all possible competitors' programs started from S. The whole performance of the programs may be described by the forest of such trees from different initial situations. To avoid technical complications in definitions we often assume to have only one initial situation.

We name the performance of the competitor C (i.e. the performance of corresponding program) in S-tree as a (*complete) S-strategy of C.*

1.3. ***Assumption3.*** *All competitors have the same sets of allowed market moves.*

Thus, any strategy of a manager in a competition will have the same number of terminal nodes equal to the number of possible samples of strategies of other competitors.

1.4. ***Assumption 4.*** *A competition, or a game, of competitors C1, C2, ..., Cm is determined by the sample of corresponding programs and by the initial situation.*

1.5. A game between C1, C2, ..., Cm results in a branch in each of strategies, or performance trees, of the programs. To evaluate a program, we have to use criterion K and evaluate all possible branches of the program which are representing its all possible games in all possible initial situations.

Assumption 5. *We evaluate the quality of a program by the forest of strategies generated by the program from all possible initial situations.*

Thus, to evaluate a program we have to consider all its possible games against all possible samples of other competitors in all possible initial situations. The number of competitors in samples depends upon the assessment objectives. For oligopoly competitions, for example, we have to consider all possible combinations of competitors which are in the oligopoly.

1.6. Since each competitor is represented by a program and the performance of the program in the S-tree is a S-strategy, in order to evaluate the quality of the program the assessment criterion K must be applied to that S-strategy. We suppose that the value of the criterion is determined by terminal nodes of the S-strategy. For example, we can use the profit gained by S-strategy as a criterion and calculate by averaging the values of the profit for all of its terminal nodes.

Assumption 6. The 01success criterion and the SM assessing method allow to order competitors in accordance with the O (01MS) by evaluating the terminal nodes of the corresponding strategies for all possible initial situations.*

2.Thus, if we order programs using 01success criterion and method MS, we will obtain an ordering consistent with, or isomorphically imbedded into, the ideal ordering O*(01SM).

For that we have to play a series of real market games for each competitor against all possible strategies in oligopoly competitions from any initial situation and then order competitors by MS method in accordance with their performances. That will be the solution of the Management Skill Assessment problem.

Since any game from an initial situation S corresponds to the appropriate terminal nodes in S-strategies, the above series of games will be represented by corresponding sets of terminal nodes of competitors' strategies.

To obtain the ideal ordering all terminal nodes of performances of all alternative programs from must be estimated and all programs have to compared with each other's.

As it was discussed in [Pogossian, 98] it is possible as a result of some robin-round tournament where each program C must meet with all possible samples of all possible strategies of competitors in all possible initial situations. The results of such tournaments may be presented by a (m*, n*) matrix where m* is number of all possible alternative competitors and n* is the number of all combinations of all possible strategies of competitors in oligopoly competitions from all possible initial situations.

Even for moderate values of the above parameters the evaluation of strategies by tournaments is a computationally difficult problem. An additional argument in favor of that comes from the reduction of that problem to the problem of program testing. Referring to the computational intractability of the program testing [Myers, 79] we can expand this result to the skill assessment problem too. One way to increase the efficiency of solutions may be to narrow the classes of problems>

INDEPENDENCE FROM IRRELEVANT ALTERNATIVES

1.Given m strategies C1, …, Cm do we need to analyze all m* alternative strategies to get an ordering of C1, …, Cm consistent with the O*(01MS)?

Luckily, we don't.

Criterion K is independent from irrelevant alternatives if the performance of any strategy in any competitive market has the same value independent from other analyzed strategies.

Criteria of a profit, market share, ROI and 01success are independent.

Method M is independent from irrelevant alternatives if for any sets of alternative strategies B1 and B2 and alternatives a, b from the intersection of B1 and B2 we have

aO*b in B1 ⇔ aO*b in B2.

Note, that there is some correlation in ideas of this independence concepts and ones analyzed in [Moulin].

It is evident, *that MS method is independent from irrelevant alternatives for the MG s produced by independent from alternatives criteria, particularly by the 01success criterion*

Thus, for any such MGs we can evaluate strategies independly and the problem of the high complexity of MGs resolution is reduced to a cutting of an amount of testing competitive markets.

In complete MG all markets are presented and they induce an ideal ordering. But how an ordering induced by representative markets coincides with the ideal one?

TESTS PRESERVING AN ORDERING OF ALTERNATIVES

1. *Tests* in classification and diagnostic problems specify measurements *sufficient* to identify target alternatives. *Minimal tests* specify *necessary* measurements for the same purpose.

In management strategy assessment problems, it is required to find minimal *tests that preserve an original ordering of the alternatives*.

2. Given MG (K,M) V we name *an **ordering preserving test for V*** (po-test for V) *any set of competitive markets of V preserving the ordering of its alternatives.*

This po-tests are based on the ordering of *representative* competitive markets induced by the ideal ordering of the programs.

An estimate of a length for such po-tests of an (m,n) MG is equal to m which is rather close to a lower bound of the po-tests is log m..

Let us describe an approach to find computationally acceptable tests that preserve an ordering of alternatives and based on using constraints. to the ideal ordering

THE TRANSITIVITY CONSTRAINT

1. To introduce constraints we need to standardize the sets of tournaments for each competitor.

In fact, each competitor C competes against samples of all possible strategies of other competitors, i.e. against all strategies, except its own ones. To regularize the analysis, we suppose that C competes against its copy, or a twin, too.

Accumption 7. Each competitor competes against all strategies including its own.

Thus, to get the ordering O(K,M) we may construct the **Matrix of Grades** (**MG**) where each row corresponds a competitor and each value in the row is determined by the results of the game of the corresponding competitor against one of possible samples of competitors in oligopoly competition from all possible initial situations.

2. To reduce the search in the MG we use **the Transitivity Constraint** (**TC)** on the ordering O*(K,M) which is formulated as follows

Assumption TC. Given programs Pi, Pj and sets of samples of competitors Bi and Bj losing to Pi and Pj, correspondingly, we have a tautology

Pi *is stronger than* Pj with respect to the ordering O*(01MS) *if and only if* Bi *includes* Bj .

TC is based on the assumption that *all competitors are essentially different in their skills.*

*Thus, to order Pi and Pj in the frame of the Transitivity Constraint **it is enough to find a sample of competitors that lose to Pi and win Pj.***

If the TC assumption were true the ideal ordering of programs would induce a corresponding ordering of sets of competitors Bi. Any sequence of the representatives of those sets would induce a scale of ratings of the skill of the management. To qualify a manager, it would be necessary to find the representative of the highest Bj that manager is able to win.

Note, that the Olympic and Swiss systems in sports are based at the similar criteria.

Unfortunately, the TC assumption is too specific and strong to be valid.

THE QUASI TRANSITIVITY CONSTRAINT

1. Let us analyze a more general than the Transitivity Constraint case where competitors may be close to each other by their skill level or even belong to the same class of equivalence in the O*(01MS).

Given that the competitor i is stronger than the competitor j in the O* let us use B(i,j) to denote the set of samples such that the competitor i loses to and j wins each sample of the B(i,j). In other words, *B(i,j) is the set of samples which in the games with the programs i and j attain results not consistent, or opposite, to the fact that i is stronger than j in the ordering.*

It is reasonable to accept that for fixed i and varying j ***the variance** function* #B(i.j) - the number of elements in B(i,j) , *will be bell shaped with respect to the distance between positions of i and j in the O*.*

Note. Our analysis is not statistical and the “variance” means a deviation from an expected value in accordance with disposition at the O*, and “bell shaped” refers only to the form of the curve of the functional dependence .Some parallels with the Elo approach [Elo, 78] were discussed in the following

chapters.
If for a given set of competitors the variance #B(i,j) is not zero then we cannot apply the Transitivity Constraint correctly but only with some errors. The relevance of the Transitivity Constraint increases if the distance between compared competitors i and j is increasing.
Let us approximate the bell-shaped variance curve by a rectangular one and formulate the following weak version of the Transitivity Constraint - the **Quasi Transitivity Constraint** (QTC).

Assumption QTC. Given programs Pi and Pj, samples of strategies Bi and Bj losing to Pi and Pj, correspondingly, and the variance function #B(i,j) it is possible to indicate constants *a* and *b* (small enough compared to the number of all strategies in the ordering O*) such that:

- *if j belongs to the segment [i+a, i-a] then #B(i,j) may exceed zero for no more than b points*
- *if j does not belong to the segment [i+a, i-a] then #B(i,j) is equal to zero and Pi is stronger than Pj with respect to the ordering O* if and only if Bi includes Bj.*

QTC appears as an analog of the "essentially improved" strategies' criterion in [Pogossian,97].
2. *Is the Quasi- Transitivity Constraint reasonable for managers?*
The Quasi-Transitivity Constraint appears naturally if considered for knowledge-based human activities, i.e. depending on the amount of knowledge or skill available for the decision making.
For managers in general as well as for marketing managers discussed in [Pogossian,97] we may support the above statement by following arguments.
First, management strategies are essentially knowledge-based. It is evident when we analyse the methodology of the construction of business or marketing plans: experienced manager differs from a novice first of all by the amount of knowledge he uses for planning.
Second, the skill of a manager is determined not only by the amount of the acquired homogeneous knowledge but, primarily, by the hierarchy of abstractions such that each new level is possible on the base of categories from the previous levels. This property strongly correlates with the ability of managers to improve their skills.
And at last, in the real management practice there are evaluations and classifications of the managers' skill which are directly reflected by the system of their rewards.

EFFICIENT ASSESSMENT FOR QUASI TRANSITIVE TOURNAMENTS

1. The QTC assumption allows to define some sufficient criterion for an efficient assessment.
The following theorem is true.
Theorem. *Given the Quasi-Transitivity Constraint for the ordering O*(01MS), a class F of competitors and competitors f and g from F , we say that f is stronger than g (i.e. the location of f is better than g in the ideal ordering O*(01MS)) if we find b samples of competitors such that f wins and g loses games against each of them.*

The proof of the theorem is similar to the one in [Pogossian, 83] for two-person games with perfect information. *Note,* that the condition of the theorem becomes not only *sufficient* but *necessary* too if #B(i,j) is equal to b for some pair i,j.

Thus, given the competitor f, the question of its strength relative to the competitor g is reduced to the construction of a special tournament and an estimation of the parameter b . *Even without the estimate of b it is true that with increasing the number of testing samples the likelihood of f to be stronger than g increases too.*

The theorem also supplies the necessary basis for an experimental investigation and a complete justification of the assumption.

CENTERS OF MANAGEMENT SKILL ASSESSMENT

The Theorem formulates sufficient conditions to order any two programs if we can indicate b competitive oligopoly markets such that the performances of one of the programs is better than the performance of the other in all of these markets.

As a consequence of the theorem we get another evident condition for assessing/ordering an *individual manager*/program. It is based on the po-tests - *representative* competitive markets induced by the ideal ordering of the programs that does not belong to the "uncertainty" zones, i.e. belongs to the sets of competitors Bi\Bij.

Po-tests determine a *scale of ratings* of management skills where a position in the scale and appropriate rating for any program may be found.

If the Management Skill Ratings (MSR) scale was constructed it would give some objective measure for management skill assessment and be the base for *Centers of Management Skill Assessment.*

The base of that scale construction could be the current computer models similar to the approach in [Chussil,Reibstein 94] that generate varieties of competitive markets.

Calibration of the scale may be realized by the recognized degrees of skills of managers like the skill degrees between different year students of the MBA colleges.

ASSESSMENT OF A GROUP OF MANAGERS

1. If a few managers must be assessed, for example a group of MBA students, we get an additional opportunity to create testing competitions by using subgroups of that managers. Let us outline the approach in MBA students skill assessment by games?

2.1. We suppose that students play tournaments in modern real market games like CEO or GEO [Thavikulwat,97].

In that games competitors/students C1, ..., Cm play oligopoly games and then the instructor/the program orders them by performances. Since all competitors participate in each of the games, the games can be interpreted as series of tournaments between each of the competitors and the sample of m-1 of the rest of them. *Thus, each game can determine m positions in the corresponding Grade Matrix.*

Different scores of any two competitors in such a game can be interpreted as winning and losing scores with the same sample of strategies. Since in the frame of the Transitivity Constraint one sample is enough to distinguish the competitors, the result of the game would order C1, ..., Cm with respect to the ideal ordering O*. The difference of the scores of the competitors appears to be correlated with the reliability of positions of the competitors in the ordering.

2.2. It is also interesting to interpret a game between C1, ..., Cm as a game of any subset of them in a market that is changing dynamically under the influence of the rest of the competitors.

It may be supposed that:

-*the winners and the losers of the above tournament by the MG belong to different classes of equivalence in the O* ordering, or have different ratings;*

-*as many intermediate grades there are between a pair of competitors in that tournament ordering the more is the probability that one of them is stronger than the other;*

2.3. Let us note that a repetition of a game for the same participants with different initial situations can test competitors for new samples of the MG and may increase a reliability/ certainty of revealed preferences.

Another way to increase the reliability is in increasing the coverage of the necessary number of tournaments from the MG by their selection with some distribution of probabilities.

2.4. Note also that to order a given set of competitors in sports robin- tournaments between pairs of competitors are used. Despite the fact that our business games are processed with simultaneous participation of the competitors we can interpret them as such robin-round tournaments and use their scores for similar orderings.

The robin-round tournament is perceived as an ideal but the most expensive method for ordering the competitors. It is interesting to reveal the relation between the degree of the approximations to that ideal ordering formed by some other not so perfect tournament and the computational resources that this "weak" tournament is using for the ordering.

ELO RATINGS APPROACH TO THE ASSESSMENT

1. In the above model the basic assumption was that deterministic programs can appropriately simulate managers' performances. To make the model more realistic we will introduce some probabilistic elements similar to the Elo rating system [Elo, 79].

The Elo's system was developed for chess players and accepted by the World Chess Federation. It is one of the most advanced systems among known standards to evaluate a competitive dynamic activity. We develop a model similar to the Elo system assuming that managers have ratings corresponding to their skills and that behaviors of managers randomly depends on their ratings.

Initially the values of all ratings are equal to some constant. Then ratings are changing in the process of testing by tournaments of the Matrix of Grades in proportion to the difference between expected and actually gained results of a testing.

The process is supposed to converge to the actual values of ratings of managers.

Let us outline the model in the frame of Sadovski's [Sadovski, 85] interpretation of the Elo's system.

2. The basic ***Assumptions*** of the Elo's system are the following.

- There exists a numerical measure R of the strength of chess players, *a rating.*
- The number of wins of a player *i* divided by the number of its loses to a player *j* which differs at *t* points in its ratings in the ideal orderings of chess players, i.e. R(i) – R(j) = t, may be approximated by an exponential function a^t (*a is in a power t)* with a constant a. *Consequence1.* Probability that i defeats j, or P(i beats j) = m/ m+n = a^t /1+a^t = f(t)
- We can differentiate chess players at their chess school level and measure that higher-level player defeats its neighbor with a probability of .75.

  Consequence2. If ratings of neighbor players differ by v units then P(i beats j) =a^v = .75. Supposing that v=200 we get that a =1.0055 and P(i wins j) = f(t) = (1.0055)^t , where t = R(i) – R(j) *Consequence3.*The function f(t) may be presented as a table, *Elo table*, which indicates expected numbers of winnings and loses of players that differ by t units in the ratings. *Consequence4.* If we know the old rating of a player Ro it may be modified after tournaments proportionally to the difference of the sums of actual outcomes Na of its performances in the tournament and expected ones Ne calculated by Elo's table.

For example, using formula: Rn(I) = Ro(I) + w(Na – Ne), where w = 10.

- A rating of new participants is accepted as equal to 2 000. Some process of assigning the initial ratings to players is assumed.

Elo had calculated ratings of the world's extraordinary players starting from Murphy (2690). In his system the 23 players got ratings more than 2600.

His system has 16 levels of tournaments depending on range of ratings: 2251-2275… 2626-2650. Let's say an international master (a gross master) playing in a level-10 tournament (2476- 2500) wins 47 (67) % of games.

Then 2/3 of the competitors must exceed rating 2250 to use the tournament for a qualification.

3. Following the Elo's ideas we suggest the following model for the management skill assessment.

We will define *market situations* as combinations of oligopoly competitors and an initial market situations. Thus, to estimate the skill of a competitor its performance in all market situations in the MG must be analyzed.

As the modified basic *assumptions,* we accept that

- the behavior of a manager j can be approximated by a *probabilistic program*
- the skill of a manager j can be evaluated by some numeric value R (j), or a *rating.*

In probabilistic models the results of testing of a manager by a market situation in the MG are random events and can be described by a random variable.

Let us introduce a function b(t) which for a pair (i,j) of competitors with difference in ratings of t points will be equal to the probability that i has a better performance than j in an arbitrary oligopoly competition where both of competitors are participating.

Note. We do not only prescribe some probabilities to the results of a competition of i and j but make also an *assumption* that the results are not essentially influenced by other competitors. *Actually, this assumption should be challenged by considering different forms of influence*.

E-assumption. The result of a competition for any two managers with a difference of t points in their ratings may be approximated by an exponential function a^t for some constant a. Thus, b(t) = P(i wins j) = m/ m+n = a^t /1+a^t .

To select the parameter *the* results of measurements of skills of management students in different years of study may be used with prescribing them some differences in ratings, for example in 200 units like the Elo's approach.

Let us evaluate the superiority of managers in a competition binary and prescribe 1 to the mangers having better performance than its opponent who is prescribed 0.

Considering a market situation T of MG as a tournament of Ci, C1… Ck managers, the expected value of Ci at that tournament T can be evaluated using the formula: Sum b(tj)/k, where tj is the difference of ratings of Ci and Cj, j = 1, …, m.

If the actual value of the testing of Ci by the same market situation T is VT then the new rating of Ci will be equal to Rn = Ro + w (VT - Sum b(tj)/k).

If testing of managers covers "enough" sets of market situations of the MG it is assumed that their modified ratings will converge asymptotically to their genuine values.

The following questions are relevant to the model.

4.1. How to select the market situations so that the ratings correction process converges?

How do the limit ratings depend on their initial values?

It is urgent to prove the convergence of that iterative process to a consistent system of ratings and to characterise the ordering. Particularly, it is important to find whether the system is consistent with the ideal ordering O*.

The proof may be similar to one used for the stochastic approximation processes. The coverage of the enough testors of the MG may be achieved by some random process of their selection.

Note, that iterative correction of the ratings might be ineffective because market competition is much more irregular than chess tournaments, particularly, caused by difficulties in a specification of the participants of each competition.

4.2. More realistic form of ratings determination may be by the Centers of Skill Assessment.
All market situations in the MG correspond to some tournaments with definite values of their ratings. The ordered sequences of testors define scales of tournaments and may be accepted as standards to assess management skills in a regular way.
Thus, to find the rating of a manager it would be enough to determine its position on that rating scale.
How does that scale match with the scale found by an iterative modification of the ratings using the formula above?
How does that scale relate to the scale defined in the deterministic model?
4.3. As we have mentioned the above E-assumption corresponds to the QTC in the deterministic model of the assessment.
The variance function #B(i,j) measures the deviation of the performance of the j program in the testors of the MG. The function b(i,j) in the rating system measures "deviation" of the program j with respect to the program in terms of the probability of the result of their competition.
Both of the measures are decreasing functions of the distance between the programs in the ideal ordering O*, i.e. are proportional to the difference of their "ratings".

Note, that in the probabilistic model the outcomes of testing by market situations are random events and to characterize them adequately one has to calculate the mean and the standard deviation.

OTHER MODELS FOR THE ASSESSMENT

1. ***Alternatives and testors***

1.1. We have presented the initial Management Skill Assessment (MSA) problem as a problem of an ideal ordering O*(K,M) of corresponding programs in accordance with their on- the-job performances. In the next step of analysis of the MSA problem we have refined the concept of the on-the-job performance by the tournaments in some real market game G

The results of such tournaments we had presented by (m, n) matrix – Matrix of Grades (MG) where m and n are the numbers of the analyzed competitors and all market situations, correspondingly.

To standardize the analysis of the MG and link it with other interpretations let us replace the concepts of “competitors” and “market situations” by more neutral ***alternatives*** A1... Am and ***testors*** T1, T2, …, Tn, correspondingly.

The MSA problem gives only one of possible interpretations of the alternatives and testors, and may be considered as a representative of the following more general and standalone class of the ***MG(K,M) problems*** :

given m alternatives, set of n testors to test alternatives individually by a criterion K and order them by a method M, find the best (K,M) alternative (or specify the ordering of the alternatives in respect of a defined ideal ordering).

There are interpretations of the MG(KM) in the sports, matrix games, voting and other multi criteria approaches to evaluation and assessment. Each particular MG(K,M) problem is specified by the pair (K,M) , the concept of an ideal ordering and the interpretation of the problem in whole. The reduction of the problems with different presentations and interpretations to the unified MG(KM) problem form will allow to use the known solutions and may become the base for a theory for that. Some links between matrix games and voting presentations of the MG(MK) problem will be outlined below.

1.2. The unified MG(KM) presentation allows to classify the testors to understand the difference between the problems. Here are some kinds of testors:

- *Completely reproducible* vs. not reproducible (technical measuring instruments vs. human experts)
- *Conditional* vs. not conditional, or predetermined sequences (dynamic testing by a dialog between an instructor and a student vs. testing by a sequence of independent questions).
- *Intentional* (ill-intentional) conditional vs. unintentional conditional (competitive games vs. games with nature)
- *Preserving original activity environment of the alternatives* vs. testing in the new environments which are a decomposition of the original one and which are believed are compensating the lack of the original environment (on-the job or real market game performance of competitors vs. measuring the characteristics of competitors like their knowledge in some areas, analytic and reasoning abilities, planing and communication skills, etc.)

In terms of that classification the management skill assessment by simulation games is reproducible and is processed in the frame of the original competitive environment - the game, the testing of the behavior of the competitors is intentionally conditional.

An assessment by standards like TOEFL and GRE tries to split the dynamic environment of the original language activity into some measurable components and then by integrating the results of separate measurements to evaluate the original activity. Unfortunately, current methods of splitting and integrating may suffer from subjective approaches and relate more to arts than to sciences.

In the more complex problem of revealing the constructs that are responsible for levels of competitive performances [Anderson,, Cannon,Malik.,Thavikulwat, 98] testors, as a rule, are not reproducible human experts acting conditionally/intentionally or unconditionally/ unintentionally.

Measurements are realized by compositions of criteria that are believed to adequately represent levels of management performances.

2. ***The maximum guaranteed grade criterion***

In two-person antagonistic matrix games [McKincey] both the alternatives and testors of the MG(KM) problem are interpreted as competing strategies and the grades are the sums of scores in the games.

To select the best strategy a popular max min criterion is used in the competition. If the saddle point exists for the method (a function) of a strategy assessment then min max choice coincides with the min max one. Although not all matrix games have the saddle point.

We found that the separate max min (but not min max!) criterion may be relevant for applying to the MSA problem in form of a MD. That criterion may be interpreted as *a maximum guaranteed grade* that the best strategy can preserve at the competition.

3. ***Comparing MG resolution with a game tree search.***

In a game tree searching [Nilsson, 79] the strategies are generated sequentially and the process of their selection is based on the comparisons of the terminal nodes.

The comparison is processed on the possible lower levels of the tree to reduce the search. For example, the technique of the branch and bounds is used for the reduction of the search [Nilsson, 79].

For MG the tree search process may be interpreted as a testing of strategies by testors.

In MG all branches of all strategies are presented explicitly as outcomes of performances for testors and any manipulation with them is allowed, e.g. the search similar to the sequential min max.

The application of the tree search methods to the assessment may be approved if the class of analyzed strategies are generated as a game tree.

The following Pro and Contra arguments may be mentioned for using MG form of presentation to a strategy searching in games:

- in applications we need to analyze only restricted number of relevant strategies and testors (for example, in market competitions);
  the variety of strategies is determined not by the possible elementary actions but instead a specialized language is used which essentially eliminates the irrelevant strategies (compare with VW); thus a lot of important applications correspond to tractable MG
- MG allows to compute in parallel
  but
- when testors (i.e. strategies) correspond to the sequences of elementary actions of a game tree the number of testors/branches in the tree is too large and its representation as MG is not approved
- it is a problem to get the outcomes of testors *simultaneously* and present them as a matrix when there is too many of them

4. Voting approach to the assessment

4.1. In the voting interpretation of the MG(KM) problems the testors are voters that have to elect the best candidate from the alternatives. They prescribe to each candidate their quantitative preferences, for example, on the scale 1, 2, …, m. Then using some method M the ordering of all candidates is performed and the best candidate is elected.

The method M is refined as consistent either with Condorcete or Borda criteria [Moulin].

For the Condorcete method Mc we have:

an alternative A is the best in the MG if for any other alternative B the number of testors Ti where A performs better then B is more than those where A is worse than B, i. e.

$\#\{Ti \,/\, Ti(A)>Ti(B)\} > \#\{Ti \,/\, Ti(B)>Ti(A)\}$.

For the Borda method Mb the best alternative Ai is determined by the max of the sum of performances of the alternative according to all testors Ti, i.e. $\max(T1(Ai) + T2(Ai) + \ldots + TN(Ai))$.

To apply Mc and Mb methods to the assessment we can refine the criterion K as a Return on Investment (ROI): ROI= Net Benefits/Total Initial Investment,

Net Benefits = Total Benefits – Costs- Depreciation/ Useful Life of Product

It is assumed that ROI can take integer value in the range [0, ..., p].

If p = 1 we get 0/1 MG for which in [Danielian, Pogossian] has proven the following

Theorem. Coplend and max score methods are equal if and only if the matrix of grades consists of 0 and 1, where 1 and 0 mean success and failure on the market, correspondingly, and the Coplend method is a version of the Mc one [Moulin].

4.2. What we have gained from the voting methods for management skill assessment?

- A variety of new Condorcete like criteria and corresponding methods
- A variety of natural constraints like monotonicity, continuity, etc. [Danielian, Pogossian] which being applied to the strategy assessment problem are promising to reduce the complexity.
- Reduction of the variety of voting criteria to the max score voting criterion for an important case of the 0/1 matrixes.

FURTHER STUDYING OF THE ASSESSMENT

A further research in management skill assessment we see in the following ways:

- Development deterministic and Elo like probabilistic models to construct corresponding rating systems and Centers for Skill Assessment
- Comparative analysis of the orderings induced by variety of criteria and models.
- *Reduction of the high complexity of the MG(KM) problems* using, particularly, new constraints, parallel methods to resolve MG, local analysis of MG, for example by acceptable taxonomy of testors.
- Broadening the collection of (KM) criteria, natural requirements to the testors and alternatives to increase an efficiency of the methods. For example, studying an influence of a compression of the set of testors similar to transformations in the test theory (min and "sufficient" sets of testors), monotonicity criterion [Moulin] etc.

A special importance has the issues related to the po-tests - tests preserving orderings.

- Finding minimal po-tests.
- Estimation of the loss in the accuracy of the ordering of alternatives given degrees of a not completeness of the po-tests and an amount of available computational resources*?*

The following *hypothesizes* looks appropriate in finding po-tests:

- *AssumptionA1.* The result of the competition of a strategy with any bundle of competitors in a given market situation is determined by a relation of the strongest strategy in the bundle with the tested one.

  It looks possible that orderings of the alternatives relatively to testors for a fixed market situation and all bundles of competing strategies and ones for the same situation but only with single competing strategies will be isomorphic. Thus, the complete set of single strategies at a market situation may be a po-test for the alternatives in competitions with all possible bundles of strategies at the same market.

If the AssumptionA1 is true each market situation will determine corresponding ordering of the alternatives invariant to the number of competitors in bundles. Thus, the complete, or the ideal ordering of the alternatives is an integration of that orderings.

Thus, the problem of the ordering of managers may be reduced to construction of those orderings corresponding to their testing in market situations with single competitors followed by their integration. into one ordering.

- How to get that particular orderings and how to integrate them? Are there regular connections between orderings?

Looks possible that alternatives including appropriate knowledge about all tested market situations have stable positions in all orderings?

- An ordered set of testors may be generated by a consecutive increasing of the knowledge base in the strategies. Given market situation and based on the above Assumption A1 it is worth to try to generate an ideal ordering of alternatives by a composition of "increasing knowledge" for each of essentially different market situations.

MANAGEMENT STRATEGY PLANNING

1. In the following chapters we are going to formulate the optimal management strategy search problem, the approaches of its programming, as well as its ties with the assessment one

2. Strategy planning, in general, answers the following questions:

- Where are we now?
- Where do we want to go?
- How can we get there? [Boon, Kurtz]

In terms of information technologies, we interpret the questions as follows:

- Recognize your current situation, i.e. identify and classify your current state and state of the environment using accessible language and its categories
- Identify your objectives using system of your values
- Formulate a search space and operationalize ways for efficient achievement of the objectives of the situation

The first two steps are aimed to identify objectives and narrow a space of a search of a target strategy, the last one must make the search effective and efficient.

Let us make a constructive the concept of strategy planning.

3. A company C which is characterized particularly by system of its values, list of possible actions and any kind of available resources and means: world perception, research and its

description, information, human, organizational, production, etc., is going to trade in the market.

Company C *perceives* its environment as a *market situation* S, *an initial* or *a current*, where it distinguishes available to its perception *any trade related elements*, particularly, the market and market place description, industries, competitors, their objectives and possible actions, general economic characteristics, etc.

It recognizes competitive companies C1, C2, …, Cm competing with respect to some criteria K1, K2, …, Km, for example, for a maximal value of the cumulative profit.

Let us emphasize that all processes are considered relatively, i.e. with respect to the company C.

Given criteria F we say that *strategy G achieves the goal F* if criteria F is true for the set of terminal nodes of G.

We name a strategy *G F projected* if are interested in whether the terminal nodes of G satisfy to criteria F or not.

Any description of an F projected strategy G aimed to make a search of G more efficient is named *a strategy plan for F* (F-SP).

An extreme and not useful representatives of strategy plans are a description of a problem under a consideration and a complete description of a solution strategy itself.

A useful strategy plan is a knowledge about a solution of the problem able to cut not perspective areas for an optimal strategy search and usually described in categories of a high-level language.

Strategy planning is a process of a narrowing of the search space of a target strategy individualized by a knowledge of a planner, particularly by used language, system of values and methods of a search.

A strategy search by planning may be classified as a knowledge-based synthesis of programs.

Contemporary marketing theory supposes that each competitor in each period of a competition acts in accordance with some strategy plan. Given criterion F competitor constructs series of strategy plans and by using constrains of the plans is narrowing the search space for an efficient selection of a strategy to resolve an actual problem.

In general, a *Management Strategy Planning* for a company C refers to an identification of the initial situations S, the objectives K of the company in S to determine the best K projected strategy plan in S.

The plan becomes a guideline for an actual strategy selection by executive managers in the competitive environments.
A methodology of a strategy planning is described in details in many studies allowing to construct plans effectively by special planning tools. But the process of a transmission from a strategy plan to a corresponding actual strategy in competitive environments is formalized minimally. We are going to complete that known management planning methods for a marketing department level to bring them to a procedure of an optimal strategy search in competitive environments. Let us to formulate the problem.

THE OPTIMAL MANAGEMENT STRATEGY SEARCH PROBLEM

1. There are a few ***levels of a strategy planning***: *corporate, department, division and unit*, which are hierarchic and consistent and each following organization specializes the plan of its predecessor.

For example, t*he corporate plan* description provides recommendations on the company's mission, policy, goals, allocation, organization, control of the business and its advantages, selects alternative scenario of its activity (for growth or survival, intensive, integrative or diversified, etc.), the competitors specification and the investments portfolio [Cravens]. A description of the corporate plan constrains next levels strategy plans search space to make the search more effective and efficient. Actually, all components of a description of a corporate plan determine, explicitly or not, the space of desired strategies and a set of recommended actions to find them.

Resolution of the strategy planning problem is connected with complete analysis of all the above stages of planning.

Here **we consider *planning on a marketing department level*** which starts the process with *input the set of initial market situations* and a *corporate plan description*, i.e. solves a version of the Management Strategy Planning problem for a given S, K and.B.
Thus, we don't consider here a huge and mainly not formalized part of essential work in planning as well as a structure of knowledge used for description of situations and selection of objectives, for example, mission identification. These constrains allow to concentrate efforts on realistic steps of strategy planning – *strategy programming*, fairly distinguished in [Mintzberg].

2. We consider a market situation/scenario and a variety of possible oligopoly competitors competing for some criteria: profit, ROI, 01success, etc.

Given set of initial situations S, competition criterion K, an integrative performance, or an optimality, evaluation method M, and a class B of alternative strategies for one of competitors in the **Optimal Management Strategy Search (OMSS)** *problem it is required to find an optimal strategy in the B.*

The OMSS problem looks similar to a variety of models of the competitive oligopoly markets.

One of them, the tool *Value War* (VW), developed by David Reibstein and Mark Chussil [Chussil, Reibstein 94] has proved its high-level adequacy to the real competitions and is an effective instrument for strategy evaluation [Chussil, Reibstein 98]. The VW uses strategy search simulation for consultancy purposes to predict results of oligopoly competition. A range of typical marketing price and quality competition strategies is used there, as well as the ability for their manual modifications.

3. Let us outline a connection of the OMSS problem with the assessment one.

Similar to the skill assessment MSGA problem at the OMSS we consider the 01succes criterion K and refine an optimality evaluation by a Maximum of Scores (MS) method.

The MSGA problem is meaningful for any subset of the entire class of alternative strategies because an ordering of given alternatives is required.

For the OMSS problem only the entire class of alternatives can be a subject of the analysis because the best alternative in the class is searched.

We suppose that the class of possible alternative solutions is specified in some form, for example, listed in a table, like in the VW, enumerated by some function, etc.

We suppose also that the set of possible competitive markets that must be tested is large enough and the chances of markets' appearances are equal.

Thus, the OMSS problem is reduced to a version of a Matrix of Grades problem where the sets of alternatives are not necessary finite and are given by some specification.

The similarity between the search and assessment problems implies that the issues considered in the assessment problem can be addressed in the search problem. However, their differences require development of a new search methods.

4. It may be argued that OMSS is computationally intractable problem and to get acceptable one we must consider an approximation to it. For example, constrains on the available computational resources may be introduced, and in the corresponding problem the best approximation to the best strategy may be required [Pogossian,1983].

Let us outline methods prospective for the search for optimal management strategies

STRATEGY SEARCH METHODS

1. *The alternatives are listed in a table.* The problem and methods coincide with ones at the assessment using Matrix of Grades. The complexity of the calculations, in general, linearly depends from the size of the matrix.

2. *The alternatives are available by an enumerative procedure and the list of testing markets is available for each of alternatives individually.*

The sets of alternatives may be recursively generated followed by summing up of their scores on the markets, selecting the one with max sum, adding it to the new generated set of alternatives and then returning back to the first step.

The *selection may relay on the MG solution methods analyzed for the assessment problem.*

3. *The alternatives are available by an enumerative procedure and QTC is adequate for their ideal ordering.*

An m-tuples of alternatives may be recursively generated for some list of markets followed by competitions to select the winners in the situations. The method and corresponding theorems about convergence of the process to the optimal solution are described in [Pogossian,1983].

4. *The alternatives are produced evolutionary by genetic algorithms.*

Generation of the alternatives is going by crossover and mutation procedures followed by selection of the best alternatives by their performances in urgent markets. In that methods we try to select a program which is absorbing an experience of successful generations by changes in a structure of the program. An ineffective averaging is possible here.

5. *The alternatives are produced in a learning process.*

The basic structure of the alternatives is determined similar to the structure of managers' decision-making procedure, particularly, including there a planning and knowledge storing, learning or inductive inferring, etc. The alternatives are processed in the set of urgent markets to achieve a behavior of a desired effectivity. The solution has to learn to act in each of possible markets and be ready to apply gained in that way knowledge to real ones.

6. *The alternatives are produced by a combination of an evolution and learning methods.*

A transparent description of evolutionary and learning methods is given in [Dhar,Stein] and applications of evolutionary methods to the economics in [Tesfatsion].

29th September, 1999

7. *The alternatives are described by a game-tree.*

As it was defined in Chapter? a game tree is a performance tree produced by all possible competitive strategies in a initial market situation. Thus, it is determined by an initial market situation, the possible actions of the competitors in any market situation and the adequate transformations of the situations caused by the actions are available.

Note, that game tree presentation preserves an opportunity to choose the best in the entire variety of possible strategies, in contrast, for example to VW where a finite number (22) strategies are under the consideration.

Because each strategy in competition is presented in the game tree by corresponding performance tree, we can search the optimal strategy among possible performances in the game tree. Thus, the whole arsenal of strategy search methods in game-trees are relevant to apply, including min max criterion, using an evaluation function, branch and bound method, first in depth or in width search, etc. [Nilsson, 98].

We have argued that the set of values of terminal nodes of each strategy in a game tree corresponds to some row in the Matrix of Grades (MG). Thus, there exist a correspondence between strategy selection criteria in those both presentations.

Particularly, *the sequential min max method of strategy search in a game tree corresponds to the max min selection method at the MG.*

The proof is by an induction through the depth of a game tree.

[Min max criteria is defined for a top-down depth in first method. Breadth in first is another search method with another outcome].

If for each situation of the game tree we mark an optimal action of our company we will get *the solution tree.* Actually, that is a set of pairs of situations of the tree.

The solution tree may be compressed by gluing pairs that transform different situations into equal ones.

Each of such pair is a chunk of an optimal strategy. Subsets of that pairs determine solutions of the problem (may be not complete).

Production systems, etc.

An example for two competitors in an initial situation.

Analogy with the chess tree and explanations for chess. The game tree, strategies, a complexity of the chess tree. Max min, branch &bound, evaluation function, A* ??

Computer chess programs and Deep Blue victory over the world champion.

8. *The alternatives are expert knowledge based.*

That is a knowledge about the solution of a particular problem and may be presented in different forms. For example, in a form of a production system which describes the solution in game tree. It may have different degrees of completeness. Incomplete production systems usually are accomplished by methods of a game tree search.

In a approach used in [Botvinnik] a knowledge is a scheme of a strategy plan and recommendation that direct the search in a game tree.

9. We are going to develop a class of expert-based management strategy search methods. The theory and models of the management strategy planning will be used that are compared with ideas of a strategy search methods applied effectively in computer games. Particularly, the ideas of M. Botvinnik and its followers realized in computer chess programs.

As a management strategy planning model, the MARKSTRAT2 approach is used for its constructive presentation and many years testing in the business education.

MARKETING STRATEGY PLANNING: MARKSTRAT2's Version

1. *Where in the strategy planning and how a level of the skill of a manager, i.e. his theoretical background, experience, other knowledge, etc., may influence on the quality of his planning performance?*

This answer may also reveal important dimensions to improve both the strategy planning performance and managers skill in general.

To make the issue constructive we outline strategy planning method for MARKSTRAT2 (MS2) [Larreche ,Catignon] version only for a particular market situation.

2.In general, MS2 method is subjected to ***the main stages of strategy planning***:

- ***identification the strategy plan for the current situation*** in a system of values and accessible knowledge of the decision maker
- ***operationalization the ways*** for achieving the goals of the strategy plan efficiently

MS2m is aimed to resolve any given market situation *to get maximal long-term profit for the brands of the company*. At each step of a decision making MS2m selects actions using either previous strategy plan or a new generated one.

Given segmentation *of the market* MS2 forms a strategy either for ***brands actually produced*** or ***brands projected*** by a product line of the company.

The strategy formation ***for a produced brand***, in general, looks like *a consecutive process of the strategy plan selection and its operationalization.*

A step by step a classification of a situation is realized to apply corresponding strategy plan, if it is known for the class, or to search a new strategy plan, in the opposite case.

For any classified situation an initial description of the strategy plan is the basic goal of a marketing, i.e. an achievement of the maximal cumulative, or long term, profit.

The system of its subgoals is the next detailed level of that guideline and strategy plan description. These subgoals and influencing them factors are given by stated below functional dependency graph [135]:

C O M P E T I T I V E A C T I O N S

⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓

Production Product characteristics Promotion Pricing

Capacity Optimal Positioning of the Product/Segment

Optimal **Maximal**

Product availability **Purchase Intention**

Distribution [Product Attractiveness]

Sales force ⇓ ⇓

Max Segment market share

⇓

Max Sales ⇐ Max Segment volume

⇓

Min Cost ⇒ **MAX cumulative PROFIT more than P***

This graph in fact is a disclosure of the concept of marketing [Boon,Kurtz].

A classification of the brand is done in respect *to each segment* by analysis of its disposition in the Market Growth rate/ Market share matrix or equated to it Market Attractiveness/Relative brand position matrix. Each square of that matrix: Star, Dog, Cow and Question mark, provides corresponding recommendations for farther formation of the strategy plan.

To recognize appropriate matrix's square for a given Brand/Segment (Br/Seg), particularly, the analysis of the following characteristics is suggested.

For Market Attractiveness.

Market factors: Market/Seg size in units, its growth rate and primary demand elasticity to price, advertising elasticity, shopping habits in Market/Seg, demand elasticity to product features, forecasting of accuracy of Market/Seg size and Market shares. *Competition:* number of brands competing in Market/Seg, the size, Marketing mix and reactivity of competitors, positioning of brands, treat of new brands, extent of new barriers. *Financial:* contribution margins, experience effect.

For Br/Seg relative position.

Market factors: Market/Seg share based on nits sold/value, sales volume in units/value, growth rate of Market shares/of sales, brand price/advertising/sales force elasticity, influence of brand on Market/Seg size. *Competition:* competitive advantage, elasticity of competitive brands in price, advertising and sales force, cross-elasticity for each Marketing mix variable. *Finance:* brand margin, cumulative production vs. the competitive brands.

These *lists of characteristics as well as recommended actions for each classified situation are constituents of the Strategy Mix and are the subject of continued improvement* both ***by the theory of the field and personal experience.***

A formation of all components of the Strategy Mix, except distribution and capacity regulation, are directed by *a general **positioning** policy*. Because positioning is a creation of a desired brand perception in a consumer mind it may be achieved by changing of both: physical characteristics of the brand and their perception by a consumer. For that purpose, MS2 recommends to measure a tendency of dynamic changes of ideal and real brands' consumer perceptions followed by elimination of the gap between them in their predicted positions.

Let's note that another way of positioning based on farther classification of the situations may be used instead of that universal one, e.g. a pricing strategy, as well as other than positioning policies depending from products life cycle may become adequate (brand reinforcement, modification, etc. [Cravens].

In result of the above process the strategy plan for a Br/Seg takes a form of recommendations in changing of the characteristics or price of the brand, advertising expenditures and their direction, capacity of production, sales force disposition, etc.

*The **operationalization of the plan** is realized by selection of legal actions consistent with the plans.* The process of an iterative adjustment of the quantitative characteristics of those actions coincides with a feedback from resources available for their performance, e.g. the budget.

The recommendations gained from the all stages of the planning have narrowed the space of perspective strategies and now the process of a particular strategy selection is starting. The system of goals and their coordination are determined by a strategy mix and a selected policy of their coordination, e.g. a positioning policy.

The process of the strategy search for all the goals may be arranged in parallel.

For a transformation of that strategy plans and goals into description of actions followed by their execution the Botvinnik's method [Botvinnik] seems very perspective.

For projected brands the above strategy planning includes also an important iterative process of transformation of the initial idea of an intended brand into actually produced one. In real marketing that process is concurrent with segmentation of the market as well as a target segment or a niche selection.

ENHANCING STRATEGY PLANNING BY COMPETITION ANALYSIS

1. So far MS2 method for a strategy planning was described. But how it can be accomplished to a complete strategy formation method to apply to simulation of a competitive environment?

In the management strategy planning by the MS2 the following two stages are emphasized:

- *A situation identification using the Knowledge base KB1.*

1. The mission determination interactively between mission options and available resources. Let's continue for a long term and more than some amount profit mission.
2. The product selection
3. The segments determination
4. New or old brand have to be marketed? Let's continue for an old brand.
5. Accepting as a given mission the max long-term profit takes as a basic strategy plan the graph of a dependency of the profit from the basic factors of a marketing (see the graph at the MS2 chapter). As follows from the graph that a guideline for Distribution and Capacity strategies must be Max product availability and for other strategies – Max product attractiveness.

- *Given type of a situation the strategy plan selection using KB2.*

To detail and specify the above guidelines the Product Life Cycle have to be determined and corresponding strategy plan recommendations chosen [Cravens].

As it follows from the recommendations of a strategy planning [Porter, Cravens] there is need for the stages:

- *Competition analysis using probable competing strategies in KB3.* Given strategy plan a competition analysis have to be done to bring the plan to a realistic strategy. It includes the following considerations
- *Final strategy or move in accord with the strategy selection.*

2. How that stages are realized in the VW?

A situation identification and some fixed strategies prescription to the competitors are going by a user. The competition analysis is realized for that strategies and the list of possible own strategies. Final strategy is selected by performance of our possible strategies in that competitive environment.

All stages of management strategy planning are presented in the VW. Nevertheless, the competition analysis is realized in a straightforward manner without enough feedback in process of competition.

In the VW only a few strategies have a feedback from some characteristics of competition, like strategies "follow the leader" and "tit for best tat" ... They are too rigid, dependent only from the integrative behavior of the competitors. There is no recommended by marketing methods enough automatic feedback from parameters of current market situations which negatively may influence to an adequacy of the simulation.

Some leverages for improvement of the strategy search. are suggested below

MANAGEMENT STRATEGY IMPROVEMENT LEVERAGES

1.Based on the above analysis of the MS2 the following elements that may change a quality of management, particularly strategy plans formation, were selected:

- a structure of the marketing procedure used by an organization
- a language used by a manager, particularly, an amount of concepts and their adequacy to the original competitive environment, i.e. the depth and width of a language used for a classification of the situations, a research methods to measure them, etc.
- adequacy, depth of hierarchy and detailing of possible classes of situations at the dependency graphs (e.g. of the initial situations and the goals graphs) that managers can differentiate at any stage of their analysis, e.g. by the Market Growth/Share matrix
- effectiveness of the Br/Seg determination procedures.

2. For simulation purposes, particularly in the frame of VW, it is important also to recognize how intensive in the MS2 method is the feedback from the changes of a situation to a modification of the current plan?

We may extract at least the following ways of that:

- Dynamic classification of current situations following by appropriate actions in the process of a game.
- Dynamic minimization of the gap between ideal and real perception of the brand if positioning policy is used.

The above regulators create a class of programs with variety of feedback.

The QTC must be also relevant for them because they essentially depend from their knowledge bases. Note, that using the same stages of planning the strategies of different strengths may be developed depending from the contests of their KBs and the form of their activation.

TOWARDS MANAGEMENT STRATEGY PROGRAMMING

Let us outline some programming approaches to resolve the OMSS problem.
1.1. A search space for the OMSS problem is the set of possible alternative strategies which as it was mentioned above may be listed, given by a generator or a game tree.

The game tree provides the most detailed description of the set of alternatives.

An assumption that in equal situations competitors make equal moves, creates one-to-one correspondence between strategies and their production systems.

The production system of a strategy G we name the set of all pairs "recognizer of situations/action" in G.

It is evident that the production system of a strategy G is a kind of a strategy plan. We may generalize components of the productions - recognizers and actions, as well as to use different types of operators to organize their coordination to get different presentations of equal production systems. Thus, a system of recommendations – productions, collected on entire stages of planning in a natural form determines a strategy plan.

1.2. An indication of a best action for each situation of the tree induces *the Complete Subtree of its Solutions (CSS)* which may be in its original *max detailed* form either compressed, or *generalized* form.

Typically, the only *Partial Subtrees of Solutions (PSS)* are available. In its primitive level of a presentation they may be production systems as subsets of a detailed or generalized CSS with some attached inference or reasoning mechanisms.

The detailed PSS are sets of productions. More developed forms of the PSS presentations may be highly integrated production systems with coordinators of a search in the productions, be in the forms of programs in a programming language or of universal recommendations like "Rise Price and Quality (for all situations)" used in the VW.

The class of PSS belongs to the Expert Systems and can use corresponding methods.

1.3. An example of a language for a convenient presentation of production form knowledge in chess - the SDL ,Strategy Description Language, was described in [Karapetian]. The Object-Oriented Languages may become the most prospective for strategy knowledge description. In a natural way they describe recognizers, operations over them, attached actions and procedures as well as their modular changes and modifications.

2. ***Chess master planning Botvinnik's method to a strategy search***

To advance in the issue let us outline an effective and efficient Botvinnik's approach for a similar problem.

2.1. A unique description of a method of using of high-level expert knowledge and experience for *strategy search in a game tree* for a combinatorial game - chess, is given in [Botvinnik]. Here we only outline the method and think it is worth to study in detail to apply to the OMSS problem.

Botvinnik's method is based on strategy planning with extensive use of both standard master level and individual chess knowledge .He describes high skilled chess player approach for a strategy search in middle game positions either completely new or studied in the past where corresponding experience in different forms may be used to cut branches to reduce the search. *The idea of Botvinnik's method* is the following.

Along with initial chess goal - a mat, the main actually searched subgoals are extracted, i.e. winning of a material, or some pieces as well as a subgoals of positional advantages. They are differentiated for perspective and current purposes.

The system of goals may be imagined as a hierarchy where subordinated goals achievement provides conditions for superior ones. Complete system of goals, especially for positional advantage, in fact, includes the whole range of chess concepts. [We have made an attempt to extract them and found more than 300 such predicates in [Pogossian,1983]].

The system of those goals is the initial description of the strategy plan. The plan is consecutively developed so far by extracting supportive and deteriorate sequences of moves considered for achieving the goals on the "clean" board - trajectories, their interrelated systems - zones, and system of zones, named "mathematical mapping".

The system of zones is in fact strategy plan for a given position. It describes the most promising sequences of moves which must be tested primarily in the search tree to achieve a target goal. Testing for that restricted search tree is arranged by a standard minmax procedure During the search initial strategy plan is continuously modified and adapted to the current position to guide the search for an optimal move.

Let's note that strategy plan initially is described in the language of primary actions of one's of the competitors without taking into account distortions of the opponent's moves. Then such plans are corrected by them to get realistic ones by an executive part of the program.

3.1. The above chess master's planning for a middle game positions is based on standard knowledge that skilled player realizes to search the best move.

In the process of the zones construction, i.e. strategy planning, or a search in the frame of a plan it is supposed a comparison of the descriptions of positions in the tree with ones stored in the memory to recognize precedents to apply successful experience. The classification of the positions is realized for all stages of the game, i.e. for debut, middle/end games.

A chess master skill differentiation is understood by completeness of a both: the standard knowledge used and amount of successful experience stored. That model of a chess master skill is very close to the one if a management planning described.

3.Botvinnik's idea was tested only for a few difficult chess etudes and combinations. The program was realized by B. Stilman who is developing it now effectively for other applications [Stilman].

The ideas similar to Botvinnik's one successfully were developed by J.Pitrat [Pitrat,72]. and D.Wilkins [Wilkins, 82] to resolve a series of complex tactically sharp chess combinations. Wilkins's program is a unique experiment in a using great amount of strategy planning expert knowledge. The two major advantages of the Wilkins' program over the Pitrat's one are in using conditionals which allow specification of different plans for different replies of the opponent and using easy modifiable language for plans which gives a natural way for chess program learning. This approach was successfully developed for robotics' application.

1. In the Botvinnik's strategy search method the following stages are crucial:

1. A situation identification. Particularly, its classification as a beginning, middle game, endgame, sharp, etc.
2. Variety of strategy plans generation. They are trajectories of one side "pure" actions for possible winnings of material.
3. Development the plans into realistic strategies by competition analysis and selection the most beneficial among the realistic ones. The techniques of zones is used and the most prospective zone is chosing.
4. The game tree is used to analyze the competition.
5. A result of the analysis is an optimal move in the position.
6. For each situation the entire above analysis is not excluded although its essential reduction is possible

FURTHER STUDYING OF THE MANAGEMENT STRATEGY SEARCH

1.The OMSS problem was a product of an analysis of the management strategy planning, extraction of the department level strategy planning and its formal description.

Given cumulative score criterion and a class of alternatives the objective is to find a program in a class that is the best in the set of possible competitive markets.

We have a set of constraints that come from higher levels of the management problem which determine a set S1 of possible initial market situations. Then we apply all available recommendations to that initial situations and try to narrow them maximally to a set S2 to apply appropriate strategy plans and at last to form adequate strategies in positions of the S2.

Those recommendations are well systemized and classified in the strategy planning theory. For example, in form of tables Market Growth Rate/Market Share, Product Life Cycle/Strategy Mix, Relation of the Market/Product to the company interest. In essence, these tables of recommendations are consistent and mutually inferred.

Thus, there is a classification process where terminal branches are recommendations for a strategy planning. These recommendations and other knowledge are used to specify the initial market situations, the rules of their transformations, the legal actions or/and class of alternative strategies. As a result, we get a solution search space in form of a game tree or/and a production system to infer the solution.

Let's outline a method of a search of management strategy based on an application of the Botvinnik's game tree search method to the strategy planning by the MS2.

2. Corresponding to the Botvinnik's hierarchy of the goals we had in MS2 the one based on graph of a functional dependency of the cumulative profit from other Strategy Mix components.

The goal of the Max Availability of the product is regulated by the capacity of the production, a distribution network with feedback from the resources of the company. An achievement of that goal may be for the first level of an approximation supposed as a standalone process.

The rest of the objectives described as a Max Attractiveness of the Product depends from Pricing, Product Quality and Promotion.

The Promotion may be also separated as the function of a difference between forming desired consumers' image about the product and the really produced one with the feedback to the available resources.

To find the Price /Quality strategies we suggest the Botvinnik method advanced to use in an explicit way any experience or additional knowledge.

We suppose the following steps:

- The set of trajectories of the P/Q moves is generated to preserve a given level of a cumulative profit without the competition
- The set of zones are constructed around the trajectories to enrich them by the competitive actions and contractions. For that the recommendations of the theory of planning may be used to direct a search.
- The process of a selection of the best zone and trajectory is arranged The generation and selection processes are accomplished by references to the knowledge and experience bases to cut an amount of calculations and make them more effective.

How to involve that data bases in the search? Particularly, probated strategies from the both of the theory recommendations and personal experience.

What is the nature of that data bases and how they must be organized?

What is the structure of programs for the search?

CONCLUSION

Interrelated problems of establishing standards for management skill assessment and optimal management strategy search were studied.

We have started from the native original problems of management skill assessment and strategy planning in corresponding management related theories, constructed their formal models and positioned them in known mathematical and computer sciences theories and developing methods to resolve them.

This alive chain from origins to formal models preserves necessary feedback for an improvement of an adequacy of the models.

The open questions were formulated and most promising plans to resolve them outlined.

The focus of the management skill assessment further studying becomes in a construction of a hierarchies of knowledge-based strategies which induce tests preserving orderings and rating scales for Centers of Management Skill Assessment.

The focus in optimal management strategy search is in a construction of a class of strategies which is knowledge-based as well and is able to acquire a new knowledge and coincide it with an earlier gained one, to use knowledge of different levels of abstraction consistently, particularly to plan hierarchically, to keep appropriate feedback with changes in current situations.

Comments on 29th September, 1999

The alternatives are described by a game-tree.

As it was defined in Chapter? a game tree is a performance tree produced by all possible competitive strategies in an initial market situation. Thus, it is determined by an initial market situation, the possible actions of the competitors in any market situation and the adequate transformations of the situations caused by the actions are available.

Note, that game tree presentation preserves an opportunity to choose the best in the entire variety of possible strategies, in contrast, for example to VW where a finite number (22) strategies are under the consideration.

Because each strategy in competition is presented in the game tree by corresponding performance tree, we can search the optimal strategy among possible performances in the game tree. Thus, the whole arsenal of strategy search methods in game-trees are relevant to apply, including min max criterion, using an evaluation function, branch and bound method, first in depth or in width search, etc. [Nilsson, 98].

We have argued that the set of values of terminal nodes of each strategy in a game tree corresponds to some row in the Matrix of Grades (MG). Thus, there exist a correspondence between strategy selection criteria in those both presentations.

Particularly, *the sequential min max method of strategy search in a game tree corresponds to the max min selection method at the MG.*

The proof is by an induction through the depth of a game tree.

[Min max criteria is defined for a top-down depth in first method. Breadth in first is another search method with another outcome].

If for each situation of the game tree we mark an optimal action of our company we will get *the solution tree.* Actually, that is a set of pairs of situations of the tree.

The solution tree may be compressed by gluing pairs that transform different situations into equal ones.

Each of such pair is a chunk of an optimal strategy. Subsets of that pairs determine solutions of the problem (may be not complete).

Production systems, etc.

An example for two competitors in an initial situation.

Analogy with the chess tree and explanations for chess. The game tree, strategies, a complexity of the chess tree. Max min, branch &bound, evaluation function, A* ??

Computer chess programs and Deep Blue victory over the world champion.

The generalizations.

- A general game tree for market competitions.
- Given operators/commands/rules of transformation of a programming language, a criterion of an optimal performance of a program and a description of input data, ideally, we can generate a performance tree of all possible programs for that data and search the best program by their performances in the tree.

The operators in the performance tree may be considered as inference rules in a calculus and the performance tree corresponds to the inference tree in that calculus. Sure, that is the most detailed inference tree with exhaustive search there.

2. Comments on 27th of November 1999

We have general Expert planning-based method for a class of competition problems. For problems of the class we have some specialized ones. All they are expert dependent and give a solution – ES.

In chess it is Chess master planning methods (Botvinnik, Wilkins, Pitrat).

In management it is Management planning methods.

In contrast to chess where we use game tree for competition analysis in management it is more adequate to formulate rules of competition resistance and reactions and test them dynamically like in VW.

REFERENCES

[1] Adelman L. Evaluating Decision Support and Expert Systems. A Wiley Interscience Publishing, 1992.

[2] Anderson P., Cannon H, Malik D., Thavikulwat P. Games Instruments of Assessment: a Framework for Evaluation. Proceedings of ABSEL-98, Maui, Honolulu, 31-37 (Jan.1998).

[3] Berliner H.J., Goetsch G., Campbell M. and S, Ebelinng C. Measuring the Performance Potential of Chess Programs. AI, 1990, v. 43, #1.

[4] Boone L., Kurtz D. Contemporary Marketing, 7th edition, the Driden Press, 1993.

[5] Botvinnik M.M. On Approximate Resolution of Search Problems. “Soviet Padio”, Moscow, .1979, p.150

[6] Chussil M., Reibstein D. Strategy Analysis with Value War. The SciPress, 1994.

[7] Chussil M., Reibstein D. Putting the Lessons before the Test. In Wharton on to Analyse & Develop Competitive Strategies, John Wesley&Sons, 343-368, 1997

[8] Cravens D.W. Strategic Marketing. IRWIN, 1994.

[9] Danielian E., Pogossian E. Voting Model for the Assessment. Proceedings of the 2nd International Conference in Computer Science and Information Technologies, Yerevan, 1999

[10] Elo A.E. The Rating of Chess players, Past and Present. London, Batsford, 1978.

[11] Karapetian B. Production Semantic of a Language for Strategy Specification. In Proc. of Workshop "Learning in intellectual systems", Yerevan, Oct., 1988, 57-65.

[12] Larreche J., Catignon H.MARKSTRAT2. Inter. Thomson Publ. Company, 1990.

[13] McKincey Inroduction to the Game Theory, !952.

[14] Mintzberg H. The Fall and Rise of Strategic Planning. Harvard Business Review, Jan.-Feb. 1994, 107-114

[15] Mitri M., Cannon H., Expert System for Developing Marketing Entry Strategy and Plans, Proceedings of the ABSEL-98, Maui, Honolulu, January1998.

[16] Moon J. Topics on Tournaments. NY, Holt, Rinehevart &Winston, 1968.

[17] Moulin H. Axioms of Cooperative Decision Making. ISBN-0-521-36055-2, Virginia Polytechnic & State University, 1988

[18] Myers G.J. The Art of Software Testing. NY, Wilsy-Interscience, 1979.

[19] Nilsson N. Principles of Artificial Intelligence. 1979.

[20] Pitrat J. A chess combination program which uses plans.AI, v.8,1972,275-371.

[21] Pogossian E. The Adaptation of Combinatorial Algorithms. (A book in Russian). The Academy of Sciences of Armenia Publ., Yerevan,1983, pp.290.

[22] Pogossian E. Ability of Efficient Evaluation of Knowledge Based Management Strategies. Proceedings of the 24th annual conference of the Association for Business Simulation and Experiential Learning, (ABSEL-97), New Orleans, March 1997.

[23] Pogossian E. Development of Management Skill Assessment, Proceedings of the ABSEL-98, Maui, Honolulu, January1998.

[24] Pogossian E. Increasing Efficiency of Management Skill Assessment.Proceedings of the ABSEL-99, Philadelphia, March 1999.

[25] Porter M.

[26] Rose W.

[27] Sadovski L.E., Sadovski A.L. Mathematics and Sport. Moscow, Fiz-Mat Literature, 1985.

[28] Tesfatsion L. A Trade Network Game with Endogenous Partners Selection. Economics Report series, No16, 1995.
[29] Thavikulwat P. Real Markets in Computerised Management Gaming Simulations Designed for Assessment. Simulation &Gaming, v.28, No.3,1997, 276-285
[30] Wilkins D.E. Using knowledge to control tree searching. AI, v.18,1982,1-51.
[31] Yakhnis, V., Stilman, B. Foundations of Linguistic Geometry. Proc. Of Symposium on Linguistic Geometry and Semantic Control, Puerto Rico, Feb. 1995.

2.Danielyan E., Pogossian E. **On Expanding of Method of Local Measurement to Evaluation of Management Strategies**, Proceedings of SEUA, 2009.

Methodology for the management strategies evaluation and ordering is proposed. It is based on strategy on-the-job performances and application of analog of the method of Local Tournament known as the Quasi-Transitivity constraint for management strategies assessment problem. A proof of the theorem describing this methodology is given.

Keywords: optimal strategy, strategies assessment, on-the-job performance, ranking

In competitive environment it is required to act in accordance with optimal strategy. Corresponding simulation models suppose that the solutions deliver recommendations on how to interpret the real world and how to act in it. One of the models for the optimal strategy selection is realized as a version of on-the-job performance assessment ideology based on the following basic assumptions [1]:

1. the strategies are ordered on the absolute scale by their on-the-job integrated competition performances in all oligopoly competitions with all other competing strategies;
2. any two strategies may be ordered by the absolute scale using local computational resources;
3. there are adequate game models to simulate the competitions.

As it was shown in [8] an ordering of strategies based on their on-the-job performances in SSGT-class problems [7] is a computationally hard problem and the question of the appropriate constraints becomes essential in reducing that complexity.

Management Optimal Strategy Provision problem

In *Management Optimal Strategy Provision* (MOSP) problem a company is competing in oligopoly market according to some success criteria (max cumulative profit, max return on investment, etc.) and is going to make decisions in market situations that are consistent with the best strategy at least for defined periods of the competition [8]. MOSP is a representative of SSRGT-class problem.

Focus on measurement of the effectiveness and on-the-job performance of management strategies in the MOSP problems allows to control the adequacy of measures by utilities of corresponding solutions [6].

The on-the-job performance is a measure that shows “how” the solution performs its “job” for individual task of MOSP problems. Management strategies can be ordered in accordance with their on-the-job performance in the game simulation model.

Maximal Sum Method

Let’s suppose that there is an ideal ordering of management strategies $O^*(K^*,M^*)$. In MOSP problem strategies $P_1, P_2, \ldots, P_m$ compete on oligopoly market and it is required to find criteria K and assessment method M for competition such that an ordering $O(K,M)$ of $P_1, P_2, \ldots, P_m$ will be isomorphically imbedded into the "ideal" ordering $O^*(K^*,M^*)$.

Integrative performance of management strategies in oligopoly competitions can be evaluated by method know as **Maximal Sum** (MS) that is similar to round-robin tournaments [3], where the best competitor is found by maximal sum of performances according to specified criterion K in all competitive oligopoly markets. Several initial assumptions should be done to advance in the MOSP problem solution by using MS method. They are described in [2, 8].

MS evaluation method required to consider for any competitor all its possible games against all possible samples of other competitors in all possible initial situations. For oligopoly competitions, ideally, all possible combinations of competitors which are in the oligopoly should be considered and then they can be ordered in accordance with their performances. The results of such tournaments are presented by mxn-size *Matrix of grades*, where m and n are the numbers of analyzed strategies and all competitive market situations, correspondingly.

Based on the matrix of grades and MS method the competitors can be ordered and obtain an ordering consistent with the ideal ordering $O^*(K,MS)$.

However, even for moderate values of the above parameters the evaluation of strategies by tournaments is computationally hard problem. Thus, to realize the idea in practice some additional constrains should be accepted.

Transitivity Constraint

To reduce the analysis of matrix of grades the **Transitivity Constraint (TC)** can be applied [8].

TC: P_i and P_j are competitors and B_i and B_j are sets of samples of competitors losing to P_i and P_j, correspondingly. So, P_i is stronger than P_j with respect to the ordering $O^*(K,MS)$ if and only if B_i includes B_j.

TC is based on the assumption that all competitors are essentially different in their skills. To order P_i and P_j in the frame of the Transitivity Constraint it is enough to find a sample of competitors that lose to P_i and win P_j. Unfortunately, the TC assumption is too specific and strong for practical use.

Quasi-Transitivity Constraint

More general constraint than the TC is **Quasi-Transitivity Constraint (QTC)** [8]. It can be used when competitors are close to each other by their strengths or even belong to the same class of equivalence in the $O^*(K, MS)$.

If competitor i is stronger than competitor j in the $O^*(K,MS)$ let's denote by $B(i,j)$ samples of competitive strategies that win i and lose to j. In other words, $B(i,j)$ is the set of samples that in games with strategies i and j achieve results opposite to the fact that i is stronger than j in the $O^*(K,MS)$ ordering. Then $\#B(i,j)$ is the variance function for the fixed i and varying j, and it is the number of elements in $B(i,j)$.

If $\#B(i,j)$ for some competitors is not zero the Transitivity Constraint can be applied only with some errors. The relevance of the Transitivity Constraint is increasing if the distance between compared competitors i and j is increasing.

QTC: Given strategies P_i and P_j, samples of strategies B_i and B_j losing to P_i and P_j, correspondingly, and the variance function $\#B(i,j)$. It is possible to select constants a and b (small enough compared with the number of all strategies in the ordering $O^*(K,MS)$) such that:

- if j belongs to the segment $[i+a, i-a]$ then $\#B(i,j)<b+1$;
- if j does not belong to the segment $[i+a, i-a]$ then $\#B(i,j)=0$ and P_i is stronger than P_j with respect to the ordering $O^*(K,MS)$ if and only if B_i includes B_j.

QTC is an expansion of method of Local Tournaments [5] that initially was developed for optimal strategies assessment in chess. Later it was proved applicable to the MOSP problem as well. The method was proposed as the main algorithm for the strategy generation and ranking process for the MOSP problem solution [2].

The QTC assumption allows formulate the following sufficient criterion for efficient assessment of management strategies

Theorem: Assuming Quasi-Transitivity Constraint for ordering $O^*(K,MS)$. Competitor f is stronger than competitor g (i.e. the location of f is better than g in the ordering $O^*(K,MS)$) if b samples of competitors can be found such that f wins and g loses games against each of them [8].

Let's represent the QTC in the following way:

there are P_i and P_j and sets $\{B_i\}$ and $\{B_j\}$, where $P_i \succ x$ for $\forall x \in \{B_i\}$ and $P_j \succ x$ for $\forall x \in \{B_j\}$ and $\exists$ a small enough that

- if $j \in [i + a, i - a] \Rightarrow \#B(i, j) > 0$ for no more than b points
- if $j \notin [i + a, i - a] \Rightarrow \#B(i, j) = 0$ & $(P_i > P_j \Leftrightarrow B_j \subseteq B_i)$

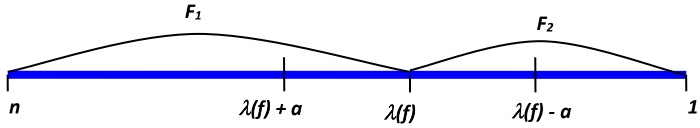
So, we have QTC for $O^*(01MS)$, the set $\{F\}$ and $f \& g \in F$.

It is required to prove that if $\exists\, t_{k=1,\dots,b}$ such, that $f \succ t_k \succ g$, then $f > g$. (Note, sign $>$ means "to be stronger than", sign $\succ$ means "to win to", and $\prec$ - "to lose").

Let's $\lambda(p)$ is the location of competitor p in the ideal ordering $O^*(01MS)$.

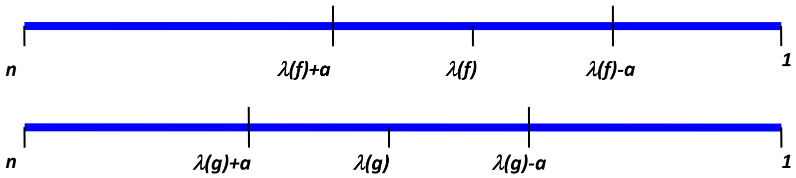
<u>Proof:</u> Let's assume that $f<g$. Then $g \in F_1$, where $F_1=[n,\lambda(f)]=[n,\lambda(f)+a] \cup [\lambda(f)+a,\lambda(f)]$ (Pic.1)

1. if $g \in [n,\lambda(f)+a] \Rightarrow g \notin [\lambda(f)+a,\lambda(f)-a] \Rightarrow \#B(\lambda(f),\lambda(g))=0$ (according to TQC) => $\nexists t$, that $g \prec t \prec f$. But there are $t_{k=1,\dots,b}$ that $f \succ t_k \succ g$ => this brings to contradiction $\Rightarrow g \notin [n,\lambda(f)+a]$.



Pic.1

2. if $g \in [\lambda(f)+a, \lambda(f)] \Rightarrow g \in [\lambda(f)+a, \lambda(f)-a])]$ (Pic.2)



Pic.2

Let $a = max\{a_g, a_f\}$, where a_g and a_f are small enough numbers for g and f correspondingly, so that QTC takes place.

If $g>f$, then according to QTC $g \in [\lambda(f)+a, \lambda(f)]$ and $f \in [\lambda(g), 1]$.

If $f \in [\lambda(g), 1] \Rightarrow f \in [\lambda(g), \lambda(g)-a] \cup [\lambda(g)- a, 1]$

If $f \in [\lambda(g)- a, 1]$ $\nexists t$ => so, that $f \succ t_k \succ g$. Thus, we come to contradiction again.

So $f \in [\lambda(g), \lambda(g)-a]$.

Hence, we have $g \in [\lambda(f)+a, \lambda(f)]$ and $f \in [\lambda(g), \lambda(g)-a]$.

Let's denote $F_3=[\lambda(g), \lambda(f)]$.

$g \in [\lambda(f)+a, \lambda(f)] \Rightarrow \#B(g, f) > 0$ no more than in b points $\Rightarrow t_k \in [\lambda(g), \lambda(f)]$

$\Rightarrow b \le |F_3| = |\lambda(g) - \lambda(f)| \le a \Rightarrow b \le a$.

But in case when $b > a$ it brings to contradiction.

Thus, if we take $b = a + 1$ and there are $t_{k=1,\dots,b}$ competitors such, that $f \succ t_k \succ g \Rightarrow f > g$.

QED.

This theorem formulates sufficient conditions to order any two strategies if we can indicate b competitive oligopoly markets such that the performance of one of the strategies is better than the performance of the other in all of these markets situations. Having the competitor f, the question of its strength relative to the competitor g is reduced to construction of a special tournament and estimation of parameter b. Even without an estimate of b with increasing the number of testing samples the likelihood of f to be stronger than g increases too.

Conclusion

Sufficient conditions and the model for ordering any two strategies in MOSP problems is formulated and proved. Based on criterion for efficient assessment of management strategies the solution of MOSP problem can be found with available computational resources. In the deterministic model it is supposed that any strategy is represented by some programs and the ordering of those programs satisfies to defined quasi-transitivity constraint. The QTC is an analog of the criterion of "essentially improvable" strategies formulated in [4] and allowed to advance in the efficient evaluation of strategies by their on-the-job performance simulation.

References

[1] Baghdasaryan T., Danielyan E, Pogossian E. Testing Oligopoly Strategy Plans by Their On the Job Performance Simulation. //Proceedings of the International Conference CSIT2005, Yerevan, Armenia, 2005.

[2] Danielyan E. Implementation of Local Tournaments Methodology for Evaluation of Management Strategies. // Proceedings of the International Conference CSIT2009, Yerevan, Armenia, 2009.

[3] Moon J. W. Topics on Tournaments. //N.Y. etc,: Holt, Rinehart and Winston, 1968.

[4] Pogossian E. Ability of Efficient Evaluation of Knowledge Based Management Strategies. //Proceedings of the 24th annual conference of the Association for Business Simulation and Experiential Learning, (ABSEL-97), New Orleans, March, 1997.

[5] Pogossian E. Adaptation of Combinatorial Algorithms. //National Academy of Sciences, Yerevan, 1983 (in Russian).

[6] Pogossian E. Business Measurements by On-The-Job Competition Scale. //International Conference “Management of Small Business: Problems, Teaching, Future”, Sevastopol, Ukraine, 2004.

[7] Pogossian E. Combinatorial Game Models for Security Systems. //NATO ARW on "Security and Embedded Systems", Porto Rio, Patras, Greece, August, 2005.

[8] Pogossian E. Models for Management Strategy Search and Assessment. //Proceedings of the International Conference CSIT1999, Yerevan, Armenia, 1999.

3. Pogossian E., **On Performance Measures of Functions of Human Mind**

ABSTRACT

We suggest a methodology for measuring computer’s performance of Functions of Human Mind (FHM). We start with an analysis of FHM and describe our proposal for improving prospects of their simulation by computers. Then, step by step, we specify measures that quantify computer’s understanding of FHM, followed by a development of customized measures for consistent knowledge-based problem solving. The steps include, categorization of FHM induced by classes of human activities, refining the criteria of understanding FHM, achieving human-computer consistency of the measures. Focusing on the problem solving for games simulated adequately by computers we demonstrate viability of the methodology for chess, a typical representative of that class.

Keywords

Measures, simulation, mind, strategies, chess, ratings.

1. INTRODUCTION

1.1. *Focusing FHM.* We cannot but state Being of Human Communities as the starting point of any reference in understanding the Being on the whole. Focusing understanding of Functions of Human Mind (FHM) in Being the main question to be answered is whether FHM can be reproduced by computers relying only on the common characteristics of the matter and energy.

Positive answer to the question can provide with essential arguments on FHM performance independency of their Human implementation and allow to concentrate research efforts on the question of origin of FHM, particularly from the matter and energy, as well as on the “damned” question of origin of the matter and energy themselves.

Because any scientific study rests on appropriate measures specific to the studied realities and pursued goals we need to refine the criteria of understanding FHM, associate them with definite dimensions of computer performances and chose corresponding measures.

1.2.1. *Criteria of Understanding.* We study FHM in frame of communicable universe of realities, i.e. in frame of totality of realities that people can identify, communicate about and operate with.

Community or exteriorized level understanding of realities can be interpreted, particularly, as the following:

- new realities have to be interpreted by or inferred from realities that are perceived at the moment as understandable
- consequences of understanding the realities can be tested by operations over those realities
- the realities can be adequately simulated.

We rely on the last criterion of understanding and require that programs simulating FHM would have performance comparable with the Human one.

1.2.2. *Patterns of FHM.* To compare Human-computer performances of FHM we need formal representation for views and models of FHM that at present are used in Human community.

We refine views on particular FHM by structure of *problems* and compositions of FHM by *systems of problems*.

The problems *deductively* or *inductively* have to describe the functions by determining the types of input and output structures, the target correspondences between them as well as criteria of quality of programs with the requirement to find the best programs realizing the correspondences.

Programs that realize the target function of a problem P and meet its optimality criteria we name *P-solutions.*

Any system S of problems representing FHM we name *S-pattern of FHM* and corresponding system of their solutions – *S-solutions.*

Actually, S-patterns with their S-solutions are constructive models of FHM adequately, we believe, representing them. Because S-patterns are identified by variety of not necessary formally defined views and models, in principle, there are not proofs of adequacy of S-solutions to the original FHM. By similar reasons a completeness of any pattern leaves a room for questions. Nevertheless, any dominating at present views on FHM induce corresponding patterns that pretend to be complete. We name them *ad hoc patterns of FHM* and given views on FHM we try to construct the most complete ad hoc patterns of FHM

1.2.3. *Consistency of Human/computer scales.* Given S-pattern we *S-understand FHM* if construct S-solutions with performances comparable with S-solutions of experts. Because scales of Human/computer performances of FHM do not necessary coincide to compare the scales we need their reciprocal interpretations.

Those interpretations as well as the measures of performances are worth to discuss in the contexts of FHM focused for simulations.

1.3. *Refining S-understanding.* To specify patterns of FHM we address to known views of behavioral and utility-oriented sciences where identify the following categories of FHM: *Innate FHM*, including Adaptation as well as

Vocabulary and Being models formation functions,

Needs Formation, *Target Utilities identification* (TU), *TU achievement* either by *exchange* or by *production,* distinguishing *Problem Formation* (interiorized, exteriorized), *Problem Solving* and *TU production.*

To bring FHM to patterns we formulate the following *basic assumptions:*

AB1. FHM can be represented by procedures which are solutions of corresponding problems specified either deductively or inductively

AB2. FHM procedures are permanent as correspondences of definite types of input realities into definite output ones but can change the content of those correspondences

AB3. The content of *knowledge* includes the solutions of problems, the components of the solutions and any their descriptions.

The patterns of FHM we define using basic assumptions and the ones induced by constructive interpretation of known views on *genetic psychology* by J. Piaget , L.Vigodski ,V.Stern, E. Meltsoff, R. Kays, J. Mandler, C.Beal, P.Kohen [2, 7, 10, 11, 24]. Particularly, we get the following analogies of *assumptions of development of mind*:

AMD1. FHM as a kind of knowledge are shared between a Human and the Communities the Human belongs to.

AMD2. There are finite number of definite and ordered types of stages of development of a Human determined by quality and amount of knowledge that Human possesses. The first stage is determined by an immense innate knowledge tuned to the environment and each new stage recurrently includes reorganized previous ones enriched by new knowledge. The intensity of formation of stages, the highest stage and the time people achieve can be different.

AMD3. Human knowledge increases using a unique adaptation procedure (AP) which is the same for all steps of Human development including transition to higher stages from the lower ones, expansion of knowledge on the stages, etc.

AMD4. AP consist of the following constituents:

- acquisition/assimilation of corresponding community expertise
- learning/accommodation to new realities and regularities as well as
- inference/reorganization procedures based on the already possessed structures and procedures

AMD5. In any pattern of FHM along with AP it is worth to include the problems of formation of vocabulary and being.

The criterion of understanding of FHM in frame of those refinements can be interpreted as the following:

given S-pattern and expert knowledge E we *SE-understand FHM if construct S-solutions consistent with assumptions AMD1-5 and having performances comparable with S-solutions of experts in S.*

1.4. We start with patterns focused on *Problem Solving* (PS) because they have long story of study, notable achievements in behavioral and computer sciences and can serve as a root to induce patterns of Problem Formation, Needs Formation, TU identification, etc. by developing the requirements to PS. In general, PS function have to follow the idea of universal problem solver, which provides solutions for arbitrary input problems. Historically, problem solving study, particularly in AI, started from a naive idea of simulation of PS by the General Problem Solver (GPS) [Simon] with a dream to outperform any Human in solution of any problem. In reality, construction of GPS is subject to doubt and actual studying of problem solving is concentrated on the classes of problems. The idea is in covering all problems by spacious and important for applications classes of problems and find corresponding solvers for each class which altogether have to approximate GPS.

Following to that rationale we study PS for class of SSGT games (*Space of Solutions simulated by Game Trees)* [18, 21] and aim to construct programs that meet the requirements of Assumptions 1-7 and for each problem of the class are able to find the solutions comparable with solutions of experts for the same problems.

Ideally, the designed programs have to be dependent on the contents of Human vocabulary, operate with those contents for approaching to the quality of Human solutions of problems, including understanding of the contents, their regular acquisition, learning, inference or combined changes, as well as to be able to represent the being of the programs and use it to improve the quality of performance of programs

In [18] for *Personalized Planning and Integrated Testing* chess programs we demonstrate an ability to achieve, in principle, a Human level of performance by using the contents of Human vocabulary. To progress in simulation of adaptation to those contents the paper reports our study in constructing Human/computer consistent measures of performance in games.

Sections 2 and 3 define the SSGT problems, refine the framework of study and measures of effectiveness of competing programs to induce corresponding requirements to them. Section 4 and 5 discuss consistency of human/computer measures of performance in chess and attainability of master solutions. The Conclusion underlines the main findings of the research.

2. SOLVING GAMES

We formulae PS for the SSGT class of game tree represented problems where *Space of Solutions* of the problems can be simulated by *Game Trees* [21].

We define the SSGT problems as a constructive subclass of a spacious class of Optimal Strategy Provision (OSP) problems [19, 22]. The OSP problems, first of all, meet the following requirements:

there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions in the (c) specified moments of time and (d) specified types of situations

there are identified benefits for each of the actors

there are descriptions of the situations the actors act in and transformed after actions.

For such problems with given arbitrary situation x and actor A, who is going to act in x, we can generate corresponding game tree GT(x, A) comprising all games started from x.

The games represent all possible sequences of legal actions of the players and situations that they can create from given initial, or root situation x. In our consideration the games are finite and end by one of goal situations of the problem.

In chess, for example, the actors are white and black players with checkmate as the goal, chess piece moves are (contra) actions and compositions of chess pieces on the board determine specific game situations (positions) transformed by actions according to chess rules in the corresponding game tree.

Assuming A plays according to a deterministic program, a strategy, the GT(x, A) represents, in fact, all possible performance trees of the strategies from the x. In that sense the GT(x, A) determines the space of all possible solutions from the x situation.

Note, that performance trees of the strategies are described not on the level of detailed commands but by their compositions, i.e. the actions of the players.

Given criterion K to evaluate the quality of strategies we can define the best strategy S*(x, A) and corresponding best action of A from x.

In the OSP problem with criterion K of quality of strategies it is required to find the best action in respect to K for any given situation.

The OSP problems comprise chess-like combinatorial problems, security and competition problems, particularly, network intrusion protection and management in oligopoly competitions problems [18 - 25]. Many other games represented security problems such as Computer Terrorism Countermeasures, Disaster Forecast and Prevention, Information Security, etc., announced by the NATO (http://www.nato.int/ science/e/ newinitiative.htm) seemingly can be reduced to OSP problems, as well.

The SSGT class comprises the OSP problems where the OSP requirement to have descriptions of situations after transforming them actions is replaced by the following stronger requirement:

the situations the actors act in and transformed after actions can be adequately simulated.

Thus, for SSGT problems the game trees can be constructively simulated what allows to create a common theory and computer-based game tree analysis methodology to find optimal solutions for corresponding problems.

In [18 - 25] it was proved that chess and chess like combinatorial problems, intrusion protection and oligopoly competition are SSGT problems and common methodology can be effectively applied to find high quality solutions for them.

3. STRATEGY EVALUATION MEASURES

Concentration on the SSGT class of problems allows to use measures of program performances induced by problems under consideration which, in contrast with universal ones, allow to control adequacy of measures by utilities of corresponding solutions.

We focus on measures of the effectiveness vs. efficiency and On-the-Job (OJP) performance measures [24, 25] vs. Attributive Scoring of Preference (ASP) based ones [5].

ASP measures evaluate performance of systems by aggregating the values of different attributes defined for different aspects of the performance into a global quality indicator (called the global preference).

Despite the fact that ASP measures are widely used in education, personal evaluation, systems and organizations evaluation, etc., they have a principle drawback in being very dependent on human subjective intervention in the measurement process. Particularly, they suppose a human choice of the base general criteria of functionality of examined systems, their partition on the subsystems with appropriate attributes and scales to evaluate them, and, finally, choice of a method to integrate measurements of the subsystems into global estimate of the system performance.

The OJP measures assume to be functions of quality of solutions of tasks comprising those problems. In other words, they have to be functions of “how” the solutions perform their “job” for individual tasks of those problems.

We distinguish two types of OJP measures based either on question-answers or on tasks solution values.

The Turing Test (TT) [16] belongs to the first type and because it is not a formal procedure can cause uncertainties and drawbacks in its applications, like the following ones:

TT procedure is very dependent on subjective characteristics of the judges and cannot be regularly reproduced

TT in spite of measuring solutions of individual problems does not use their internal criteria and, instead, pretends to be a universal measure

TT is based on local questions –answers to the solutions of problems and needs a framework of their correct integration into global estimates of the quality

TT compares the “external” question-answers “shell” of the solutions and does not measure actions in the world caused by the solutions, or is not “grounded” [11, 14, 15].

The tasks solution values (TSV) criteria are long time used to measure human and program performance. Based on the analysis of performance trees TSV criteria are widely used for testing of programs [9].

In chess TSV criteria coincide with a widely applied tournament-based ones [6, 12]. The OJP criteria for measuring performance of chess programs, management and intrusion protection skills were developed in [18 - 25]. For further studying we chose TSV OJP measures of effectiveness of solutions.

Finally, we need human-computer consistent measures, allowing to measure human and program performances in problem solving at the same scale. Particularly, to measure at the same performance scale an ability to incremental growth of the quality of performance by acquisition of expert knowledge.

4. CONSISTENCY OF HUMAN / COMPUTER SCALES IN CHESS

At present, Elo’s rating system is in the use for chess players [6]. Therefore, computer performance measures consistent with Elo ratings are needed.

We argue that max Sum criteria of strategy performance can play that role.

Justifying computer strategy quality criteria acceptable for a Human. Let’s eliminate strategy evaluation criteria from criteria that definitely are not acceptable for a Human.

OJP criteria for program strategies can be interpreted as functions of values of terminal nodes of performance trees of strategies. Terminal values of strategies determine the results of games of given strategies against all possible competitors (opponents, etc.). That is why they can be represented by vectors where the positions of components of the vectors correspond to the names of particular competitors and the values of the components correspond to the results of appropriate competitions.

Based on the analysis of those vectors and terminal nodes of the game trees we distinct the following two types of criteria and corresponding optimal strategies:

- induced by *Min max* search of the best strategies in the game trees
- induced by *direct comparison of the values of components of the vectors of strategies*

It is clear that Min max criterion is a pure program strategy criterion and there is no idea on using it for evaluation of Human strategies.

In the Testing Matrix Analysis (TMA) methods the vectors of the terminal nodes of examined strategies comprise a matrix and the optimal strategies are determined by comparing those vectors by criteria that are functions of the results of the analysis of the values of those vectors components.

From the variety of TMA criteria [106] we chose the max Sum of values of components of the strategy vectors which by definition, is equal to the *robin, absolute tournament* among all possible strategies. The following considerations must frame our analysis:

- At present criteria of quality of performance of experts are determined, primarily, for particular problems. Chess is one of the most advanced representatives of them.
- Criteria used to evaluate Human performance can be essentially dependent on peculiarities of Human cognition of the world

Let's analyze Elo ratings to find conditions when chess players evaluation criteria which are at present in use can be reduced to the max Sum one.

Elo ratings for strategy evaluation. The Elo's rating system was developed for chess players and accepted by the World Chess Federation [6, 12]. It is one of the most advanced systems among known standards to evaluate a competitive performance activity.

It is supposed that initially the values of all ratings are equal. Then each rating is changing in the process of testing by tournaments between players. The changes are proportional to the differences between expected and actually gained results of tournaments. The process supposedly converges to the actual values of ratings of tested players.

Elo's ratings is a system of chess performance evaluation based on the following assumptions [6, 12]:

Eexist: the rating R, a numerical measure of the strength of chess players, exists and determines corresponding ordering, *ER-ordering*

Eexp.: the number of wins of a player *i* divided by the number of its loses to a player *j* which differs at *t* points in its ratings in the ideal orderings of chess players, i.e. R(i) – R(j) = t, may be approximated by an exponential function a^t (a is in a power t) with a constant a..

Eprob: probability that i defeats j, or P(i beats j) = m/ m+n = a^t /1+a^t = f(t)

Elo's ordering is equal to ordering of means of scores in Human robin tournaments.

Chess players in contrast with the programs do not restrict themselves by deterministic procedures. They can change their strategies at the same starting situations and different strategies can be chosen with different preferences.

Well known model of human strategy behavior in games is based on the probabilities of selection of particular strategies from a given set of them. Unfortunately, the lack of constructive mechanisms to determine the probabilities makes that approach difficult for applications.

Elo's measure all diversity of behaviors of chess players is compressing into rating coefficients which are enough stable to measure their performances. We interpret that stability as *existence of means* of performances of players as the following assumptions:

ATMexist: a Human at the same position can do different moves and play by different strategies. If a Human plays all possible absolute tournaments by its all possible strategies we can get a distribution of its scores in the tournaments and corresponding means for them. Those means – absolute tournaments means (ATM), can be a stable representation of the strengths of the players.

ATMord: the ATM induce *ATM-ordering* of the players, which coincides with Elo ratings-based ER-ordering

ATMlocal: Eexp assumption can be interpreted for the ATM scale as indication of local zones of variance around means of players scores where results of games of neighbors differ from expected ones determined by their ranks at the scale.

Elo ratings are determined by knowledge of chess players

What is the base of a statistical nature of the scores in the absolute tournaments?

In other words, why players in absolute tournaments can demonstrate performances where the means despite to all diversity of disturbing factors are adequate concepts to apply?

We believe the *knowledge factor in ratings* (KFR) assuming *that strength of decision making of players is measured by their ratings which, in depth, are determined by the amount and quality of integrated knowledge the players possess.*

In their local appearances those decisions may wave but there is a mean strongly correlated with amount and quality of the knowledge of the players.

For knowledge-based strategies the rating sample means (RSM) are acceptable approximations of the rating means.

If behavior of players is determined by knowledge, they possess the deviation of scores of that players seemingly have to be very narrow, i.e. sample means have to be very close to the means.

In other words, a variance coming from the diversity of possible strategies that players can produce in a position add a little to the variance caused by diversity of competitions of one of the strategies of the players against all variety of their opponents in the absolute tournament. In other words, the following *RSM assumption* can be formulated:

the measure of quality of indeterministic programs based on integration of their scores against all possible competitors with a variety games in given situations against each of competitor can be acceptably approximated by the same measure but only one game against each competitor is considered.

Thus, if we measure strategies of a player in given positions against strategies of all possible opponents the integrated measure of the performance of that player can be a good approximation to the measure. In that case we along with considering all strategies of the opponents are taking into account all probabilistically possible strategies that the player can generate in the analyzed positions. RSM assumption states that the diversity of situations and opponents in the deterministic framework of testing a Human can comprehensively cover the entire variety of tests caused by non-deterministic nature of the players. As a corollary of RSM assumption we can state that arbitrary robin tournament for chess players is an acceptable approximation for the Elo ratings-based ER-ordering.

Linking Elo and max Sum orderings.

Let's **CIS** names isomorphism of orderings in frame of some assumptions, i.e. *conditional* isomorphism, and AT[x] is ordering induced by absolute tournament between x competitors.

Summarizing the above assumptions, we can infer the following statements:

ER-ordering **CIS** *ATM-ordering* **CIS** *any AT[all players]* **CIS** *AT[all strategies].*

In [24] it was argued that *AT[all strategies]* **CIS**

Therefore, ER-ordering is conditionally isomorphic to max Sum-ordering and any strategy S' is stronger than strategy S" by ER-ordering *if and only if* S' is stronger than S" by max Sum one.

Thus, max Sum scale can be a common measure to compare the strengths of Human-computer FHM performances.

Note, that it is not correct to measure Elo ratings for programs because for them Elo assumptions, in general, do not take place.

Max Sum as a measure of knowledge acquisition

Following to the KFR assumption we directly link changes in ranks of chess players in ER-ordering with changes in their knowledge and skills. If players rise chess skills, they improve their locations in the ER-ordering. Particularly, that is true for hierarchies of common chess knowledge that comprise standard part of chess education.

The vise-versa statement can be accepted, too: players with higher location in ER-ordering possess more chess knowledge and skill than the lower located ones.

Thus, we can assume that ER-ordering is sensitive to knowledge acquisition by chess players and changes of the players ranks in the ordering is directly proportional to changes of their Elo ratings.

But what are the relationships between changes in locations of programs in the max Sum ordering and inclusion of hierarchies of common chess knowledge in their work?

We state the following:

rising the ranks of programs in the max Sum orderings, in contrast with chess players is not necessarily caused by possessing more chess knowledge and skill than the lower positioned programs.

But what about changes in the ranks of programs in the max Sum ordering caused *ceteris paribus* by changes in knowledge bases of the programs, particularly by inclusion in the decision making some hierarchies of common chess knowledge?

In this case the changes in the ranks of max Sum scale can measure the changes in quality of programs caused by knowledge acquisition.

Therefore, if we test the procedures of knowledge acquisition or learning and max Sum criterion indicates a regular increasing of performance of programs in accord with acquired knowledge, we can conclude about adaptive abilities of those programs in human ER scale as well.

5. ATTAINING EXPERTS' SOLUTIONS

Whether is it possible to construct chess programs that acquiring or learning Human knowledge in frame of acceptable computational resources can solve chess on the level of chess masters?

To advance in this question we need to specify chess knowledge as well as programs able to acquire it. A comprehensive specification of "knowledge" is on the stage of development, yet, and we rely only on its recognized types and state the following:

- any description of any solution of any problem is knowledge
- the units of chess vocabulary with associated contents represent chess knowledge.

In [18, 16] we try to classify units of chess vocabulary and provide procedural representations for their contents. Then knowledge-based chess programs – Perspective Plans Dynamic Testing (PPDT), were defined in a way to change performance of the programs by acquiring chess knowledge [18, 24, 26].

Our belief that PPDT programs are able to improve performance by acquiring Human knowledge is supported by known results of [3, 17] in effective usage of chess knowledge for decision making.

In [20, 22] Intermediate Goals at First (IGAF) programs of PPDT class were developed demonstrating a strong ability to improve performance by increasing goals and rules types of expert knowledge in networks intrusion protection.

Experiments with *Personalized Plans Integrated Testing* programs of the PPDT class [18] demonstrated strong dependency of the quality of performance on typical knowledge of chess players. Particularly, two knowledge intensive Reti and Naderashvily etudes [3] were successfully resolved by those programs.

If assume that PPDT class contains programs comparable with masters can we achieve them procedurally?

Because PPDT programs are knowledge based the analog of Elo assumptions can be correct for them (see Quasi Transitivity Constraint (QTC) assumption in [24, 25]). Then, in frame of QTC, we can apply methods to achieve target program, at least, in an adaptive way, iteratively, with computationally acceptable cost of each iteration. Those methods are induced by QTC based local tournaments to measure quality of chess programs [24].

Thus, chess master programs, in principle, can be achieved if PPDT class contains them.

6. CONCLUSION

6.1. To progress in understanding of FHM by simulation we aim to construct programs having performance of FHM comparable with a Human one and consistent with Human knowledge processing by acquisition, learning and inference.

We specify FHM by systems of problems with criteria induced by those problems and demonstrate viability of the

methodology for chess.

In paper we focus problem solving FHM where distinguish the following two consequent stages of the success:

- search of strategies consistent with chess expertise
- formation and understanding of chess expertise, i.e.

- acquisition of expertise with understanding of corresponding vocabulary
- learning precedents with forming or acquiring vocabulary
- inferring consequences with forming or acquiring vocabulary

6.2. We argue that max Sum can be a common measure for Human/computer strategy performances. In addition, we state that rising the ranks of programs in the max Sum orderings, in contrast with chess players, is not necessarily caused by possessing more chess knowledge and skill.

If changes of the ranks of programs in the max Sum ordering are caused *ceteris paribus* by knowledge acquired by programs from the hierarchies of chess knowledge (e.g. from handbooks) then those changes can be interpreted as caused by knowledge acquisition and can be compared with a human ones by the same max Sum criterion.

In frame of those constraints max Sum can be used as a common measure for players and programs to evaluate acquisition, learning or inference of chess knowledge.

6.3. On the consequent stages a transition from problem solving to studying problem formation is planned in the following two research directions:

- decreasing chess specification from complete one to the models of inborn FHM like Jean model in [2]
- accumulating chess specification from given models of inborn FHM to the complete one.

To represent the schemes of actions and cognitive structures which comprise the core of Human knowledge by J. Piaget we are going to try *image schemas language* [2] along with core means of the Java classes. We expect to extend the concept of classes to distinguish in the methods interiorized actions – the operations, from real world actions, to interpret the overloading and overwriting tools as a sort of assimilation mechanism and develop an approach to the accommodation as a kind of OO program synthesis.

The last may be based on the Java beans ideology combined with inductive, e.g. use case approach in UML.

It is worth to consider some assembling and disassembling operations over Java classes as well which might be able to integrate and disintegrate classes into objects and back with appropriate criteria of generalization.

6.4. The next approximation to understanding FHM could require programs that are able to achieve ad hoc patterns of FHM with acceptable performance based on the min starting (innate) knowledge.

On the consequent stages of FHN studying we plan to understand what is the min starting knowledge to preserve continuous development by adaptation.

An exiting question is whether the starting knowledge and adaptation procedure can be reproduced by programs using common characteristics of matter and energy and if not where from is their origin?

In parallel, we need models to explain Being of Communities as a superior function relatively to Being of their members.

As a constructive step to those ends at present, we develop adaptation procedures for regular transition from one level of knowledge to the other higher ones. The following research framework is currently in use, particularly in chess [17]:

- choosing a valuable pattern of FHM
- constructing programs acceptably simulating the pattern
- searching means to restore those programs starting from possibly min starting knowledge.

REFERENCES

[1] Adelman L., Riedel S.1997. Handbook for Evaluating Knowledge–Based Systems, Kluwer Acad. Publ.

[2] Beal C., Cohen P.LGIST: Learning Generalized Image Schemas for Transfer Thrust D Architecture, Report, USC Information Sciences Institute Marina del Rey, CA, October 1, 2006

[3] Botvinnik M. 1984. Computers in Chess: Solving Inexact Search Problems. Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag: NY.

[4] Dhar V, Stein R.1997. Seven Methods for Transforming Corporate Data into Business Intelligence, NJ, Prentice-Hall.

[5] Dujmovic J., 1996. A Method for Evaluation and Selection of Complex Hardware and Software Systems. The 22nd Inter.Conf. for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG 96 Proceedings, Vol. 1, pp. 368-378, 1996.

[6] Elo A.E. 1978. The Rating of Chess players, Past and Present. London, Batsford.

[7] Flavell J. 1962. The Develometal Psycology of Jean Piaget, D. Van Nostrand Company Inc., Princeton, New Jersey.

[8] Harnad S.2001. Grounding Symbols in the Analog: world with Neural Nets- a Hybrid Model. Psycoloque: 12(034) Symbolism Connectionism

[9] Kaner C., Falk J., Nguen Q.N. 1993. Testing Computer Software. Inter. Thomson Comp. Press.

[10] Mandler J. The Foundations of Mind: Origins of Conceptual Thought. Oxford Univ. Press, 2004.

[11] Roy D. 2005. Grounding Language in the World: Signs, Schemas, and Meaning Cognitive Machines Group, The Media Laboratory, MIT (http://www.media.mit.edu/cogmac/projects.html)

[12] Sadovski L.E., Sadovski A.L.1985. Mathematics and Sport. Moscow, Fiz-Mat Literature.

[13] Searle J. 1990.Is the brain's mind a computer program? Scientific American 262, pp26-31.

[14] Searle, J. 1980. Minds, brains, and programs. Behavioral and Brain Sciences 3 (3): 417-457

[15] Winograd T., Flores F. 1986. Understanding Computers and Cognition (A new foundation for design). Publishers, Chapter 2, pp. 11–59, Huntington, NY.

[16] Turing, A.M. 1950 Computing Machinery and Intelligence. Mind 49 433-460 [Reprinted in Minds and machines. A. Anderson (ed.), Engelwood Cliffs NJ: Prentice Hall, 1964.]

[17] Wilkins D.E.1980. Using Patterns and Plans in Chess. Artificial Inelligence, v.14, 165- 203.

[18] Pogossian E., Vahradyan V., Grigoryan A. On Competing Agents Consistent with Expert Knowledge. Lecture Notes in Computer Science, AIS-ADM-07: The Intern.Workshop on Autonomous Intelligent Systems - Agents and Data Mining, June6-7, 2007, St. Petersburg.

[19] Pogossian E.: Specifying Personalized Expertise. International Association for Development of the Information Society (IADIS): International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006), 8-10 Dec., Barcelona, Spain (2006) 151-159

[20] Pogossian E., Javadyan A., Ivanyan E.: Effective Discovery of Intrusion Protection Strategies. The Intern. Workshop on Agents and Data Mining, St. Petersburg, Russia, Lecture Notes in Computer Science, Vol. 3505 (2005) 263-274

[21] Pogossian E.: Combinatorial Game Models for Security Systems. NATO ARW on "Security and Embedded Systems", Porto Rio, Patras, Greece, Aug. (2005) 8-18

[22] Pogossian E. Djavadyan A.: A Game Model for Effective Counteraction Against Computer Attacks in Intrusion Detection Systems, NATO ASI, Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Manag-t", Armenia, August19-30 (2003) 823-851

[23] Pogossian E.: Focusing Management Strategy Provision Simulation. Proceedings of the CSIT2001, 3d Inter. Conf. in Comp. Sci. and Inf. Technologies, Yerevan (2001) 37-42
[24] Pogossian E.: Adaptation of Combinatorial Algorithms. Academy of Sciences of Armenia, Yerevan (1983) 1-.293 (in Russian)
[25] Pogossian E. Management Strategy Search and Assessment Programming. CSIT99, Yerevan, August 1999
[26] Pogossian E., Hambartsumyan M., Harutunyan Y.: A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations. National Academy of Sciences of Armenia, IPIA, (research reports 1974-1980) (in Russian) 1-55.

4.Pogossian E., **On Assessment of Performance of Systems by Combining On-the-Job and Expert Attributes Scales**, International Conference in Computer Sciences and Information Technologies, Yerevan, Armenia, 2015 pp. 331-335.

Abstract
We develop a methodology of assessment of performance of systems based on the following two methods:
- *On-the-Job Competition Scales (JCS)* method developed for measuring performance of competitive systems or their constituents by the original, used in practice criteria of their effectiveness and
- *Logic Scoring of Preference (LSP)* method where for evaluation of systems elementary criteria, expert attributes are used for assessment followed by their aggregation into a global quality indicator called global preferences.

The methodology aimed to combine the strengths of LSP and JCS approaches while weakening some of their shortages to be applied to quantitative assessment of performances of competitive, education and software systems.

1.The On-the Job Competition Scales

These JSC methods aimed to evaluate decision making systems (DMS) or their constituents in competitions. Given competition it allows to order the DMS by their *on-the-job performance*, or *absolute* scales, in accordance of comprehensive comparisons of performances of all competitors by the base criteria of success declared in the original definitions of the competitions.
For example, on-the-job performance in sport competitions, e.g. in chess, correspond to robin –round tournaments that induce *absolute scales* determined by sum of scores of participants gained in those comprehensive competitions.
Those scales may be explicit or implicit, hypothetical or in a real usage.

Well known explicit and real usage scales are the ones in chess and tennis. Implicit scales are used for personnel assessment, for grading scientist and any professional skill, organizations, and in general, any human based systems.

Those absolute scales possess *transitivity*, either *quasi-transitivity* characteristic doing possible by local tournaments for arbitrary groups of competitors to form their orderings consistent with, or isomoprphically embedded into the absolute scale.
The quasi-transitivity tournaments are widely used in chess where Elo based rating system is a worldwide accepted scale to measure chess masters performance in an absolute sense, i.e. invariant to the location of chess masters, a time period when they are playing and number of participants in the local tournaments.
A framework for DMS software evaluation on the base of combinatorial games like chess was developed in [Pogossian77, 80,83]. Further development of those ideas resulted in development of an absolute

scale for management skill assessment in oligopoly competitions understandable as a on- the-job performance of managers in all possible oligopoly competitions [Pogossian97,99,03]. State-of- the-art of that approach is presented in [185].
Adequate simulation of the DMS is an essential precondition for constructing the JCS scales. That simulation includes the models of competing environment and knowledge based (KB) game tree search procedures.
If alternative strategies are presented as a manageable table the best strategy may be found by the max min, max Sum, Condorcet' criteria.
Exhaustive search-based methods and approximations to them to find an acceptable strategy are based on the min max method with a variety of cutting search heuristics: evaluation function, branch and bound method, etc. Among them the Intermediate Goals at First (IGAF) and methods by [10 - 12], Common Planning and Dynamic Testing (CPDT) and PPIT (Personalized Planning and Integrated Testing) [13 - 15] are notable by their intensive inclusion of expert knowledge.

2. The Logic Scoring of Preference Scales

The LSP methods are widely used for assessment of performances of people or systems in education and industry, accumulate the experience and knowledge of experts in applications and there are no restrictions on the new areas of applications.
For example, evaluation of software requires for a given user in some applications to select a software system, constituents of software or tools for their development that are the best among their competing alternatives.
LSP is applied to arbitrary software while evaluation criteria are determined by the view of experts what, in general is worth, to measure there.
According to [8] the LSP method can be classified as one of methods for multiattribute and multiple-objective decisions based on cardinal ranking of alternatives under certainty and with prior articulation of preferences.
The LSP method is used to create complex criteria, which are multiattribute value functions for computing the global preference indicator for each of n competitive systems or alternatives. Let's present a condensed overview of the approach to Windows Environment evaluation.
First the classes of WE users are investigated and the type of users for which we want to develop the criterion function is selected. The next step is to develop a WE user model. Using a questionnaire developed in [2] the selected users specify value preferences for all main WE attributes. The attributes that are selected as relevant for WE evaluation are called *performance variable*s. They are used as inputs for the LSP criterion function. A statistical analysis of the user questionnaire answers is used to compute the best values of parameters of the LSP criterion function. Therefore, the LSP criterion function is calibrated to reflect the needs of the selected type of users. It includes *all* requirements that the evaluated WEs are expected to satisfy. For each competitive WE it is necessary to prepare the set of values of all performance variables (inputs for the LSP criterion function). For each competitive system the LSP criterion generates the overall performance indicator which is called the *global preference* and used for cardinal ranking of competitive systems. The global preference can be approximately interpreted as the percentage of satisfied requirements. Consequently, at the end of the evaluation process we generate a cardinal ranking of competitive systems using the scale from 0 to 100%.

3. Requirements to Combining LSP and JSC Scales

3.1. We aim to develop application areas context dependent assessment methodology complementing LSP by JSC and based on simulation, statistics and tournaments.

The methodology applied, particularly, to software assessment – the LSP+, could complement expert attribute based LSP scales with competitive performance measuring JSC ones and let us to evaluate performance of constituents of software systems (but, yet, not the tools for their development).
Focusing software assessment LSP+ have to combine the strengths of LSP and JCS on-the-job performance scales.
We consider LSP as the base for the LSP+ which have to meet the following additional requirements.

1. It is natural to expect that the best alternative software chosen by the LSP+ method can be equal , or , at least, comparable with ones chosen by original, used in practice quality criteria. Thus, the LSP+ have to clarify how the best alternative chosen by the LSP relates to one chosen by the original criteria of the performance of the system.
   The LSP+ have to minimize human subjective intervention in the measurement process. Particularly, for the LSP stages of selection the base criteria, say functionality, testability, etc., the constituents of the software to evaluate, the attributes to detail their characteristics and scales to quantify their values, and , finally, the selection of a method to integrate measurement of all constituents to get global estimates of the systems.
   Ideally, orderings of the set of competing alternative software by different experts using the LSP+ have to coincide.
   3.2. To meet the above requirements, we modernize LSP by JCS scales as follows:

1. A software evaluation problem for DMS have to be chosen in a competition and for alternatives the JCS scale constructed which meet the following requirements:
   - it uses the DMS success original criteria
   - human intrusion and corresponding subjective factors are minimal
   - JCS scale comparing the alternatives have to the result of a comprehensive comparison of all DMS in the competition.
2. The LSP and JCS scales are compared and the LSP method complemented by an iterative procedure to make the LSP scale consistent with the JCS one.
3. The tuning procedure will be adjusted by series of similar experiments for DMS in competitions to make it stable and reliable in constructing LSP scales consistent with the JCS ones
4. The results of experiments are generalized and the LSP method modernized comprising the new LSP+ method.

4. Constructing JCS Scales

4.1. Let M is a competitive environment, or a competition, say, chess, oligopoly, military, economic, others.
And let K is a given criterion of the success in the competition,
{C} – the set of DMS in that competition M, e.g. organizations, human, other DMS, C- some DMS from the {C},
{T}- a variety of versions of a constituent T of the DMS that is the target for a measurement of a quality, e.g. a promotion instrument
We assume (**A1**) that performance of DMS in M can be measured by comprehensive tournaments or other regular ways of comparisons thus provide an absolute ordering, or a scale A(M,K) of measuring the quality of DMS in M.
And we assume (**A2**) that for fixed DMS C and varied constituents T in C the subscale A(M,K,C,T) of the scale A(M,K) can be considered for measuring the quality of T in terms of criteria defined for the quality of performances of C.
Finally, for creating the JCS scale we accept the following assumptions:
A3. There are adequate game models to simulate competitions and their determiners including the competitive environments and DMS, with all variety of their constituents.
A4. The ordering of models of competitors is isomorphic to the ordering of real competitors.

A5. The assumptions 3 and 4 are sufficient to induce a scale of models of DMS in a model of some competitive environment which is isomorphic to the absolute scale of the original DMS in the original competitions and satisfies to the quasi-transitivity constraint.

Assumption A5 comprise A3 andA4 ones and let to measure performances by simulation.

Namely, using Assumption 3-5 the scale A(M,K) and its subscales for the original competitions will induce an isomorphic scales A(M,K)m and subscales A(M,K,C,T)m for the models of those competitions that can be run by simulation.

Thus, following A1-A5 the scales A(M,K,C,T)m will meet the quasi-transitivity constraints and thus will let to order arbitrary pairs of target constituents T1 and T2 by local measurements in the metrics of the absolute scales A(M,K,C,T).

4.2. We argue the assumptions as the following.

1. *To preserve an adequacy* of simulation of competitions we are going to relay, particularly, on the models of business competitions widely used in educational business games [Markstrat, Brandmaps, see www.towson.edu/~absel], models of organizations and DMS constructed by the specialized simulation tools like Proforma [] as well as KB DMS , particularly, RGT (Reproducible Game Trees) Solvers with CPDT or PPIT acceptable strategy search engines developed in [Pogossian01, 03].

Given variety of game models we are going to choose the best of them by comprehensive comparative experiments.

The first question to be answered is whether the game tree model contains all relevant for the competition strategies?

To answer to this question game tree strategies exhaustive search procedure will be compared with the performances of a representative sample of experts in the field.

The experiments will be organized by the scheme successfully approbated in constructing game tree models for dynamic counteractions against network intrusions [Pogossian, Djavadyan03].

The second question to be resolved is whether strategy search algorithms are able to preserve the quality of search comparable with the exhaustive one?

For that goal an experience gained in the [Pogossian, Djavadyan03] and [Pogossian, Bagdasaryan03] can be applied.

The third question relates to preserving diversity of the DMS models sufficient to create a scale of models with increasing strength in decision making.

Increasing universality of the organizations simulation tools like Proforma is a good premise for it. Another leverage to achieve it is the KB embedded in the DMS that allow to relay on human ability to improvement of performance by gaining new knowledge.

The assumption A4 can be valid if, particularly, KB DMS were considered as the models of competitors, able to store knowledge and improve their performances correspondingly with the amount and quality of gained knowledge [Pogossian 98].

Particularly, the organizational knowledge includes many measurable and communicable components that may be reproduced in their models along with human depended ones measurable, yet, only to some extent (*Moorman*).

It is worth to mention that DMS performance measurement issues are tightly interconnected with a fundamental issues of the nature of the knowledge and its processing.

5. Conclusion

We develop a methodology of assessment of performance of systems combining the strengths of methods measuring performance of competing systems or their constituents by original, used in practice criteria of effectiveness and ones evaluating systems by expert attributes followed by their aggregation into a global quality indicator called the global preferences.

LSP and JCS approaches while weakening some of their shortages and to be applied to quantitative assessment of performances of competitive, education and software systems.

A dimension of further studying of combining LSP with JCS is planned for two years period as the following:

The 1st year.

Chousing alternated software and competition-based application area that meets requirements of the LSP and JCS methods and constructing adequate computer models for them.

The 2d year.

Building LSP- and JCS- scales for the software alternatives and reveling conditions for improving the quality of evaluation by integrating the strengths of the scales.

The US team will contribute, particularly, with its experience to evaluate urgent software systems, the collections of such systems, knowledge and position in software evaluation research activity

The Armenia team will contribute, particularly, by its experience and knowledge in tournament-based metrics development and their application to the DMS in oligopoly competitions and software evaluation.

Each team has a unique experience in software evaluation what is a promising base for a new quality in the collaborative research.

References

[1] E. Pogossian (1977): Tournaments Approach in Comparing Effectiveness of Information Structures. Proceedings of the conference "Mathematical Models in Complex Systems Simulation", Kiev, pp.16-19(in Russian).

[2] E. Pogossian, "Adaptation of combinatorial algorithms", Academy of Sci. of Armenia, Yerevan, (in Russian), p. 293, 1983.

[3] E. Pogossian (1997): Ability of Efficient Evaluation of Knowledge Based Management Strategies. Proceedings of the ABSEL-97, New Orleans.

[4] E. Pogossian, "Increasing efficiency of management skill assessment", Proc. of the 26th annual conference of the Association for Business Simulation and Experiential Learning (ABSEL-99), Philadelphia, March 1999, pp. 144-148.

[5] T. Baghdasaryan, E. Danielyan, and E. Pogossian, "Testing oligopoly strategy plans by their on the job performance simulation", Proceedings of the International Conference CSIT2005, Yerevan, Armenia, 2005, pp.8.

[6] E. Danielyan, "Implementation of local tournaments methodology for evaluation of management strategies", Proceedings of the International Conference CSIT2009, Yerevan, Armenia, 2009.

[7] E. Pogossian "Ability of efficient evaluation of knowledge based management strategies", Proceedings of the 24th Annual Conference of the Association for Business Simulation and Experiential Learning, (ABSEL-97), New Orleans, March, 1997, pp. 93-97.

[8] E. Pogossian, "Business measurements by On-The-Job competition scale", International Conference "Management of Small Business: Problems, Teaching, Future", Sevastopol, Ukraine, 2004, pp.47-51.

[9] E. Danielyan and E. Pogossian, "On expanding of method of local measurement to evaluation of management strategies", Proceedings of SEUA, 2009, 3p.

[10] M.M. Botvinnik, "Computers in chess: solving inexact search problems", Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag, New York, 1984.

[11] D.E. Wilkins (1982) Using knowledge to control tree searching. AI, v.18, 1-51.

[12] E. Pogossian (2001) Focusing Management Strategy Provision Simulation, CSIT2001, Yerevan.

[13] E. Pogossian, A. Javadyan, and E. Ivanyan, "Effective discovery of intrusion protection strategies", The International Workshop on Agents and Data Mining, St. Petersburg, Russia, Lecture Notes in Computer Science, vol. 3505, 2005, pp. 263-274

[14] E. Pogossian, V. Vahradyan, and A. Grigoryan, “On competing agents consistent with expert Knowledge”, Lecture Notes in Computer Science, AIS-ADM-07: The Intern. Workshop on Autonomous Intelligent Systems - Agents and Data Mining, St. Petersburg, June 5 -7, 2007, 11p.
[15] Pogossian E. Effectiveness Enhancing Knowledge Based Strategies for RGT Class of Defense Problems NATO ASI 2011 Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems, Salamanca, Spain, September 19-30, 2011, pp.16.
[16] M. Mollaghasemi and J. Pet-Edwards, Making Multiple-Objective Decisions. IEEE Computer Society Press, 1997.
[17] J.J. Dujmovic, A Method for Evaluation and Selection of Complex Hardware and Software Systems. The 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG 96 Proceedings, Vol. 1, pp. 368-378, 1996.

5.Berberyan L. **Modular Tool for Regulating and Analyzing Activities in Chess**, 2016.

1. Introduction

This Specification includes both API and Usage Guides.
Software has been tested using Ad hoc Testing approach.
In this work an approach and computer software for regulating and analyzing different activities in chess is suggested. The article starts with analyzing some standard approaches for regulating and analyzing activities in chess, listing some their pros and cons, especially concerning chess engines comparison and other interactions with them. Further an improved approach to regulating and analyzing activities in chess is specified. Then the implementation of proposed approach, providing flexible mechanisms for different manipulations is described. Developed software usage tips are also provided.

2. Modules

Application has modular construction. Modules are separated from each other. At the same time some modules can call the others.

Every module should have its own API. API can be divided into several parts.

Every module has a number of components that implements API in different ways.

There are different approaches to regulating and analyzing activities in chess. The standard approach includes chess software, which provides following components:

a. Board component - shows the chess board to user in some predefined way, visualize chessmen positioning.

b. Engines component - allows to run some chess engine supporting one of supported protocols as player.

c. Game component - allows to run chess games between players, sometimes supports remote gaming.

d. Position Component - allows to load chess game states or to store them.

e. Game Logging Component - allows to log the whole game.

f. Analyzer Component - allows to analyze some game state or the whole game.

User can open board UI, edit chessmen positioning, select players, can use some engine from the available listing as players, setup and run a game, then analyze positioning using some available component or analyze the whole game.

One of the most popular software solutions that implements standard approach is Arena Chess GUI.

The mentioned mechanism of regulating and analyzing activities in chess:

a. Allows to work with only available components and their implementations.
b. Is not flexible enough.
c. Doesn't provide ability to build custom usage scenarios.

Main constraints with available software:

a. Limited set of supporting chess protocols.
b. Constraints for use cases.
c. Limited set of tools for handling, logging and game assessment.

Problems with overcoming constraints in existing software tools:

a. Not flexible architecture.
b. Closed source code for most stable solutions.
c. Paid license for many solutions.

An improved approach to regulating and analyzing activities in chess

We are developing approach and software that is providing extended possibilities to regulate and analyze activities in chess. In order to ensure them software is designed with modular construction. Modules are separated from each other. At the same time some modules can call the others.

Every module has its own API. API can be divided into several parts. Thus, we can modify modules or add new ones separately. API covers module functionality and represents the mechanism for using it.

Every module has a number of components that implements API in different ways. With this design style we can just add some components to extend the module functionality.

So, we can have a flexible tool, that can be modified on several levels both the backend structure and use cases.

Such an approach allows to design custom chess scenarios like assessment tasks or chess tutors.

A tool was developed to implement an improved approach with modular design [4].

Software has been tested using Ad hoc Testing approach. Testing was done in Ubuntu, Windows, Mac OS platforms.

2.1. Chess Engine Session Module

This module allows to run new chess engine session. Chess engines session can be started in different ways. Several ways are implemented in current version, the others can be added manually or in the next versions of product.

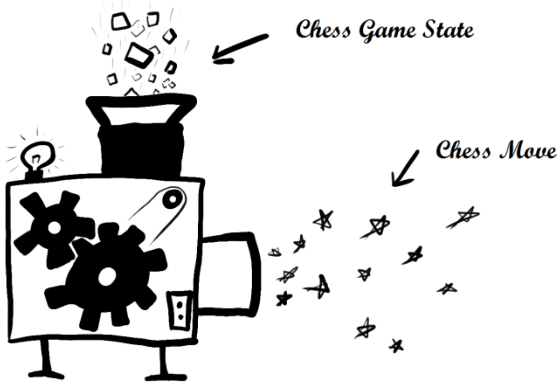


Figure 1. Chess Engine

Implemented ways of engines running:

- start native OS programs and connect to their input and output streams
- connect to web API through HTTP request with GET method

Also, Chess Engine Session Module has 2 parts:

1. Chess Engine
2. Running Chess Engines

Chess Engine Session Module API for part 1:

- createNewEngineSession(path) - create a new chess engine session by its path.

- writeLineToEngine(sessionId, theLine) - writes line(request) to engine with defined session id.
- readLineFromEngine(sessionId) - reads line(response) from engine with defined session id.
- destroyEngineSession(sessionId) - closes(quit) the session with defined session id.

Chess Engine Session Module API for part 2:

- getChessEngine(sessionId) - returns reference to one of running chess engines by its session id.

Components of part 1:

- ChessEngineAsProcess - implementation of running engine as native OS application. Process has its input and output stream (data stream) available for reading and writing.

createNewEngineSession(path) - create a new chess engine session running new process by the file located on specified path.
writeLineToEngine(sessionId, theLine) - writes string line to input stream of engine using session id.
readLineFromEngine(sessionId) - reads string line from output stream of engine using session id.
destroyEngineSession(sessionId) - destroys the process of engine using session id.

- ChessEngineAsWebGetResource - implementation of connecting to chess engine through Http.

url - url for requests, that can be set and modified.
response - string that contains response string received after request.
createNewEngineSession(path) - create a new chess engine session by its path, that is url.
writeLineToEngine(sessionId, theLine) - makes a HTTP GET request to specified url and writes theLine.
readLineFromEngine(sessionId) - reads response string received from engine after last request using session id.
destroyEngineSession(sessionId) - closes the engine session using specified session id.

Components of part 2:

- RunningChessEnginesHandler

getChessEngine(sessionId) - returns reference to one of running chess engines by its session id

//ChessEngineAsWebGetResource should be improved

2.2. Chess Engine Communication Module

This module allows to communicate with running chess engines. Chess engines communication can be done using different protocols different ways. In current version UCI [5] protocol is partly supported; the others can be added manually or in the next versions of product.

Chess Engine Communication Modules calls Chess Engine Session Module to refer to particular engine, add additional functionality.

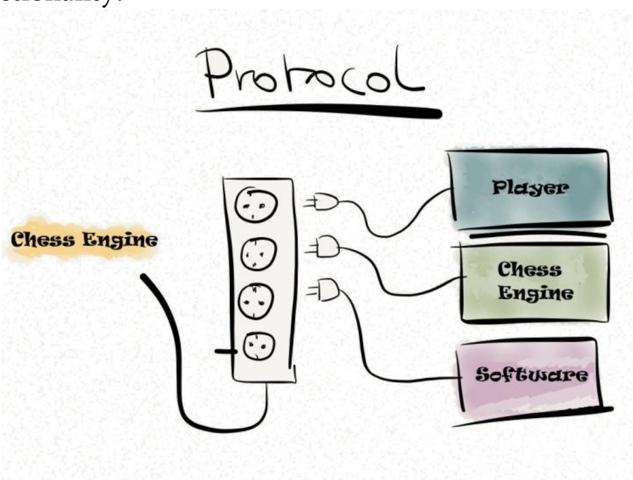


Figure 2. Chess Engine Communication Protocol

Chess Engine Communication Module API:

- setProtocol(sessionId) - sends to engine with specified session id command to set protocol as UCI.

- setNewGame(sessionId) - sends to engine with specified session command to start a new game.
- checkIsReady(sessionId) - sends to engine with specified session command to check if engine is ready for requests.
- getMoveCalculations(sessionId) - sends to engine with specified session id command to calculate move.
- setPositionFen(sessionId, fen) - sends to engine with specified session id command to set game state using board FEN positioning.
- quit(sessionId) - sends to engine with specified session id command to end game, this command also stops the related session.

Components:

- UCIChessEngineCommunicationHandler - handles commands that are based on UCI protocol.

//All UCI commands should be handled

//UCI responses should be handled in smarter way (check the response when it's full condition, not just in some milliseconds)

2.3. Chessmen Module

This module allows to represent chessmen, convert one representation form to another.

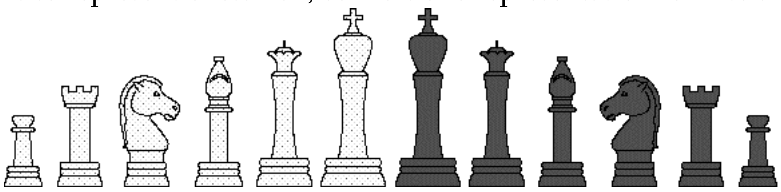


Figure 3. Chessmen

Chessmen can be represented in different ways. In current version 2 ways are supported, that are presented by 2 parts of module:

1. Abstract chessmen (example: King);
2. Symbolic chessmen representation (example: K).

Chessmen Module API for part 1:

- convertSymbolicChessmanToAbstract(theChessmanSymbolic) - converts representation of chessman from symbolic to abstract.

Chessmen Module API for part 2:

- getXFromChessmanSymbolic(letter) - gets the X coordinate from symbolic chessman first letter.
- getYFromChessmanSymbolic(theOrderLetter) - gets the Y coordinate from symbolic chessman second letter.
- convertAbstractChessmanToSymbolic(ChessmanAbstract theChessman) - converts chessman representation from abstract to symbolic.

Components of part 1:

ChessmanAbstractForm - allows to handle abstract chessman representation.

Components of part 2:

ChessmanSymbolicForm - allows to handle symbolic chessman representation.

//Possibility of manual chessmen creation should be added

2.4. Chess Board Module

This module allows to represent chess board. Chess Board can be represented in different ways: graphical and non-graphical. There are some GUI implementations like the concept in XBoard.

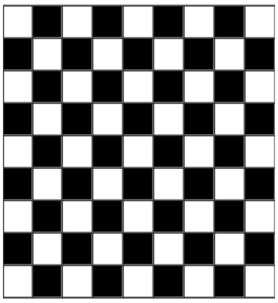


Figure 4. Chess Board

In current version several representation methods are supported:

1. Symbolic Array;
2. Swing Window.

Chess Board Module has 2 parts:

1. Chess Board
2. Active Chess Boards

Chess Board Module API for part 1:

- create() - creates new chess board and activates it.
- getBoardMatrix() - gets matrix of chessmen with their positioning on board.
- setBoardMatrix(theBoardMatrix) - sets the positioning of chessmen on board based on chessmen matrix.
- initialize() - set the board default state.
- getChessmanOnXY(y, x) - gets the chessman located on (y;x) coordinate.
- setChessmanOnXY(y, x, theChessman) - sets the chessman located on (y;x) coordinate to theChessman.
- show() - shows the board state.
- applySymbolicMoveToBoard(theSymbolicMove) - applies symbolic move to board.

Chess Board Module API for part 2:

- getActiveChessBoard(theActiveChessBoardId) - return active chess board y it's id.

//Possibility to represent chess board using JavaFX should be added

2.5. Chess Move Module

This module allows to define chess moves. Chess moves are entities related to chess board: Chess Move -> Chess Board.

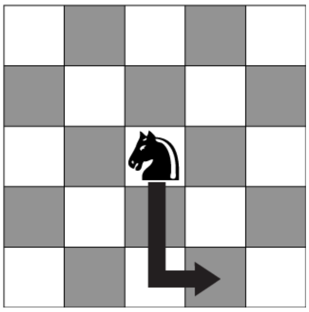


Figure 5. Chess Move Example

Chess Move Module has 2 parts:

1. Chess Move
2. Chess Move Validator

Chess Move Module API for part 1:

- uciMoveToSymbolicMove(theUciMove) - converts the move received using UCI to symbolic representation.
- symbolicMoveToAbstractMove(theSymbolicMove) - converts the symbolic representation of chess move to abstract one.

Chess Move Module API for part 2:

- isValid(theMove) - checks if the move is valid.
- checkEndOfGame(theChessGameState) - checks if end of game for specified chess game state.

2.6. Chess Game Module

This module allows to handle chess games. Chess Game is related to Chess Board, Chess Piece, Chess Move, Chess Player, Chess Communication. Chess game is a sequence of game states.

Figure 6. Chess Game

Also, Chess game can have different situations:

- simple attack
- move with no effect
- castling possibility
- etc.

Chess Game Module API:

- create() - creates new chess game
- initialize() - initializes new chess game.
- run() - runs chess game with specified options.
- checkEffect() - checks the effect on game like castling effect, etc.
- registerMove() - registers move for particular chess game.

2.7. Chess Game Logging Module

This module allows to handle chess games logging. Logging can be done for game state or for the whole game.

There are different approaches for description of chess board state. Currently software supports FEN [7]. The logging of whole game can be done with PGN [8].

Chess Game Logger Module API:

- convertToLoggingFormat() - converts chess data to logging format.
- convertFromLoggingFormat() - convert logging data to chess data.

3. Usage Guide

3.1. Basic Commands

Usage scenarios are based on calling modules API functions related to particular components. Calling modules API functions can be organized in any sequence what provides flexible use cases mechanism.

Basic commands for UI:

- modules - shows available modules.
- module [module name] - shows API for module.
- module [module name] -component [component name] -function [module API function name] -options [list of options values separated by coma]: calls the function from particular module API by component name that implements it with specified options values.

quit - quits the application.
scenarios - shows available usage scenarios.
scenario [scenario name] - runs particular scenario.
scenario create [scenario name] - creates new scenario with specified name.
scenario [scenario name] add [module name] [module API function] - adds the function from particular module API.

quit - quits the application.

3.2. Usage Scenarios

Usage scenario for the tool - is sequence of calls of functions from modules APIs.

Software working modes

Chess interaction handler software tool is designed to work in three main modes:

1. Handling chess games between two specific persons as users on current PC.
2. Handling chess games between two different chess engines compatible with chess communication protocol running on current PC;
3. Handling chess games between specific person as user and chess engine compatible with particular chess protocol running on current PC;
4. Handling chess educational process between specific person and chess engine supporting tutor mode compatible with particular chess protocol running on current PC;
5. Testing chess knowledge of specific person as user using chess engine compatible with particular chess tutoring protocol running on current PC;

Software modules

Chess interaction software tool design is divided into several parts (modules) according to functionality specifics.

1. User session handler;
2. Engine session handler;
3. Chess board presenter;
4. Chess communication protocol packer;
5. Chess game handler;
6. Game state logger;
7. Game logger;
8. Chess move validation controller;
9. Chess teaching protocol packer;
10. Chess game simple analyzer.

Use cases

Chess interaction tool allows to implement different use cases, that can be compiled into several general ones:

1. Player vs Player type use cases while:

*user is choosing Player vs Player game mode →
*user1 is initializing his session params →
*user2 is initializing his session params →
*user1 is initializing chess game params →
*the game is running →
*game results are show →
*user can request some game details.

2.* Engine vs Engine type use cases while user:

*is initializing sessions for two engines ->
*is initializing chess game ->
*is configuring chess game handler->
*runs the game->
*analyze the game.

3.* Engine vs Person type use cases while user:

*is initializing session for him->
*is initializing session for chess engine->
*is initializing chess game ->
*is configuring chess game handler->

*runs the game->
*analyze the game.
4.* Person & Engine type use cases while user:
*is initializing session for him->
*is initializing tutor params->
*runs the tutor->
*runs the tutor checker->
*analyzes the results.

Software Requirements

Platform:
*Windows OS
*Linux-type OS
*Mac OS
Environment:
*Java 1.8

Software features

Software has some specific features:
*Every module can be configured independently from the others
*Multi-platform environment support
*Flexible architecture (modules can be upgraded)

Usage Model

How to Run?
Execute program .jar file under Java Machine.
Startup.
On the startup you should choose mode of game.
Game modes.
There are 3 game modes:
1. Player vs Player
This mode represents game between two users on same computer.
To enter this mode user should insert “pp” into tool terminal.
2. Engine vs Engine
This mode represents game between two chess engines on same computer.
To enter this mode user should insert “ee” into tool terminal.
3. Player vs Engine
This mode represents game between user and chess engine on same computer.
To enter this mode user should insert “pe” into tool terminal.
1. ~
2. ~
3. Player vs Engine game mode has two interaction modes:
*playing mode
To enter this mode user should insert “play” into tool terminal.
*tutoring mode.
To enter this mode user should insert “tutor” into tool terminal.
Interaction modes.
3.1. ~
3.2. Tutoring mode allows users to learn interactively some game concepts or test one's knowledge.
There are two groups of tutoring commands: for tutoring chess concepts and for tutoring chess strategies.
Tutoring commands.

For tutoring chess concepts:
*getListOfTutorConcepts – prints a list of concepts available for tutoring.
*learnConcept[name] – start the process of tutoring a concept by name.
*testConceptKnowledge[name] – start the process of testing the knowledge of concept by name.
For tutoring chess strategies:
*getListOfTutorStrategies – prints a list of strategies available for tutoring.
*learnStrategy[name] – start the process of tutoring a strategy by name.
*testStrategyKnowledge[name] – start the process of testing the knowledge of strategy by name.
There is also “exit” command that allow to close program any time and “escape” command allowing to leave current program branch and to enter to one level up.

References

[1] Arena Chess GUI website. [Online]. Available: https://www.playwitharena.com
[2] Edward Pogossian, “On Assessment of Performance of Systems by Combining On-the-Job and Expert Attributes Scales”, CSIT-2015.
[3] Sedrak V. Grigoryan and Levon S. Berberyan, "Developing Interactive Personalized Tutors in Chess", Mathematical Problems of Computer Science, volume 44, 116-132, 2015.
[4] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design patterns, software engineering, object-oriented programming", Addison-Wesley, 1994
[5] R. Huber and S. Meyer-Kahlen, UCI (universal chess interface), CCC, November 28, 2000.
[6] XBoard website. [Online]. Available: https://www.gnu.org/software/xboard/
[7] Steven J. Edwards, "Forsyth-Edwards Notation", Portable Game Notation Specification and Implementation Guide, 03.12.1994.
[8] Steven J. Edwards, "Portable Game Notation", Portable Game Notation Specification and Implementation Guide, 03.12.1994.
[9] K. Khachatryan and S. Grigoryan, “Java programs for matching situations to the meanings of SSRGT games”, Proceedings of SEUA Annual conference, pp. 135-141 Yerevan, Armenia, 2013.

Appendix 3: OJP Management

1.Danielyan E., Pogossian E., "**An Application of Voting Methods to Strategy Assessment**", Proceedings of the Conference on Computer Science and Information Technologies (CSIT'99), Yerevan, Armenia, 1999

Классифицируется ряд известных методов голосования с позиций их эффективнного приложения к выбору наилучшей стратегии менеджмента. Сформулированы условия эквивалентности методов суммирования очков и Копленда.

1. Постановка задачи о наилучшей стратегии менеджмента. На рынке действуют несколько конкурирующих фирм. Поведение каждой фирмы описывается набором всевозможных стратегий, зависящих от состояния рынка и действий остальных фирм в дискретнные промежутки времени t. Требуется найти оптимальнную стратегию для одной из этих фирм, например, наиболее прибыльную.

Переформулируем задачу следующим образом.

Даны m альтернатив A1, Am (стратегий), n тестеров T1, Tn. и матрица M(mxn), где aij- результат применения тестера Tj к альтернативе Ai. Требуется найти оптимальную альтернативу согласно результатам тестирования [Pogossian99].

Очевидно, что в такой постановке проблема может быть интерпретирована как проблема голосования и интересна возможность использования его методов для задачи о стратегиях

2.1. Рассмотрим методы голосования, описанные в [Мулен91]. Предварительно опишем критерии-аксиомы теории голосования, осмысленные и для проблемы стратегий.

Оптимальность по Парето, анонимность и нейтральность – очевидные требования в задаче о стратегиях.

Монотонность: альтернатива остается выбранной, если ее относительное положенние на рынке, согласно некоторым тестерам улучшается, а относительное положение остальных фирм не изменяется.

Пополнение: две непересекающиеся группы тестеров N1 и N2 оценивают одно и то же множество альтернатив. Пусть согласно тестерам N1 и N2 избирается одна и та же стратегия. Тогда согласно тестерам N1UN2 избирается та же стратегия.

Участие: при рассмотрении нового тестера должна избираться либо та же стратегия, что и раньше, либо та, которая для нового тестера строго лучше предыдущей.

Непрерывность: Пусть согласно тестерам из N1 избирается стратегия Ap, а по непересекающейся с N1 малой группой тестеров N2 избирается стратегия Aк. Тогда существует достаточно большое число r дублей группы N1, такое что комбинированная группа тестеров (rN1) UN2 изберет стратегию Ap. Это свойство означает возможность усиления N1 дополнительными тестерами с целью устранения влияния малой группы N2.

Классифицируем методы голосования

2.2.1. Правило Борда. Наилучшей считается альтернатива, получившая максимальную сумму по очкам тестеров. Это правило оптимально по Парето, анонимно и нейтрально. Оно является также монотонным и удовлетворяет критериям пополнения, участия и непрерывности. Верхняя оценка сложнности равна mxmxn+mxn+m-1= mx(mxn+n+1)-1. Действительно: приписывание очков стратегии Ai согласно тестеру Tj относительно остальных стратегий - mxmxn действий,.суммирование очков по n тестерам - mxn действий, и нахождение стратегии с максимальной суммой очков - m-1 действий.

2.2.2. Правило Кондорсе. Проводится парное сравнение стратегий согласно результатам тестеров. Строится матрица mxm, результат кругового турнира между стратегиями A1, Am где aij = 0, если i=j или Ai хуже Aj согласно большинству тестеров, и aij =1, если Ai для большинства лучше Aj.

Стратегия Ap является наилучшей если j apj=maxij aij. Если maxijaij=m-1, тогда Ap является победителем по Кондорсе. Правило оптимально по Парето, анонимно и нейтрально, если существует победитель по Кондорсе. Оно также удовлетворяет критериям монотонности, пополнения, участия и непрерывности.

Верхняя оценка: m2+(m-1)x(nxm+2)/2. Действительно: для каждой пары стратегий производится n сравнений - nxCm2, подсчитывание суммы очков для каждой из m стратегий - mxm., и нахождение стратегии с максимальной суммой - m-1 дейсвий.

2.2.3. Правило голосования с подсчетом очков (*суммирование очков*). Фиксируется неубывающая последовательность действительных чисел
$s_0 \leq s_1 \leq \leq s_{m-1}$, при $s_0 < s_{m-1}$. Выборщики ранжируют кандидатов, причем s0 очков дается за последнее место, s1-за предпоследнее и т.д. Избирается кандидат с максимальной суммой очков.

Правило подсчета очков - анонимно, нейтрально, выполняется аксиома пополнения и непрерывность. Анонимность, нейтральность и оптимальность по Парето выполняются, если приписанные очки разные $s_k <$ s. Из теоремы 9.3 [124] следует, что оно также удовлетворяет аксиоме участия.

Правило Борда является примером правила с подсчетом очков, где $s_0=0$, $s_1=1$,
$s_{m-1}=(m-1)$.

На случай стратегий метод подсчета очков интерпретируется также как правило Борда и имеет ту же сложность.

Методы голосования, которые выбирают победителя по Кондорсе, если таковой существует, и определены произвольно в противном случае, называются <u>состоятельными по Кондорсе</u>. Правила Копленда и Симпсона являются таковыми. Это два наиболее естественных обобщения победителя по Кондорсе. Оба правила - оптимальны по Парето, анонимны и нейтральны. Аксиома пополнения не выполняется для этих методов, если не существует победителя по Кондорсе. При m>3 не выполняется и аксиома участия.

2.2.4. Правило Копленда. A_p является выигрывающей стратегией по Копленду, если $\Sigma_j a_{pj} = max_i \Sigma_j a_{ij}$,, где $a_{ij}= 0$, если i=j, или если $A_i=A_j$, согласно большинству тестеров, $a_{ij}=1$, если A_i побеждает A_j и $a_{ij}= -1$, если A_j побеждает A_i.

Правило Копленда - это прямое следствие правила Кондорсе и имеет ту же сложность. Его можно сформулировать и так: a - победитель по Копленду, если он побеждает в максимальном количестве парных сравнений по правилу большинства.

2.2.5. Правило Симпсона. Рассмотрим стратегию f, любую другую стратегию x и обозначим через $N(f,x)$ количество тестеров, у которых оценка f лучше оценки x. Оценкой Симпсона для f будет min $N(f,x)$, $\forall x \neq f$.. Выбирается та стратегия, у которой эта оценка максимальна. Верхняя оценка сложности: $mn(m-1)+m(m-2)+(m-1)=m^2n-mn+m^2-2m+m-1=m^2(n+1)-m(n+1)-1=$ m(m-1)(n+1)-1. Действительно: попарное сравнение по n тестерам - $nm(m-1)$ действий, нахождение min $N(f,x)$ для каждой из m стратегий- $m((m-1)-1)=m(m-2)$, и максимум по для всех min $min\ N(f,x)\ m-1$ действий.

Правила Копленда и Симпсона выбирают подмножество стратегий, которое может состоять из нескольких стратегий, получивших одинаковую оценку.

2.2.6. Метод относительного большинства с выбыванием. Этот метод относится к методам подсчета очков. Он исключает расточительные выборы. Хотя сложность этого метода и мала, но для стратегий он не подходит, т.к. метод описывает состояние фирм на рынке лишь по одному или двум параметрам, что затрудняет выбоп оптимальной стратегии.

2.2.7. Метод альтернативных голосов. Он не всегда является монотонным и использовать его для оценивания стратегий нецелесообразно.

2.2.8. Голосование с последовательным исключением. Этот метод удовлетворяет аксиоме состоятельности по Кондорсе. Но голосование при последовательном исключении не является нейтральным и может нарушаться оптимальность по Парето. Оно по существу несправедливо: альтернатива, представленная последней, имеет очевидное преимущество, поскольку ей надо выиграть только у одного своего оппонента. Это свойство неестественно и метод не будет рассматриваться для стратегий.

2.2.9. Метод параллельного исключения - состоятелен по Кондорсе и оптимален по Парето, когда при бинарных выборах нет равенств. Тем не менее если равенства возможны, то оптимальность по Парето может нарушаться. Этот метод также не подходит для задачи оценивания стратегий.

Возможно ли построить схему последовательного исключения так, чтобы при произвольном количестве альтернатив избирался оптимальный по Парето исход?

Такой метод голосования существует и это

2.2.10. Дерево многоэтапного исключения Для каждого конкретного упорядочения альтернатив существует по одному такому дереву [Мулен91].

Хотя дерево многоэтапного исключения велико (при p кандидатах получаются 2^{p-1} финальные вершины), его решение (т.е. вычисление выигрывающего кандидата) может быть получено с помощью очень простого алгоритма. (См. теорему Шепсл-Вейнгаста из Мулен91).

Теорема Шепсл-Вейнгаста вполне применима для стратегий.

Для стратегий в качестве T берем турнир, который проводится по результатам тестеров стратегий, и fTg означает, что в турнире стратегия f набрала больше очков, чем стратегия g. Метод многоэтапного исключения состоятелен по Кондорсе, оптимален по Парето, анонимен и монотоннен. Однако, как показал Мулен, не существует бинарного дерева, победитель по которому всегда является победителем Копленда (Симпсона).

При использовании многоэтапного метода исключений победитель находится быстрее, т.к. алгоритм (теорема Шепсл - Вейнгаста) в среднем требует сравнения не более половины от всех m(m-1)/2 пар.

3. **Метод Кендал-Вейа** позволяет ранжировать участников турнира [26]. Этот метод может использоваться для ранжирования кандидатов в задаче стратегиы: дана матрица предпочтений и турнир T по принципу большинства, соответствующий данной мартице. Метод Кендал-Вейа применяется к турниру T. Этот метод состоятелен по Кондорсе. Удовлетворяет критериям оптимальности по Парето, анонимности, монотонности.

В случае стратегий строится турнир T, основанный на парном сравнении стратегий по тестерам, вычисляются силовые вектора по Кендал-Вейю, что дает возможность упорядочить стратегии.

Данный метод может применяться, если правило Копленда (Симпсона) выбирает подмножество стратегий (победителей) с одинаковыми оценками.

4. Итак, все выбранные нами для оценивания стратегий методы являются оптимальными по Парето, монотонными, анонимными и нейтральными. Нейтральность нарушается лишь при методе многоэтапного исключения.

5. Сводная таблица соотношений методов и критериев-аксиом выглядит следующим образом:

	Парето	Анонимность	Нейтральность	Монотонность	*Пополнение*	Участие	Непрерывность	Верхние Оценки Сложности	Применимость к стратегиям?
1.	+	+	+	+	+	+	+	*mx(mxn+n+1)-1*	+
2.	+	+	+	+	+	+		*m^2+(m-1)x (nxm+2)/2*	+
3.	*	*	*	+	+	+	+	*mx(mxn+n+1)-1*	+
4.	+	+	+	+	**	***		*m^2+(m-1)x x(nxm+2)/2*	+
5.	+	+	+	+	**	***		*mx(m-1)x x(n+1)-1*	+
6.	+	+	+	-	+		+		-
7.	+	+	+	-	+		+		-
8.	****	+	-	*****					-
9.	****	+	-	*****					-
10.	+	+	-	+				*mx(m-1)/4*	+

* - условие выполняется, если в методе суммирования очков все очки разные, $s_k < s_{k+1}$; ** - не выполняется, если нет победителя по Кондорсе; *** - не выполняется, если *m>3*; **** - может нарушаться; ***** -не всегда выполняется.

Взимосвязь методов голосования приводится в таблице.

<table>
<tr><td colspan="3">Правило суммирования очков *(3)*</td><td colspan="4">Правило Кондорсе *(2)*</td></tr>
<tr><td rowspan="2">Метод относительного большинства с выбыванием *(6)*</td><td rowspan="2">Правило Борда *(1)*</td><td rowspan="2">Метод альтернативных голосов *(7)*</td><td colspan="3">Правило Копленда *(4)*</td><td rowspan="2">Правило Симпсона *(5)*</td></tr>
<tr><td>Голосование с последовательным исключением *(8)*</td><td>Правило параллельного исключения *(9)*</td><td>Дерево многоэтапного исключения *(10)*</td></tr>
</table>

6. Эквивалентность основных представителей методов голосования выявляет ледующая

Теорема. *Методы Копленда и суммирования очков эквивалентны, тогда и только тогда, когда матрица тестирования состоит из 0 и 1, где 1,0 интерпретируются как успех и неуспех стратегий на рынках.*

ЛИТЕРАТУРА

[1] Мулен Э. “Кооперативное принятие решений: аксиомы и модели” “Мир”, 1991.

[1] Moon J. W. “Topics on Tournaments” NY; Holt, Rinehevart & Winston, 1968.

[2] Pogossian E. M. “ Models for Management Strategy Search and Assessment.”, CSIT99.

2.Baghdasaryan T., Danielyan E, Pogossian E. **Testing Oligopoly Strategy Plans by Their on the Job Performance Simulation**. //Proceedings of the International Conference CSIT2005, Yerevan, Armenia, 2005.

Abstract

A game tree approach to dynamic testing and assessment of Strategic Plans is developed based on simulation of their on the job performance in oligopoly competitions.

The approach covers the issues of cutting strategy search in the tree, organizing the results of testing into a matrix and visualizing them.

The software package is presented that analyzing the matrix of tests for a given set of questioned strategies recommends the best strategy optimal by one of the Borda, Condorcet, Copeland or Simpson’s voting criteria.

Introduction

In the Management Strategy Provision (MSP) problem a company is competing in oligopoly markets for some success criteria (max cumulative profit, max return on investment, etc.) and is going to make decisions in market situations that are consistent with the best strategy at least for defined periods of the competition [Pogossian 2001].

To advance in solving the MSP problem by simulation an adequacy of the model it needs to be preserved along with constructing effective and efficient strategy search algorithms that will be able to acquire regular management knowledge for increasing quality of the search.

In our model of oligopoly competition a few companies compete for max profit. They form various Strategic Plans that describe in a qualitative manner changing of basic competition parameters - the Price and Quality of goods produced by them [Pogossian 2001].

An approach of selecting the best strategy by Matrix of Tests is realized as a version of the On the Job Performance assessment ideology based on the following three basic assumptions [Pogossian 2004]:

1. *ideally,* it is meaningful to construct on-the-job competition *absolute* scale of organizations ordering them by their integrated performances in all oligopoly competitions with all other competing organizations
2. arbitrary two organizations may be ordered by the absolute scale using local computational resources
3. there are *adequate* game models to simulate oligopoly competitions.

To find the most preferable strategy for our company all combinations of Strategic Plans of the companies are considered and each combination is tested via producing the most expected strategies for selected combinations of plans. Each combination of produced strategies is run in corresponding competitions (dynamic testing) and the results of testing are allocated into the Matrix of Tests for choosing the best strategy by one of criteria - maxmin, maxSum, Condorcet, etc. [Pogossian E., 99], [Danielyan, Pogossian 99], [Pogossian, Baghdasaryan 2003].

A large number of those combinations and their possible quantifications provide difficulties on working up and assessment.

We describe a method to reduce strategy search in the tree and allocate the results of simulation of quantified behaviors of Strategic Plans in different competitions into Matrix of Tests (MXT) [Pogossian99] as well as introduce a way for visual representation of the process for convenience of further analysis.

A software package is developed allowing to analyze the matrix of tests and recommend the optimal strategy using known Borda, Condorcet, Copeland and Simpson voting optimality criteria [Moulin88] and procedures. [Danielyan, Pogossian99].

Forming combinations of Strategy Plans and putting results of their testing into the Matrix of Tests

1. In terms of market competition Strategy Plans (SP) specify qualitative changes of operating parameters - Quality and Price, of competitors. From a variety of possible strategy plans to choose the best one we test them dynamically in competitive environments by a simulation.

In process of evaluation and selection a competitor's best SP, each of its tested plans is kept unchanged while all possible moves of other competing companies considered in accordance with their strategy plans are analysed. Thus, to get a complete behaviour of SP, its testing is repeated as many times during the simulation as many sub-combinations of other companies` plans can be generated.

Before the above described detailed testing of the plans they have to pass "static" tests aimed to eliminate plans with a big chance to be losing. It is based on estimation of the "ideal" results of companies when they act in the markets by their plans without presence of counteractions of other competitors (pure planning and acting). Other words, the plans have to be enough perspective at least without competitions to be considered in them.

Static testing allows to decrease a number of acceptable plans for each company and to reduce the amount of calculations.

In some experiments we don't perform SP static testing using the whole variety of plans and considering all possible combinations of them. Despite on the increasing of the amount of calculations they provide more information for understanding the processes and their analysis. Particularly, it allows to find (if necessary) the worst performance of the companies.

For convenience, SP are indexed and those indices are not changed during creation of the Matrix of Tests.

In this model of testing all meaningful SP are considered as a common resource in form of a vector and are available to all companies. Thus, all competitors have the same chances to choose perspective for themselves from the sets of SP.

2. Let's c is the number of companies competing in the market and p is the length of the SP vector.

A *combination of SP* is an arbitrary sub-vector of a length **c** of the basic SP vector where each competing company has only one corresponding plan (Fig. 1).

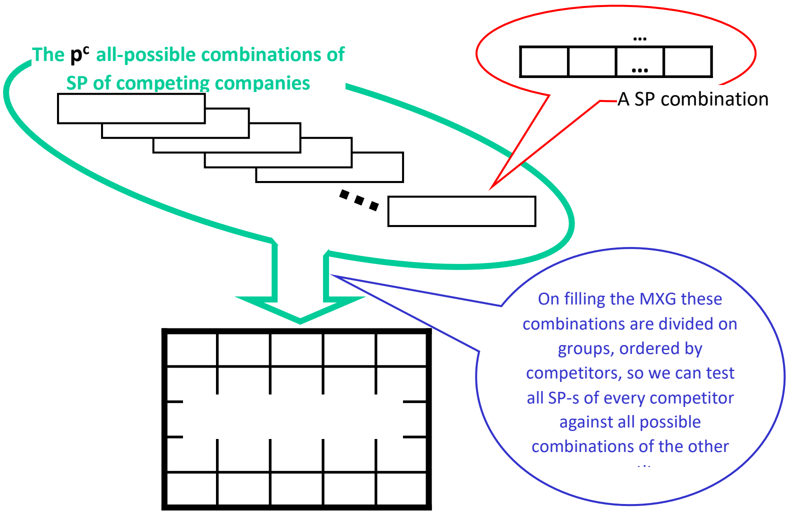


Fig. 1

In general, it is possible to test dynamically all p^c combinations of SP by the game tree.

A heuristic approach to the dynamic testing is that each of SP combinations is described in the consequent section as a process of "forming a single cell of row by dynamic testing of SP combination in the game tree" which is similar to the corresponding procedure in [Pogossian, Hovanissyan, Sukia¬syan 2001], [Pogossian, Baghda¬saryan 2003]. Ordered results of the testing are put into the Matrix of Tests (MXT) (Fig.1).

3. Since we have different participants-competitors we get the results of testing group-by-group.

A single group contains results of testing by all possible combinations of the SP referring to one given competitor.

Thus, the entire MXT consists of **c** sub-matrixes MXT_1, MXT_2, ... MXT_c (Fig. 2) where the number of MXT cells does not exceed $\mathbf{c^*p^c}$.

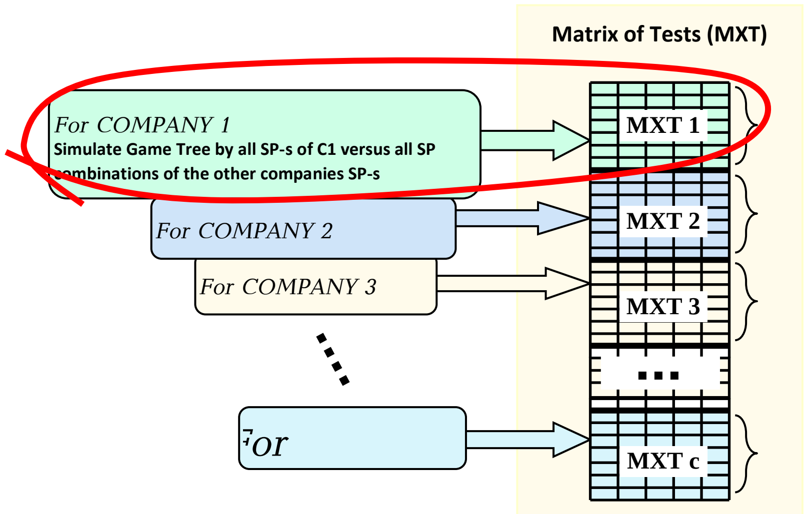


Fig. 2

To fulfill the MXT there is no need in all those $\mathbf{c} * \mathbf{p^c}$ computations. It is enough to get no more than $\mathbf{p^c}$ of them and allocate them consistently by those **c** sub-matrixes.

That is a right action as any combination of SP is relevant to all those groups for which corresponding competitors have in their sets of SP at least one SP that is included into that combination.

4. For an individual company the sub-matrix of the MXT consists of no more than **p** rows where each row represents results of testing of some SP of our company against all other combinations of SP of other **c**-1 companies.

Thus, there may be no more than $\mathbf{p}^{(c-1)}$ cells in a row, **c**> 1.

Fig. 3, 4 illustrate the structure of the MXT1 matrix and process of forming rows and cells for a company (say Company 1).

5. We have considered how combinations of SP are formed and how they are associated with the cells of the MXT matrix. The results of their game tree based dynamic testing are allocated in the MXT matrix cells.

For dynamic testing of the SP all analyzed SP get their quantitative interpretations to determine the game trees followed by evaluation of the quality of all perspective SP for the considered company by running their competitions with quantitative interpretations of combinations of all SP of other companies.

Our company determines quantitative values of its actions in accordance with its SP under the question, chosen "portions" of discrete changes "delta" for the P and Q (Price and Quality) and taking into account all possible quantitative variations of actions of other competitors in frame of considered combinations of their SP.

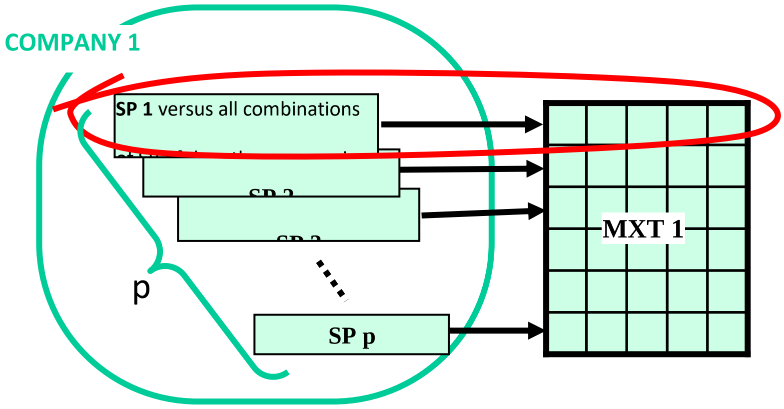


Fig. 3

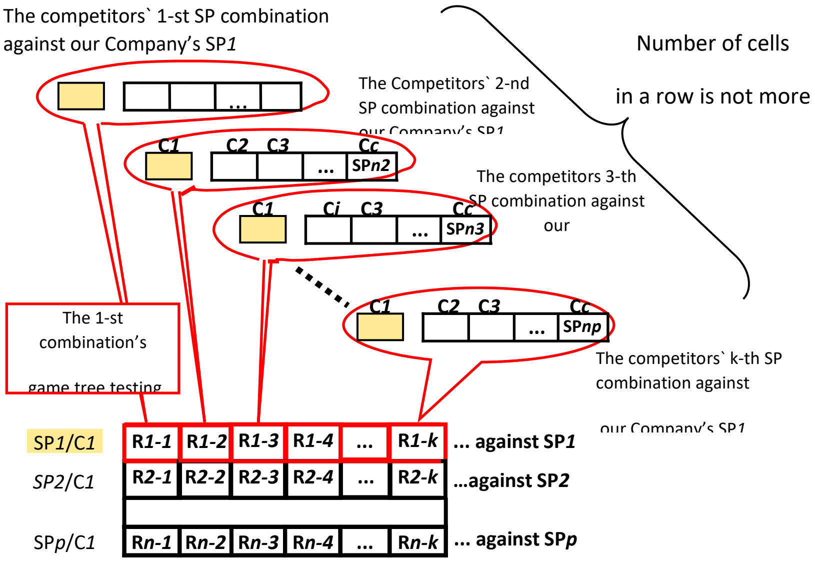


Fig. 4

Variations of quantitative actions of the competitors are generated as versions of the company actions in some range of possible changes described below.

Thus, each SP of the considered company has a variety of choices of the quantified actions in the market situations and for each of those choices a range of quantified actions of all competitors is considered.

The variety of all possible actions of all competitors determines a game tree associated with the SP under consideration. The best strategy for considered company in the game tree determines the value that the analyzed SP may provide (given success criteria for the competitions, e.g. profit, market share, etc.) which is allocated into the cell of the MXT prescribed to that SP.

To find the best strategy one of management strategy search methods may be used [Pogossian E., Pogossian L.,2000]. The main problem for these methods is in increasing the efficiency of the search due to a huge amount of possible quantifications that makes the size of the game tree too big to achieve that goal.

Some game tree based on dynamic testing methods were described in [Pogossian, Hovanissyan, Sukiasyan 2001], [Pogossian, Baghdasaryan 2003] demonstrating applicability of the mentioned above principles.

Let's suggest a heuristic approach for reducing the size of the game tree and making reasonable an application of standard game tree search methods like min max.

The idea of that reduction is in generation not all actions of opponents but determining by the value of actions of our company.

For each P and Q combination for our company and given units of their discrete changes (delta) only the following options for changes of the actions for the opponents are permitted:

- more than for our P or Q,
- equal to our P or Q changes and
- less than for our P or Q (Fig. 5).

Thus, for each choice of P and Q for our company 32 variations around them will be considered for other ones.

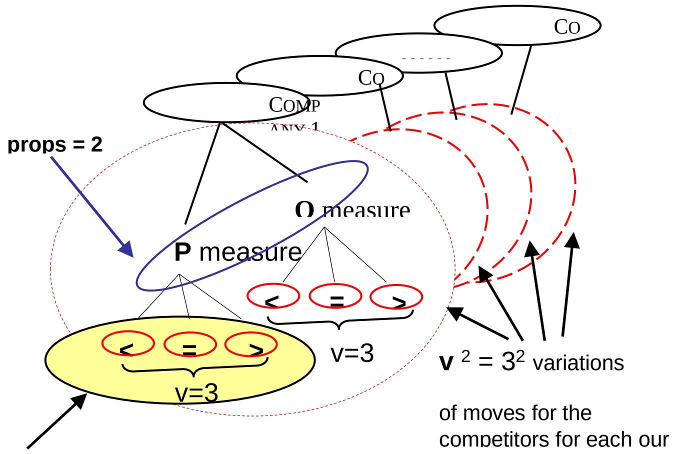
On this assumption we make actions that change our P and Q according to our strategy plan and as a response consider possible quantitative countermoves of other competitors by changing their P and Q to be "more", "equal" or "less" than our ones, Fig. 5.

Figure 6 shows a fragment of the game tree for a combination of two competitor's plans. The C1 company has **v** variants of quantitative actions for the parameter P while the opponent C2 has only three choices around the each C1 choice.

Figure 7 shows a fragment of the game tree for a combination of **c** competitors and their plans.

Our company again chooses **v** variants of quantitative actions according to our current SP, while our *opponents* have $3^{(\text{parameters*competitors})} = 3^{2^*(c-1)}$ variants for changes.

For any Δi change (delta P or delta Q) the delta values of opponents' changes - $\Delta P ijk$, $\Delta Q ijk$, $j=1, 2, \ldots, c-1$, $k=1, 2, \ldots, 3^{2^*(c-1)}$ are more, less or equal to Δi.



The idea of reduction of the size of the tree is in using for competitors` only moves that are “less”, “equal” and “more” than moves for our company.

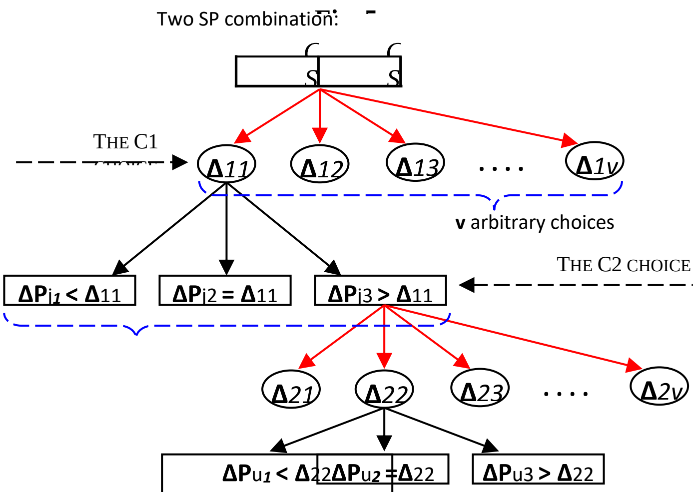


Fig. 6 A game tree with the use of the “more-equal-less” approach for the P parameter and two competitors.

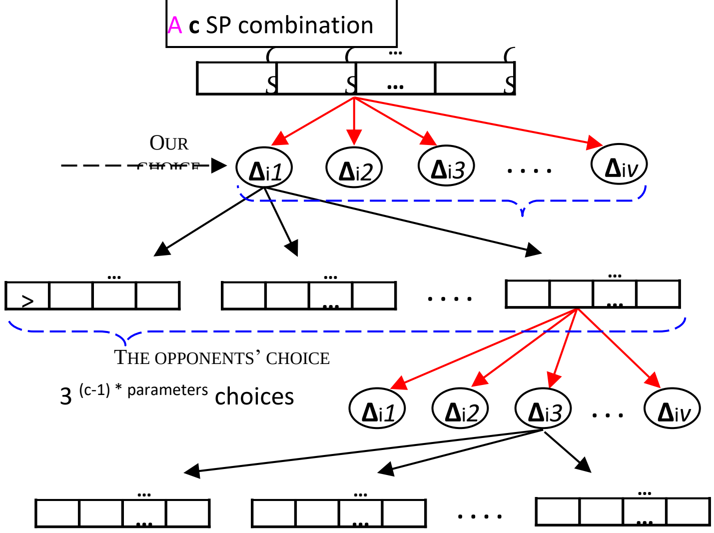


Fig.7 A game tree with the use of the “more-equal-less” approach for the P parameter and **c** competitors

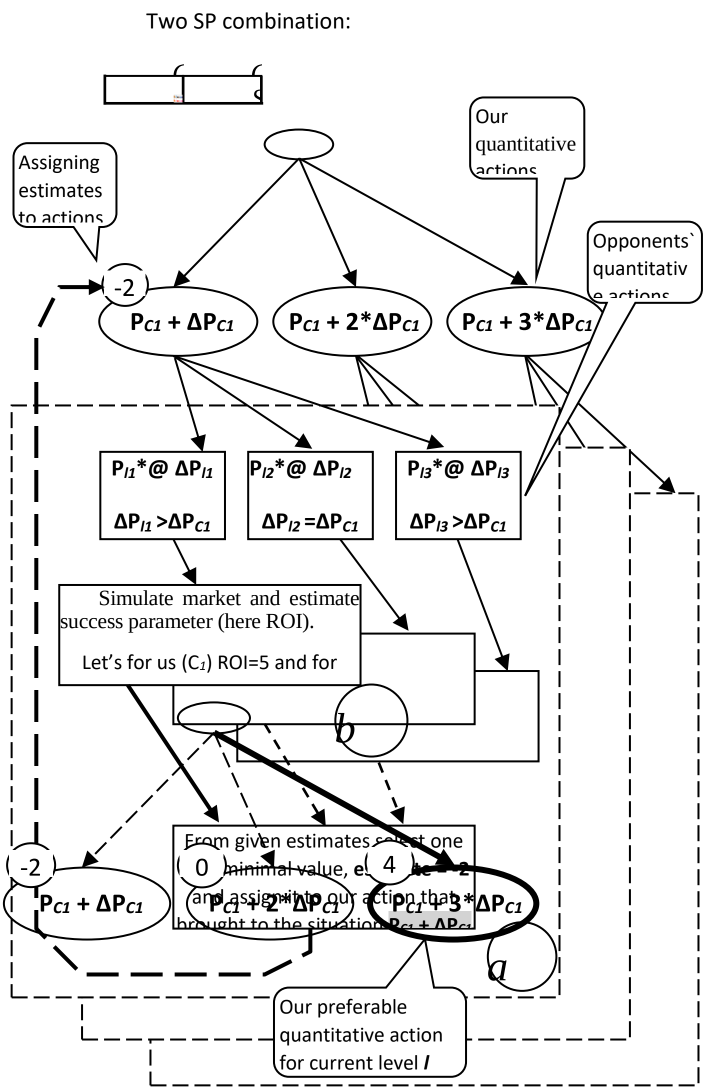


Fig. 8. Finding for our company C_1 the most preferable quantitative action on the "more-equal-less" basis by the min-max estimates lifting (illustrated for the case of two competitors and "Price" parameter).

Let's remind that P and Q values of the competitors are changed on the above "deltas" corresponding to the "logic" of their individual SPs.

The most preferable quantitative actions for the companies are selected using the min max search method (Fig. 8a, b).

Let's two companies are competing for the highest value of Return on Investment (ROI) and our company C_1 in accord with its SP have to rise its Price parameter.

Let's quantify that our SP rising policy as increasing the $\mathbf{P}_{c1}$ by $i{*}\Delta\mathbf{P}_{c1}$, $i=1, 2, \ldots, v$ (Fig.8 a).

In response to each of our actions the opponent makes its quantitative response-actions according to its SP where changes of its $\mathbf{P}_{I1}{*}$ by $\Delta\mathbf{P}_{Ii}$ ($\mathbf{P}_{I1}{*}@\ \Delta\mathbf{P}_{Ii}$) are permitted only in three ways:

$\Delta\mathbf{P}_{I1} > \Delta\mathbf{P}_{C1}$, $\Delta\mathbf{P}_{I2} = \Delta\mathbf{P}_{C1}$ and $\Delta\mathbf{P}_{I3} < \Delta\mathbf{P}_{C1}$.

After inputting new prices, the market model is run in order to determine the ROI of the companies for each of new combination of prices.

Defining the evaluation function as a difference (ROI_{c1}-ROI_{c2}) between ROI values of the companies gained at that market the root action of the C1 that had induced the responses of the opponent C2 is estimated by the best (the worst for us) action of C2, i.e. by one with the min difference ROI_{c1}-ROI_{c2} (here it is -2).

Estimating in the same way the all our actions, finally, we chose the one with the max possible value (here with value 4, Fig. 8b).

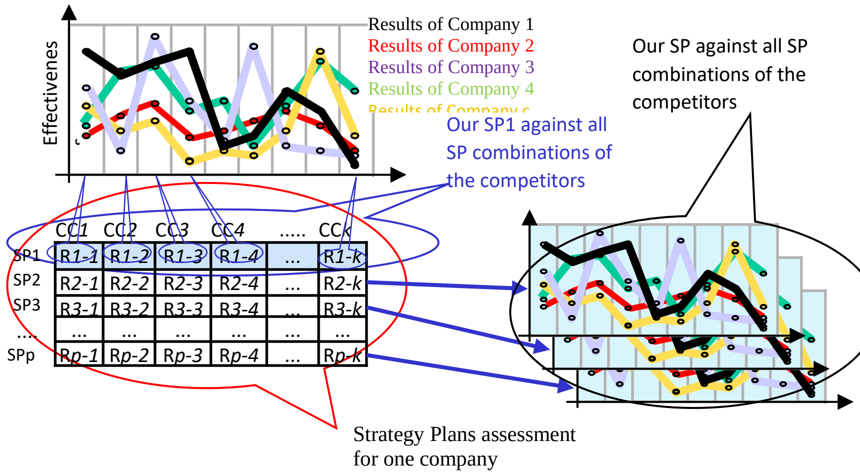


Fig.9

Number of calculations required to fulfill one cell of the MXT is $N = \sum_{i=1}^{d} v^i * 3^{i*p*(c-1)}$ nodes, where v is the number of our possible actions in a market situations, p – number of parameters, c – number of competitors, d – depth of MinMax search.

For experiments with the MXT matrix a software tool is realized in C++ /Visual C++ languages for Windows environment. The tool allows to set and change the competitors' parameters, generate the game tree and vary the min max search depth, number of our quantitative actions and delta parameter for their changing, the logic for changing the actions of competitors and criterion for evaluation the actions.

In preliminary experiments for the Windows 98, CPU 400Mhz, RAM 64M configuration fulfillment of one MXT cell for the $c \le 4$, $d \le 2$ or $c \le 3$, $d \le 3$ takes less than one minute.

Visualization of the Matrix of Tests

Distributions of effectiveness of SP of competitors are visualized for the convenience of testing results analysis (Fig.9).

For that along with a graph visualizing the distribution of effectiveness of our SP against all possible combinations of competing SP the graphs of the effectiveness of each company are provided.

That unified presentation of effectiveness graphs in the context of our SP assessment allows comparing them in the frame of each competition for the further analysis.

The Package for Optimal Strategy Plans Selection

1. The goal of this package is to provide users an easy-to-use, quick and useful tool for the optimal SP/strategy recommendation based on the given Matrix of Tests and optimality criteria.

The package is realized in C++ / Visual C++ programming languages for Windows OS and is a convenient tool with a friendly interface to get answer to the question:

Which SP /strategy is the best according to the tests results?

The core of the package is based on the well-known Borda, Condorcet, Copeland and Simpson's voting optimality criteria and procedures applied to the strategy assessment problem.

All mentioned methods are Pareto optimal, anonymous, neutral and monotone. Let's remind that such characteristics, as Pareto Optimality, Anonymity, Neutrality and Monotonicity along with the efficiency are the necessary requirements for the management optimal strategy provision problem.

2. Let's give a brief description of the package workflow.

- The program gets on **input** the *M(mxn)* matrix with tests' results $T = \{T_1, \ldots, T_n\}$ for the set of SP/ strategies $A = \{A_1, \ldots, A_m\}$. The matrix element m_{ij} is the value of test T_j result for the SP/strategy A_i. Actually *M(mxn)* is the Matrix of Tests.
- The user selects one of the suggested by package methods for the assessment of SP/strategies.
- After selecting some voting method for assessment, the program transforms the given matrix of tests results to the matrix corresponding to the selected method and makes appropriate estimation/calculation.
- The **output** of the package is the recommended SP/strategy (or the set of strategies) based on the optimality criteria and matrix of tests results.

3. Each voting method of the package uses its own matrix for assessment as the following:

- the matrix of tests is transformed to Borda matrix, if Borda rule is selected;
- if Condorcet method is selected the matrix is transformed to the Condorcet matrix, and so on.

The Borda Matrix is a matrix with

$$m_{ij} = p,$$

where *p=1, 2, …, m-1, m* according to what minimum score the *i*-th strategy has for *j*-th test. That is, $m_{ij}=1$ if *i-th* strategy has the minimum value to the test *j*, $m_{ij}=2$ if strategy *i* has second by minimum value to the test *j*, and so on. If strategy *i* has the maximum value to the test *j* then $m_{ij}=m$.

The Condorcet Matrix:

$$m_{ij} = \begin{cases} 0, & A_i \text{ is worse than } A_j \text{ for the majority of testers, or } i = j \\ 1, & A_i \text{ is better than } A_j \text{ for the majority of testers} \end{cases}$$

The Copeland Matrix:

$$m_{ij} = \begin{cases} 0, & \text{if } i = j \\ 1, & \text{if } A_i \text{ is better than } A_j \text{ for the majority of testers} \\ -1, & \text{if } A_i \text{ is worse than } A_j \text{ for the majority of testers} \end{cases}$$

The Simpson Matrix:

$$m_{ij} = \begin{cases} n, & i = j \\ N(A_i, A_j) \end{cases}$$

where $N(A_i, A_j)$ is a number of testers for which strategy A_i is better than strategy A_j.

More detailed description of the methods is presented in the [Moulin88], [Danielyan2004], [Danielyan, Pogossian99].

Conclusion

A game tree approach to dynamic testing and assessment of Strategic Plans is developed based on simulation of their on the job performance in oligopoly competitions. It covers the issues of cutting strategy search in the tree, organizing the results of testing into a matrix and visualizing them.

Preliminary experiments demonstrated a viability of the approach that is realized as a software tool in the Windows environment. Next step of its development is associated with construction of agents for optimal decision making in supply chain management applications.

The software package for strategy plans assessment having as an input the matrix of tests, transforms it to the matrix of appropriate method of voting and outputs the best strategy for a given set of strategies according to selected Borda, Condorcet, Copeland or Simpson's voting optimality criterion.

The users work with the package needs to go throught the following steps:

1. providing the matrix of tests
2. selecting the optimality criteria from the suggested list
3. processing the calculation of the best startegy according to matix of tests and selected optimality criterion.

The package can be applied to a variety of optimal strategy provision problems and is easy for expanding. Particularly, it is broadening now by the multistage elimination tree method based on Shepsle-Weingast algorithm [124] and majority-based optimality criterion [183].

Оценивание Стратегических Планов в Олигополической Конкуренции Посредством Моделирования их Реального Исполненения

В статье рассмотрен подход использования игрового дерева для динамической проверки и оценки стратегических планов, основанный на моделировании их поведения в условиях олигополической конкуренции.

В данном подходе рассматриваются вопросы сужения дерева поиска стратегий, распределения результатов проверки в виде матрицы и обеспечения их визуального представления.

Представлен программный пакет анализирующий матрицу для данного набора рассматриваемых стратегий и предлагающий наиболее оптимальную из них по одному из критериев голосования: Бордо, Кондорсе, Копеланду или Симпсону.

Օլիգօպօլիկ Մրցակցության Ստրատեգիկ Պլանների Գնահատում Նրանց Գործնական Կատարման Մոդելավորման Միջոցով

Յոդվածում դիտարկվում է խաղային ծառերի օգտագործման մոտեցումը ստրատեգիական պլանների դինամիկ տեստավորման և գնահատման համար` հիմնված օլիգոպոլիկ մրցակցային միջավայրում նրանց վարքագծի մոդելավորման վրա:

Տվյալ մոտեցման շրջանակներում դիտարկում են ստրատեգիաների փնտման խաղային ծառի նվազեցման հարցը, ստացված արդյունքների կառգավորումը մատրիցի տեսքով և վիզուալ ներկայացման հարցերը:

Ներկայացված է ծրագրային փաթեթ դիտարկվող ստրատեգիաների հավակածուի համար Բորդայի, Կոնդորսեի, Կոպելանի կամ Սիմպսոնի ընտրման չափանիշներից մեկով մատրիցի վերլուծումման միջոցով առավել օպտիմալի առաջարկման համար:

References:

[1] *Chussil M., Reibstein D.* 1997. Putting the Lessons before the Test. In Wharton on to Analyse & Develop Competitive Strategies, John Wesley & Sons, 343-368,

[2] *Chapman R.* 1997.BRANDMAPS: The Competitive Marketing Strategy Game. Prentice Hall,

[3] *Botvinnik M.M.* 1979. On Approximate Resolution of Search Problems. Sov. Radio, Moscow.

[4] *Danielyan E., Pogossian E.,* "An Application of Voting Methods to Strategy Assessment", Proceedings of the Conference on Computer Science and Information Technologies (CSIT'99), Yerevan, Armenia, 1999

[5] *Danielyan E.,* Application of Voting Methods to Strategy Analysis and Assessment", Mathematical Problems of Computer Science, 23, 2004, Yerevan, 102-118

[6] *Moulin H.* Axioms of Cooperative Decision Making". ISBN-0-521-36055-2, Virginia Polytechnic & State University, 1988

[7] *Porter M.* 1980. Competitive Strategy Techniques for Analysis Industries and Competitors. Free Press.

[8] *Pogossian E.* Adaptation of Combinatorial Algorithms. Academy of Sciences, Yerevan, 1983, 293pp.

[9] Pogossian E. Management Strategy Search and Assessment Programming. CSIT99, Yerevan, August 1999, p.46-66.

[10] Pogossian E., Pogossian L. On Management Strategy Search Simulation//Artificial Intelligence, 2000, N2, p.424-431. http://www.maus-soft.com/iai ;http://www.iai.donetsk.ua/_u/iai/dtp/ai.html

[11] *Pogossian E.* Focusing Management Strategy Provision Simulation. Proceedings CSIT2001

[12] *Pogossian E., Hovanissyan R., Sukiasyan M.* Java Program for Management Plans Dynamic Testing. Proceedings CSIT2001

[13] *Pogossian E., Baghdasaryan T.* 2003. Cutting Time of Strategy Search By Sequential Quantification Of Management Plans. Proceedings CSIT2003.

[14] *Pogossian E. Javadyan A.* A Game Model And Effective Counteraction Strategies Against Network Intrusion. 4th International Conference in Computer Science and Information Technologies, CSIT2003, Yerevan, 2003, pp.5

[15] *Pogossian E .* Business Measurements By On-The-Job Competition Scales. International Conference "Management of Small Business: Problems, Teaching, Future", Sept. 20th -24th 2004, Sevastopol, 4p.

[16] *Simonyan R.* Tolerance Spaces Induced by Majority Relationships. Mathematical Problems of Computer Science, 16, 1986, Yerevan, 50-67

Appendix 4: Security OJP Solvers

1.Pogossian E.: **Combinatorial Game Models for Security Systems**. NATO ARW on "Security and Embedded Systems", Porto Rio, Patras, Greece, Aug. (2005) 8-18

Abstract
An effective method of game based dynamic analysis of disturbances in security of systems and their preservation by elaboration of optimal strategies that can meet the requirements to embedded systems is presented. The results of preliminary experiments in strengthening intrusion protection strategies by expert knowledge and arguments in conceptual strategy knowledge simulation are discussed for further applications.
Key words: security, embedded systems, game model, strategy, knowledge based.

1. Introduction

Embedded systems are intensively intervening in business, management, service, military and other applications and although their development meets, in general, common problems of system design, they require harder parameters for efficiency, especially, for high speed, low cost, simplicity of implementation and usage.

We aim to present an effective method of dynamic analysis of possible disturbances in security of systems and their preservation by elaboration of optimal strategies that can meet the above requirements to embedded systems. In the variety of problems, we identify the class where *Space* of possible *Solutions* can be specified by combinatorial *Game Trees* (SSGT) and develop strategy formation algorithm - *Intermediate Goals at First* (IGAF).

The SSGT is a spacious class of problems with only a few following requirements to belong to:

The SSGT is a spacious class of problems with only a few following requirements to belong to:

1. there are (a) interacting actors (players, competitors, etc.) performing (b)identified types of actions in the (c) specified moments of time and (d) specified types of situations
2. there are identified benefits for each of the actors
3. the situations the actors act in and transformed after the actions have adequate models.

Many security and competition problems belong to SSGT class. Specifically, these are network Intrusion Protection (IP), Management in oligopoly competitions and Chess-like combinatorial problems. Many other security problems such as Computer Terrorism Countermeasures, Disaster Forecast and Prevention, Information Security, etc., announced by the NATO (http://www.nato.int/science/e/newinitiative.htm) as well as problems in [4,5,7,12] may be reduced to the SSGT class.

We prove an adequacy of game models for security problems by demonstrating their effectiveness compatible with system administrators [20, 21].

The IGAF like algorithms were studied in [1, 2, 13] and in [22, 24] where optimal strategy provision algorithms for chess and oligopoly market competitions were described. Suggested by Botvinnik the cutting-down tree algorithm for chess is based on the initial extraction of subgoals within a game tree, which allows to reduce sharply the searching tree as compared to the method of minimax [11]. The algorithm allows determination of moving trajectories of confronting parties in order to construct a zone around the extracted subgoal trajectory.

The IGAF algorithms were successfully probed in the network IP and management problems. For the IP problem the IGAF1 was outperformed system administrators and known standard protection

systems in about 60% in experiments on fighting against 12 different types of known network attacks [20, 21, 22].
To increase the efficiency of the IGAF1 algorithm its more advanced version was suggested able to acquire a range of expert knowledge in form of goals or rules and to increase the efficiency of strategy formation with increasing the amount of expert knowledge available to the algorithm. A viability of the IGAF2 algorithm was tested against representatives of six classes of attacks [17].
In the consequent chapters we define the SSGT class and give two detailed examples how to represent different application problems as the game models. Then we describe the concept of optimal strategy and two on the-job-performance base criteria for measuring the quality of strategies – the min max and Testing Matrix Analysis. In Chapter 4, we concentrate on the security problems describing the IGAF method and evidence in its compatibility with advanced IP programs and experts. Then, we present the results of preliminary experiments in strengthening IP strategies by expert rules and goals, and argue why expert conceptual strategy knowledge can be simulated and how the adequacy of models has to be proved. Conclusions summarize the results and further plans.

2. Combinatorial Game Tree Models

2.1. SSGT problems are identified in a unified way by game tree constituents, which creates the base for a unified methodology of their resolution. The constituents include, particularly, the list of competing parties and their goals, their actions and contractions, states of trees and rules for their transformations.
For the above problems the GT constituents are determined as the following:

- The Chess OSP problem:
  - white and black players with checkmate as the goal,
  - chess piece moves as (contra)actions, and
  - composition of the chess pieces on the board as specific game states transformed by actions corresponding to chess rules.
- The MOSP problem for the Value War [3, 23] model's interpretation:
  - a company competing against a few others with the goal to maximize Return-On-Investment,
  - changes of the product price and quality as the actions,
  - tree states are determined by the scenario of competition (i.e., the competition template formed from the conceptual basis of management theory), the set of parameters of current competition with that scenario and actions of competitors,
  - transformation rules are determined by general micro- and macro- economic laws, which applied to input states/market situations create new ones.
- The IP OSP problem:
  - network protection systems, e.g., system administrators or IP special software, combat against intruders or network disturbing forces (e.g., hackers or disturbances caused by technical casualties) to ensure that the network is kept in a safe and stable state,
  - network states are determined by the composition of current resources vulnerable to disturbances of the networks,
  - actions and (contra)actions are the means able to change resources and therefore transform states [20, 21].

To bring SSGT problems to adequate game tree models the science and art have to be fused. Two fundamental issues of *correctness and attainability* of the models problem arise. *Correctness*, in general, answers whether the model is adequate to the corresponding reality of the world to an extent that its solution can be utilized in that world. *Attainability* - whether one can get the solution of the model problem at an acceptable computational cost. We refine these concepts as following: whether the model space of solutions includes the best solution of the problem and whether one can achieve that best solution in the model [22].

For example, both correctness and attainability take place in the computer chess problem if the optimality is understood as a superiority of the program over the best human chess player. Indeed, the correctness is evident due to the formal nature of chess. The corresponding game tree generated in a computer is a complete model for representing all possible chess strategies. The attainability of the problem is proved by convincing success of chess programs over chess champions in many announced tournaments.

The chess game tree model has a form of And/Or tree [11] and visualized chess strategies look as in Fig.1. To unify concepts, algorithms, strategies, etc., game models for other SSGT problems are reduced to And/Or tree, as well.

In contrast to computer chess, accomplishing correctness for the MOSP needs special evidence. The MOSP problem includes, in particular, sub-models of markets, alternative strategies and competitions, evaluation and selection of strategies. Thus, the correctness of the MOSP is derivable from the adequacy of the market models and spaces of alternative strategies for the competitors. The attainability of the best solution is derivable, similar to chess, from strategy search and evaluation methods.

If an oligopoly competition is represented by a game tree, the correctness of the MOSP will depend on whether the market and the rules of its transformations can be described adequately.

Recalling that market models and rules are intensively used in business simulation games and have successful applications at least in business education we can argue about acceptable correctness of the MOSP game tree-based models.

2.2. Let us consider how the MOST problem is reduced to game tree model. We assume that each competitor is identified by a corresponding deterministic program and the competition in a market may be described by sets of situations, actions and strategies in discrete time periods.

Competitors make allowed actions, or *moves*, from corresponding sets **A1, …, Am** simultaneously, step by step and in T periods. We name a vector of such actions as a *bundle of actions*. Bundles of actions transform the initial situation S into sequences of new situations (one period of actions on the left tree of Fig.2).

We call a tree of all possible sequential bundles of actions of competitors from an initial situation S in T periods a *S-game tree,* or *S-tree*.

In fact, S-tree is the sum of performances of all possible competitors' programs started from S. The whole performance of the programs may be described by the forest of such trees from different initial situations. To avoid technical complications, we assume to have only one initial situation and all competitors have the same sets of allowed market moves.

We name the performance of the competitor C (i.e. the performance of corresponding program) in S-tree as a (*complete) S-strategy of C*.

A competition, or a game, of competitors C1, C2, ..., Cm is determined by the sample of corresponding programs and by the initial situation.

We evaluate the quality of a program by the forest of strategies generated by the program from all possible initial situations. Thus, to evaluate a program we have to consider all its possible games against all possible samples of other competitors in all possible initial situations. The number of competitors in samples depends upon the assessment objectives. For oligopoly competitions, for example, we have to consider all possible combinations of competitors which are in the oligopoly.

Since each competitor is represented by a program and the performance of the program in the S-tree is a S-strategy, in order to evaluate the quality of the program the assessment criterion K must be applied to that S-strategy. We suppose that the value of the criterion is determined by terminal nodes of the S-strategy. For example, we can use the profit gained by S-strategy as a criterion and calculate by averaging the values of the profit for all of its terminal nodes.

Given criteria F, we say that a *strategy G achieves the goal F* if criteria F satisfied for the set of terminal nodes of G. The strategy *G* will be called *F-projected* if we are interested in whether the terminal nodes of G satisfy criteria F or not. Any description of an F-projected strategy G aimed to make a search of G more efficient is named *a Strategy Plan* for *F.* A description of the problem under consideration and a description of the solving strategy itself constitute an extreme example of strategy plans. A useful strategy plan would systematically identify the directions that are not promising and eliminate them there for reducing the search space. Such a strategy plan would have to be described in a high-level language. *Strategy planning* is a process of narrowing the search space for the target strategy which also reflects the specifics of the planner such as knowledge of language, the system of values and methods of search.

Figure 2 illustrates reduction of an initial game tree model for competitors C1 against C2 and C3 to the AND/OR tree.

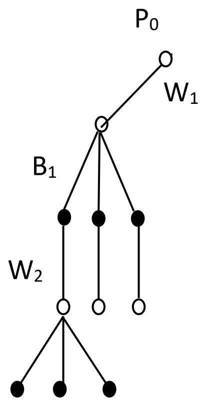


Fig.1

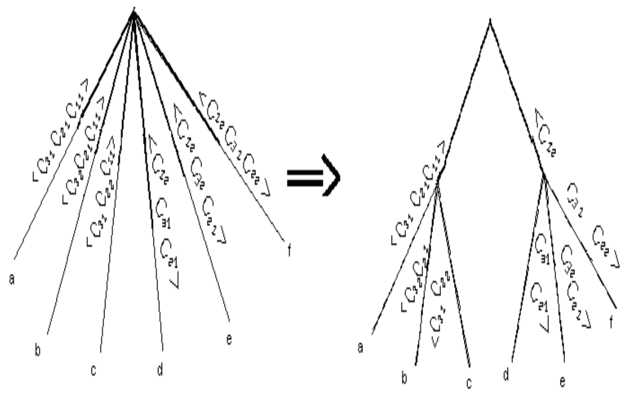


Fig. 2

2.3. The game tree model for security problems in its IP application version is defined as follows [20, 21].

It is a game between two playing in turn sides with opposite interests - the attacker (A) and the defender (D), described by a set of states and a collection of conversion procedures from one position to another defined as the following.

In particular, *system resources* are processor time, the size of TCP buffer, and a number of incoming packages. Let R = {r} be a non-empty set of the system resources and Q be a set of resource parameters. Different measuring scales, such as seconds, bytes, numbers of incorrect logins or incoming packages, etc., are used to measure different parameters. Each $r \in R$ is associated with a pair <q; w> - a real system resource, where $q \in Q$, $w \in W$ and W is a set of possible scales.

A criterion function is an arbitrary function f with the range of values Z = [0, 1] and F is the set of such functions f.

A local system resource state on a non-empty set $R' \subseteq R$ is called the value $e \in Z$ of the criterion function $f \in F$ on this set: e = f(r1, r2, …, rk), where R' = (r1, r2, …, rk) & $\varnothing \neq R' \subseteq R$.

The *local state* is called *normal* if e = 0 and *critical* if e = 1, and L will denote the set of local states e. Intuitively, by criterion functions are measuring “distance” of current states from those that are considered as normal. A system state on a non-empty set $L' \subseteq L$ is called the value $s \in Z$ of the criterion function $g \in F$ on this set: $s = g(e1, e2, \ldots, en)$, where $L` = (e1, e2, \ldots, en)$ & $\varnothing \neq L` \subseteq L$ The *state* is called *normal* if $s = 0$ and *critical* if $s = 1$, and S will denote the set of states s.

The main goals of the attackers and defenders are to bring the system in the critical states and avoid them, correspondingly.

Let us call an arbitrary function p (si, sj), the ranges of definition and values of which are subsets of R, *a conversion procedure* of system from the state si to sj:

$p (si, sj) : \{\{r1, r2, \ldots, rk\}\} \rightarrow \{\{r`1, r`2, \ldots, r`k\}\}$, where $\{r1, r2, \ldots, rk\} \subseteq R$ & $\{r`1, r`2, \ldots, r`k\} \subseteq R$.

Let P is the set of conversion procedures, Pa and Pd are its subsets for the attacking and the counteracting sides, $pa(si, sj) \in Pa$ and $pd(si, sj) \in Pd$ are the conversion procedures from the state $si \in S$ to $sj \in S$ for the attacking and the counteracting sides, correspondingly.

The counteraction game model is represented by "AND/OR" tree G (S, P). At first the attacker moves from the initial state $s0 \in S$ then the defender replies in turn. Thus, the initial node s0 is an "AND" type. The terminal nodes correspond to the winning states of the defender.

3. Game Model Security Strategies are Compatible with Experts

To prove correctness of the security game tree model strategies of the model were compared with the experts and existing contemporary security systems strategies by the following on –the-job performance testing methodology.

The quality of strategies, ideally, is determined by analysis of their on-the-job performance, i.e. the results of all possible interactions/competitions with all possible opponents/competitors. The results of competitions correspond to the values of terminal nodes of corresponding strategies on the game tree.

We use the following two concretizations of the on-the-job performance criteria:

- the Min max criterion, where the optimal strategy and corresponding best move is determined by raising up the values of terminal nodes,
- the Testing Matrix Analysis (TMA) criterion, where the vectors of terminal nodes of strategies comprise a matrix analyzed by voting-like methods [9, 19].

We implement the Min max and TMA criteria through corresponding software. Although the Min max criterion is widely presented in strategy search studies and our software is following the state-of-the-art, we have developed original software for the TMA. The first tool aims to form the matrix of tests by results of on-the-job performance of strategies [17]; a current version of the tool is specialized for experiments with defense strategies to support IP Systems. The second package takes as input the matrix of tests and selects the best strategy by a variety of methods including voting ones [18]. It must be acknowledged that the Min max and TMA provide different criteria of optimality.

We demonstrate the correctness and attainability of the game model by its effectiveness for IP systems in a series of experiments on protection against attacks, such as Syn-Flood, Smurf, Fraggle, overflow of the buffer, IP-Hijacking, etc., where the model recommends decisions compatible with system administrators [20, 21].

It was found that for making decisions compatible with experts it was enough to make an exhaustive, min max search in the game tree for depth in 3 "moves". The quality of decisions was permanently increasing when the depth of search was increasing and for depth in 5 "moves" the model was avoiding about 70% experts' false alarms. Unfortunately, the search time was increasing exponentially and for depth higher than 5 it became unacceptable.

The IGAF2 algorithm succeeds to reducing the burst of time in the strategy search and to achieve acceptable efficiency.

4. Empowering Security Strategies by Expert Knowledge

The IGAF2 algorithm is composed of the following operations [17]:

- it uses standard min max technique with alpha-beta pruning based on the range of critical/normal state values introduced as the goal. The current node is created and the value of its local state calculated. If the node is terminal, the local state value is compared with sibling nodes and their max (or min) value is sent to the parent node [20, 21],
- it determines all suspicious resources,
- it builds the game subtree for suspicious resources starting from the root state of the tree and using the 4th group of rules it determines the trajectories of attacks,
- it calculates the values of the terminal states of the tree, it finds the values of others using the min-max procedure and determines the best min-max action from the root state,
- it determines the trajectories of attacks induced by the best action from the root of the tree to its critical states and it considers them as targets,
- it builds the zones of counteractions for the target trajectories using the 4th group of rules and the 5th rule; then, it calculates the values of the states of the corresponding subtree using min-max,
- it chooses the defender's action from the root as the one leading to the state with min value, i.e. to the most stable state estimated by min-max,
- it finishes the defense analysis and waits for the attacker's actions.

The IGAF2 algorithm is able to acquire a range of expert knowledge in the form of goals or rules and to increase the efficiency of strategy formation by increasing the amount of expert knowledge available to the algorithm. The following expert goals and rules have been embedded in the IGAF2 algorithm [17].

The goals are:

- the *critical vs. normal states* are determined by a range of values of the states of the system; for example, any state of the system with a value of corresponding criterion function, that is more or equal to some threshold, may be determined as a critical goal,
- the *suspicious vs. normal resources* are determined by a range of states of the classificators of the resources; combinations of values of the classificators identified as suspicious or normal induce signals for appropriate actions.

1. The rules are:

- identify the suspicious resources by the classifiers and narrow the search to the corresponding game subtree,
- avoid critical states and tend to the normal ones,
- normalize the state of the system; first, try such actions of the defender that influence on the resources that caused current change of its state; if they don't help, try other ones,
- when building the game subtree for suspicious resources, use
  - defending actions able to influence on such resources,
  - normal actions until there is no critical states,
  - if some defensive actions were used on previous steps decrease their usage priority
- Balance the parameters of resources by keeping them in the given ranges of permitted changes.

IGAF2 was tested in experiments against six attacks with a depth of the game tree search up to 13 and the following controlled and measured criteria and parameters: distance to safety, productivity, working time and number of game tree nodes searched , new queue of incoming packages, TCP connection queue, number of processed packages, RAM, HD, unauthorized access to files and login into the system [17]. The results of the experiments show:

- Sampling means for Distance to Safety and Productivity of the IGAF2 and min-max algorithms are compatible.
- Number of nodes searched by the IGAF2 algorithm with all expert rules and subgoals are decreasing compared with the IGAF1 algorithm or the min-max one.

- Number of nodes searched by the IGAF2 algorithm with all expert rules and subgoals is the smallest compared with the IGAF1 algorithm or the min-max one when the depth of search is increasing up to 13
- The time spent by the IGAF2 algorithm with all expert rules and subgoals is the smallest compared with the IGAF1 algorithm or the min-max one when the depth of search is increasing up to 13.

We plan to strengthen the IGAF2 algorithm through systematic enrichment of the knowledge base by new expert goals and rules. The question is whether it is possible to construct adequate models of strategy expert knowledge.

5. On Simulation of Strategy Expert Knowledge

We consider a comprehensive repository of *communicable* expert knowledge - concepts, descriptions, rules and other related texts, for a typical representative of the SSGT class – chess problem, in order to answer the following principal questions:

- Is it possible to find an adequate computer representation for an “alive” fragment of expert knowledge associated with solving problems of a target class?
- Is it possible to reproduce, or learn, procedurally with an acceptable computational complexity the expert knowledge associated with the problem?

In [16] the following statements were argued. Chess concepts identify configurations of elements of positions having winning by Zermelo [15] utility, which provides additional reasons to possibility of their computer simulation.

An uncertainty in representation of chess concepts is generic, caused by limited resources and individualized their learning by players. Despite concepts with the same name, which can have different interpretations by different players and coincide only in some “skeleton” parts, this does not create big casualties because of an intermediate usage of concepts for further deeper analysis of positions. Due a generic uncertainty in representation of concepts, it is worth to formulate the chess problem in a way allowing both forming concepts with strong common meaning and ones that exist in the forms individualized to the particular players. Correspondingly, adequacy of the models of concepts have to be examined taking into account those individualized representations of concepts.

6. Conclusion

- The IGAF algorithms are consistent with strategy expert knowledge and can be applied to security problems have the following advantages:

- elaborate optimal defense strategies for any identified moment of decision making,
- counteracting a broad spectrum of disturbances and attacks that is possible to generate in the given alphabets of elementary attacks and defense actions,
- improvement of generated strategies by acquiring expert knowledge in the field,
- inheritance and development of achievements in combinatorial games, particularly chess, one of the most advanced studying in Computer Sciences.

Applied to the intrusion protection problem, the IGAF2 algorithm –for a range of types of knowledge in form of goals and rules— demonstrates strong tendency to increase the efficiency of strategy formation with an increase in the amount of knowledge available to the system. For further increase of the efficiency, new adequate models of expert knowledge are needed. For chess, a typical and representable game problem, it was argued that chess specifies in an individualized and quasi way the classes of winning by Zermelo positions, which provides important reasons to possibility of their computer simulation. By our preliminary analysis, the remaining chess knowledge repository –plans, ideas, attributes, etc— specify additional tangible constructions of the game tree –the winning strategies— which can be simulated as well. Current experiments [18] aim to prove those assertions.

If experiments are successful, it can be expected that concepts and strategy knowledge of the SSGT problems can be computer simulated, because simulation of chess knowledge was induced by their reduction to constructions of the game tree and the game trees are the base for other SSGT problems, as well.

Chess conceptual and strategy knowledge simulation add arguments in favor of simulation of learnable knowledge not only for the SSGT problems but for a wider class of problems as well; this is, because, usually, the problems can be represented in form of games as well as to the power of prepositional form of simulation [10]. Finally, to fuse the high effectiveness of IGAF algorithms in embedded systems, it is necessary to estimate the computational resources in the context of embedded systems.

References

[1] Botvinnik M.M., About solving approximate problems, S. Radio, Moscow, 1979 (in Russian)

[2] Botvinnik, M.M., Computers in Chess: Solving Inexact Search Problems. Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag: New York, 1984.

[3] Chussil M., Reibstein D. 1997 Putting the Lessons before the Test. In Wharton on to Analyse & Develop Competitive Strategies, John Wesley&Sons, 343-368,

[4] Chi, S.-D., Park, J.S., Jung, K.-C., Lee, J.-S.: Network Security Modeling and Cyber Attack Simulation Methodology. Lecture Notes in Computer Science, Vol.2119 (2001).

[5] V. Gorodetski, I. Kotenko: Attacks against Computer Network: Formal Grammar Based Framework and Simulation Tool. Proc. of the 5 Intern. Conf. "Recent Advances in Intrusion Detection", Lecture Notes in Computer Science, v.2516, Springer Verlag, pp.219-238, 2002.

[6] Flavell J. The Develometal Psycology Of Jean Piaget, D. Vannostrand Company Inc., Princeton, NJ, 1962.

[7] I.Kotenko, A. Alexeev E., Man'kov. Formal Framework for Modeling and Simulation of DDoS Attacks Based on Teamwork of Hackers-Agents. Proc. of 2003 IEEE/WIC Intern. Conf. on Intelligent Agent Technology, Halifax, Canada, Oct. 13-16, 2003, IEEE Computer Society. 2003, pp.507-510.

[8] Kruchten P. The Rational Unified Process. An Introduction, Addison-Wesley, 2000.

[9] Moulin. H. Axioms of Cooperative Decision Making. Virginia Polytechnic & State University, 1988, ISBN-0-521-36055-2

[10] Pylyshyn Z. Seeing and Visualizing: It's Not What You Think, An Essay On Vision And Visual Imagination, http://ruccs.rutgers.edu/faculty/pylyshyn.html

[11] Russell S., Norvig. P. Artificial Intelligence: A Modern Approach. Prentice-Hall, Englew. Cliffs, NJ, 2002.

[12] Sheyner O., Wing J., Tools for Generating and Analyzing Attack Graphs, to appear in Proceed. of Formal Methods for Components and Objects, Lecture Notes in Computer Science, 2005.

[13] Stilman, B. Linguistic Geometry: From Search to Construction. Kluwer Academic Publishers. 2000.

[14] Winograd T., Flores F. 1986. Understanding Computers and Cognition (A new foundation for design).

[15] Zermelo E. Uber Eine Anwendung Der Mengenlehre Auf Die Theorie Des Shachspiels. Proceedings of The Fifth International Conference of Mathematicians, Cambridge, 1912, Cambridge Press, p.501-504.

[16] Pogossian E. On Simulation of Strategy Conceptual Knowledge, CSIT2005, Yerevan, 8pp.

[17] Pogossian E., Javadyan A., Ivanyan E. Effective Discovery of Intrusion Protection Strategies. Lecture Notes in Computer Science, Ais-Adm-05: The International Workshop on Autonomous Intelligent Systems - Agents and Data Mining June 6-8, 2005, St. Petersburg, Russia http://space.iias.spb.su/ais05/, LNAI 3505, pp.263-276, 2005

[18] Karapetyan G., Vahradyan V., Pogossian E. Experiments in Simulation of Chess Conceptual Knowledge Proceedings of the CSIT2005, 5th International Conference in Computer Science and Information Technologies, Yerevan, 2005

[19] Baghdasaryan T, Danielyan E., Pogossian E. Testing Oligopoly Strategy Plans by Their On The Job Performance Simulation, CSIT 2005, Yerevan, 2005, pp.7
[20] Pogossian E. Javadyan A. A Game Model and Effective Counteraction Strategies Against Network Intrusion., CSIT 2003, Yerevan, 2003, pp.5
[21] Pogossian E. Javadyan A. A Game Model for Effective Counteraction Against Computer Attacks In Intrusion Detection Systems, NATO ASI 2003, Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management", Tsahkadzor, Armenia, August 19-30, pp.30.
[22] E. Pogossian Focusing Management Strategy Provision Simulation. CSIT 2001, Yerevan, 2001
[23] Pogossian E. Increasing Efficiency of Management Skill Assessment. Proc. of the 26th annual conference of the Association for Business Simulation and Experiential Learning (ABSEL-99), Philadelphia, March.
[24] Pogossian E. The Adaptation of Combinatorial Algorithms. National Academy of Sciences of Armenia, Yerevan, 1983, pp.290 (a book in Russian).

2. E. Pogossian, A. Javadyan and E. Ivanyan., "**Effective Discovery of Intrusion Protection Strategies**" The International Workshop on Agents and Data Mining, Lecture Notes in Computer Science, St. Petersburg, Russia, pp. 263-274, 2005.

Abstract. Effectiveness of discovery of strategy knowledge is studied for problems where the space of hypothesis of solutions is specified by game trees and target solutions are discovered by methods capable of systematic acquisition of expert knowledge about them. A version of Botvinnik's Intermediate Goals at First algorithm is developed for strategy formation based on common knowledge planning and dynamic testing of the plans in the corresponding game tree. Applied to the intrusion protection problem the algorithm for a range of types of knowledge in form of goals and rules demonstrates strong tendency to increasing the efficiency of strategy formation with an increase in the amount of knowledge available to the system.

1 Introduction

Our starting idea about the nature of knowledge states that solutions of problems, components of the solutions or any of their descriptions, as well as any procedure or record acquired by systems in a way that improves their performance, are knowledge.

In the framework of that idea, solutions of problems with human independent formal specifications, such as problems of mathematical optimization, program synthesis, improvement, etc., may deliver knowledge of various kind but say little about the nature of human knowledge.

That is why we associate further progress in understanding human knowledge with studying the problems where the quality of target solutions is determined by the adequacy of models of human knowledge imbedded in those solutions and being inseparable from them. The corresponding class of problems, named Solvable by Adequate Models of Human Knowledge, includes, in particular, problems where solutions have to be learned by systematic acquisition of human knowledge in a given application area.

A pioneer research of strengthening the performance of chess programs simulating the process of chess masters' decisions by systematic acquisition of human knowledge was studied in [2, 3, 4] and developed in [24].

A significant advance in ontology of the security domain, ontology-based representation of distributed knowledge of agents, formal grammar of attacks and their application to the network intrusion protection systems as well as a comprehensive review of ontology studying in the field are presented in [6, 8].

In [15] an attempt was undertaken to study viability of a decision-making system with a variety of chess knowledge, including the ontology of about 300 concepts. For problems of that class hypothesis about solutions are *strategies* and *space* of their search is specified by *game trees* (SSGT) and target solutions have to be discovered by methods able to systematic acquisition of knowledge about them. Many security and competition problems, belong to the SHGT class, particularly, networks intrusion protection, management in oligopoly competitions as well as disaster forecast, computer terrorism countermeasures and prevention, information security and medical countermeasures announced by NATO may be reduced to the SSGT class (www.nato.int/science/e/newinitiative. htm).

To solve the SSGT problems we define a class of combating agents, based on the following models and procedures:

- a game tree model for the target competition, including the sub-models of the states, actions and contractions, the rules to apply (contra)actions to the states and transform them to the new ones, descriptors of the goal states
- the optimal strategy search procedure, including the strategy planning unit, aimed to narrow the search area in the game tree, the plans quantification, their game tree based dynamic testing and the best actions selection units.

A Common Planning and Dynamic Testing methodology for combating agents is developed allowing to construct agents with the best, in the framework of corresponding game tree models, strategies. The methodology in its Botvinnik's approach version - Intermediate Goals At First (IGAF1), was successfully probed in the network intrusion protection (IP) and some other SSGT problems. For example, for the IP problem it was outperforming system administrators and known standard protection systems in about 60% in experiments on fighting against 12 different types of known network attacks [16, 17].

To increase the efficiency of the IGAF1 algorithm we suggest its more advanced version able to acquire a range of expert knowledge in form of goals or rules and to increase the efficiency of strategy formation with increasing the amount of expert knowledge available to the algorithm.

A viability of the IGAF2 algorithm was tested against representatives of four classes of attacks. There is not comprehensive classification of the variety of network attacks [5, 6, 10], because of widely distribution of mechanisms of attacks, their distribution in time, influence on different types of resources, dynamic increase of their number, possible damage and many other reasons. That is why the attacks for our experiments only approximate the coverage of a few known classes of attacks.

A Linux version of the IGAF2 algorithm is now realized for intrusion protecting system of the ArmCluster [1].

Compared with [20, 22, 23], where network-vulnerability analysis is based on finding critical paths in the graphs of attack , in our game tree based model we search counteraction strategies comprised from elementary and universal units – elementary procedures, an alphabet, that intruder or administrator use to combine either attacks or defense procedures, correspondingly. Some of those procedures can coincide, particularly, with elementary attacks of the [20, 22, 23]. But the aim is to find procedures enough elementary to cover diversity of intruders and defenders' behaviors but nevertheless being meaningful for human understanding and operations. Alphabetic approach to representation of attacks and defense operations causes game tree size explosion which we try to overcome using computer chess successful experience.

IGAF based agent combats against other ones (one or more) where the game tree integrates their all possible behaviors – strategies. The agent has to discover new knowledge - the optimal strategy, analyzing enormous amount of mainly useless data. These intersections make strategy discovery interesting for traditional DM and multi agents researchers to use experience gained in many years studying of strategy discovery.

Development of IP agents can directly contribute to other SSGT problems what is confirmed by our experience in computer chess [15] as well as in management support systems for oligopoly competitions

[19]. That is why we restrain to specify control problems impossible to reduce to the SSGT class and are ready to share experience with enthusiasts studying other representatives of the SSGT class.

The paper is structured as follows. Section 2 outlines game tree model for intrusion protection, defines expert goals and rules and the IGAF algorithm for game tree knowledge-based strategy formation. Section 3 describes the strategy evaluation methodology and Section 4 presents the experiments in strategy discovery. The Conclusion outlines the main results of the research.

2. Game Tree Knowledge Based Strategy Discovery for Intrusion Protection

2. 1 Game Tree Model for Intrusion Protection

The *game tree* model for Intrusion Protection is presented in [16, 17]. In brief it is a game between two playing in turn sides with opposite interests - the attacker (A) and the defender (D), described by a set of states and a collection of conversion procedures from one position to another defined as the following.

System resources are particularly processor time, the size of TCP buffer, and a number of incoming packages. Let R = {r} is a non-empty set of the system resources and Q is a set of resource parameters. Different measuring scales, such as seconds, bytes, numbers of incorrect logins or incoming packages, etc., are used to measure different parameters. Each $r \in R$ is associated with a pair <q; w> - a real system resource, where $q \in Q$, $w \in W$ and W is a set of possible scales.

A criterion function is an arbitrary function f with the range of values Z = [0, 1] and F is the set of such functions f.

A local system resource state on a non-empty set $R` \subseteq R$ is called the value $e \in Z$ of the criterion function $f \in F$ on this set: e = f(r1, r2, …, rk), where R` = (r1, r2, …, rk) & $\varnothing \neq R` \subseteq R$.

The *local state* is called *normal* if e = 0 and *critical* if e = 1, and L will denote the set of local states e. Intuitively, by criterion functions are measuring "distance" of current states from those that are considered as normal. A system state on a non-empty set $L` \subseteq L$ is called the value $s \in Z$ of the criterion function $g \in F$ on this set: s = g(e1, e2, …, en), where L` = (e1, e2, …, en) & $\varnothing \neq L` \subseteq L$ The *state* is called *normal* if s = 0 and *critical* if s = 1, and S will denote the set of states s.

The main goals of the attackers and defenders are to bring the system in the critical states and avoid them, correspondingly.

Let us call an arbitrary function p (si, sj), the ranges of definition and values of which are subsets of R, *a conversion procedure* of system from the state si to sj:

p (si, sj) : {{r1, r2, …, rk}} $\rightarrow$ {{r`1, r`2, …, r`k}}, where {r1, r2, …, rk} $\subseteq$ R & {r`1, r`2, …, r`k} $\subseteq$ R.

Let P is the set of conversion procedures, Pa and Pd are its subsets for the attacking and the counteracting sides, pa(si, sj) $\in$ Pa and pd(si, sj) $\in$ Pd are the conversion procedures from the state si $\in$ S to sj $\in$ S for the attacking and the counteracting sides, correspondingly.

The counteraction game model is represented by "AND/OR" tree G (S, P), where S and P are finite, non-empty sets of all states (nodes, vertices) and all possible conversion procedures (edges, ribs), correspondingly (Fig. 1).

At first the attacker moves from the initial state s0 $\in$ S then the defender replies in turn. Thus, the initial node s0 is an "AND" type. The terminal nodes correspond to the winning states of the defender.

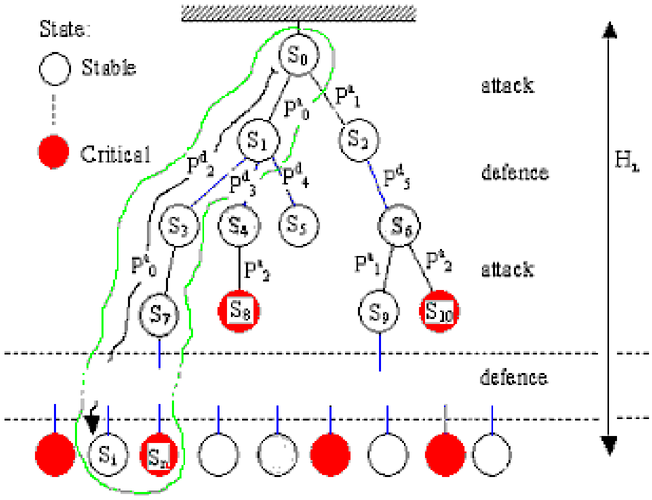


Fig. 2. A game tree model for the Intrusion Protection problem.

2.2 Expert Goals and Rules Based Strategy Discovery

2.2.1 Using the above game tree model we experiment with a variety of algorithms to counteract to intrusions. Our Intermediate Goals at First algorithm (IGAF1) is similar to Botvinnik's chess tree cutting-down algorithm. The last is based on the natural hierarchies of goals in control problems and the assertion that search algorithms become more efficient if try to achieve subordinate goals before fighting for the main ones. The trajectories of confronting parties to those subgoals are chained in order to construct around them the zones of the most likelihood actions and counteractions.

As the result of comparative experiments with the minmax and IGAF1 algorithms in [16, 17] we state the following:

- the model, which is using the minimax algorithm, is compatible with experts (the system administrators or specialized programs) against intrusions or other forms of perturbations of the base system
- the IGAF1 cutting-down tree algorithm along with being compatible with the minimax one can work enough efficient to be used for real IP problems.

Here we suggest more advanced version 2 of the algorithm – IGAF2, which is able

- to acquire a range of expert knowledge in form of goals or rules
- to increase the efficiency of strategy formation with increasing the amount of expert knowledge available to the algorithm.

Let us describe the types of that knowledge and their inclusion into the IGAF algorithm.

2.2.2 We rely on the following concepts of "trajectory of an attack" and "zone of counteraction".

The trajectory of an attack is a subtree Ga(S', P') (Fig.2), where S' is a subset of the system states $S' \subseteq S$ and P' is a subset of the actions, consisted of an offensive's conversion procedures Pa and a defender's normal conversion procedures Pdn, i.e. $P' = Pa \cup Pdn$, $P' \subseteq P$, $P' \neq \varnothing$.

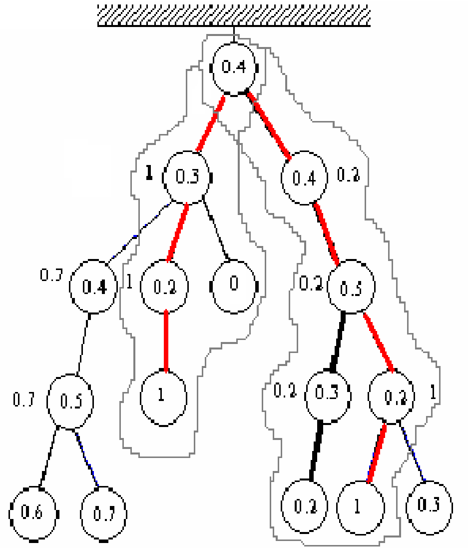


Fig. 3. Two trajectories of attack with zones of counteractions

The zone of counteraction is a subtree Gz(S", P") built around the graph of the trajectory of an attack Ga(S', P'), i.e. Ga $\subseteq$ Gz, where S" is a subset of the system states S" $\subseteq$ S, which belong to the trajectory of an attack, hence S' $\subseteq$ S", and P" is a subset of the actions, which consist of the conversion procedures, defined on the trajectory of an attack, P' and the defender's special conversion procedures Pds, i.e. P"= P'$\cup$ Pds, P"$\subseteq$P, P" $\neq \varnothing$, hence P'$\subseteq$ P".

2.2.3 The following expert goals and rules have been embedded into the IGAF2 algorithm:

The goals:

1. the *critical vs. normal states* are determined by a range of values of the states of the system; for example, any state of the system with a value of corresponding criterion function, that is more or equal to some threshold, may be determined as a critical goal
2. the *suspicious vs. normal resources* are determined by a range of states of the classificators of the resources; combinations of values of the classificators identified as suspicious or normal induce signals for appropriate actions.

The rules:

1. Identify the suspicious resources by the classifiers and narrow the search to corresponding game subtree
2. Avoid critical states and tend to the normal ones
3. Normalize the state of the system. First, try such actions of the defender that influence on the resources caused current change of its state and if they don't help try other ones
4. In building game subtree for suspicious resources use

– defending actions able to influence on such resources
– use normal actions until there is no critical states
– if some defensive actions were used on previous steps decrease their usage priority

5. Balance the parameters of resources by keeping them in the given ranges of permitted changes.

2.2.4 The IGAF2 algorithm is composed using the following instructions:

1. We use standard min max technique with alpha-beta pruning [21] based on the range of critical/normal state values introduced as the goal 1. Current node is created and the value of its local state calculated. If the node is terminal, the local state value is compared with sibling nodes, and their max (either min) value is sent to the parent node. C++ program level realization of it is described in [16, 17].

2. Determine all suspicious resources.
3. Build the game subtree for suspicious resources starting from the root state of the tree and using the 4th group of rules determine the trajectories of attacks (Fig.3).
4. Calculate the values of the terminal states of the tree, find the values of others by minmax procedure and determine the best minmax action from the root state (the green branch on the Fig.4).
5. Determine the trajectories of attacks induced by the best action from the root of the tree to its critical states and consider them as targets (Fig 5).

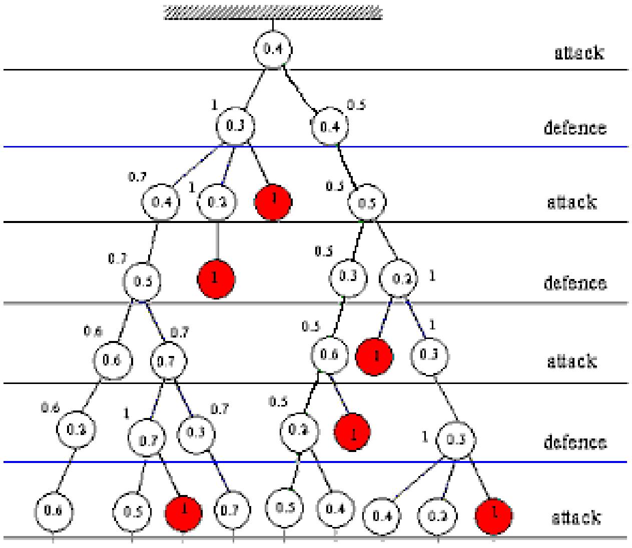


Fig. 4. The game subtree

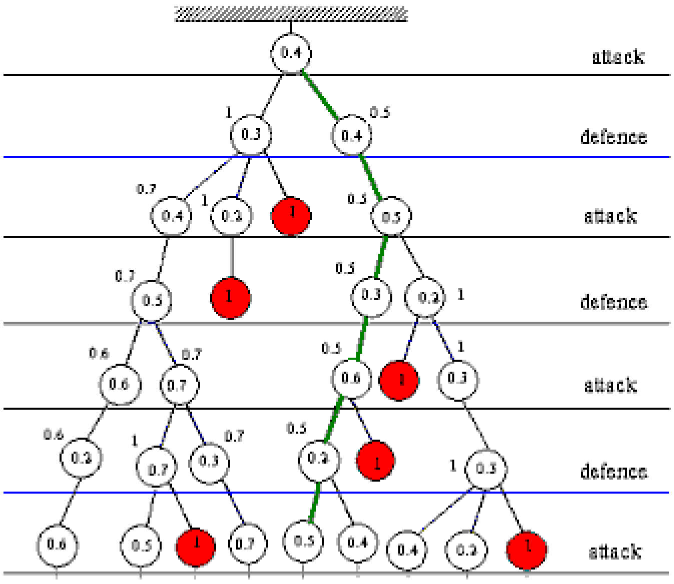


Fig. 5. The best minmax action.

6. Build the zones of counteractions for the target trajectories using the 4th group of rules and rule 5th, then calculate the values of the states of the corresponding subtree using the minmax (Fig.6).
7. Choose the defender's action from the root as the one leading to the state with min value, i.e. to the most stable state estimated by the minmax.
8. End the defense analysis and wait for the attacker's actions.

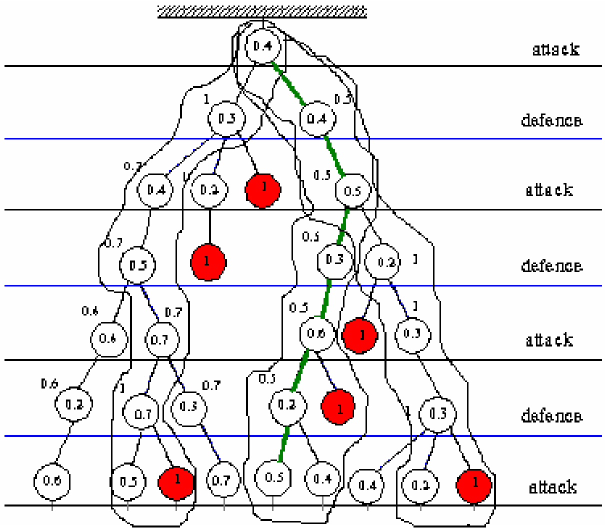


Fig. 6. The trajectories of attacks

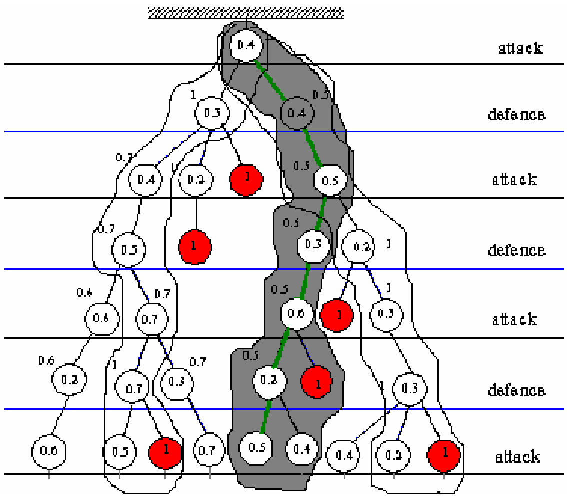


Fig. 7. The zones of counteractions

3 Strategy Evaluation Methodology

3.1 The criteria for comparing the IP algorithms estimate the 'distance" of current states of protected systems from the normal ones and the level of performance that the IP algorithms can preserve for them. They are:

1. *Distance to Safety (DtS)* estimates in the scale [0,1] the distance of the current state of the system from the safe or normal ones.
2. *Productivity (P)* is the number of performance units preserved in a unit of time for the system in average in process of an experiment. The performance units are defined in the context of current focus of interests of the users and may be measured, for example, by number of packages or files the system processes, by number of served of users, etc.

3.2 A special tool is developed to estimate the quality of protection against unauthorized access [18]. The tool allows to vary the component parts of the experiments, such as estimated criteria, the types of attacks, and the parameters of the IP systems.

Each experiment includes the imitation of the work of the base system with/without suspicion of an attack (or any other perturbation in the system), during which the IP algorithm makes a decision about the best strategy and chooses the best action according to the strategy. The data of the experiments of any attacks contain the system state's safety estimate for each step, the actions taken by each side, the system's performance.

3.3 The experimented attacks have to be representative for the variety of possible ones. We assume combinatorial, individual nature of the attacks that are unified in the classes of similar ones. Experimenting with a few representatives of the classes we hope to approximate a coverage of their variety. We chose the following groups of criteria to classify them:

1. Denial of Service
   a. Remote DoS
   b. Local DoS
2. Scanner
   a. Network scanner
   b. Vulnerability scanner
3. Penetration
   a. Remote penetration
   b. Local penetration
4. Information gathering
5. Password cracker
6. Sniffer
7. Suspicious activity
8. Unauthorized access
9. Code errors

In the experiments we choose representatives of four classes of attacks described in [10, 11]. The criteria they satisfy by the groups are listed in the (Table 1).

Table 2. Classes of attacks

	Attack Name	By groups
1	SYN-Flood	1.a,1. b
2	Smurf	4, 3.b, 1. b
3	Fraggle	8, 1. b
4	Login-bomb	5

3.4 Each protecting system was tested against all attacks and corresponding vectors of the results were compared and analyzed for each attack separately and by all attacks in average.

The performance of a protection system against any attack was estimated by the means of corresponding distributions of measurements by each of two criteria in the series of offense –defense pairs of actions.

Any series were including about 200 pairs of those actions.

For each series the sampling means for 200 values measured by the criteria – Distance to Safety and Productivity, were calculated and identified with the means of the distributions. That identification is consistent with the Central Limit Theorem applied to a large sample of measurements with unknown distribution.

5 Experiments in Strategy Discovery

5.1 The experiments were aimed to proving that the IGAF2 cutting-down tree algorithm along with being compatible with the minimax one is working more efficient increasing the efficiency with embedding the amount of expert knowledge.

The investigated version of the algorithm was using the following components:

- Over twelve single-level and multilevel solver-classifiers of the local system states.

- 6 actions-procedures of the attacking side
- 8 “normal” actions/procedures of the system
- 22 known counteractions against actions/procedures of attacks (the database of counteractions).

It was tested in experiments against four attacks with a depth of the game tree search up to 13 and the following controlled and measured criteria and parameters: distance to safety, productivity, working time and number of searched in the game tree nodes, new queue of incoming packages, TCP connection queue, number of processed packages, RAM, HD, unauthorized access to the files and login into the system.

The results of experiments below are presented mainly for the SYN-Flood attack because they are typical for all of them.

Table 3. Sampling means for Distance to Safety (with Standard deviation) and Productivity of the IGAF2 and minmax algorithms against attacks SYN-Flood, Fraggle, Smurf and Login-bomb attacks are compatible

	Distance to Safety		Standard. Deviation		Productivity	
Attacks	Min-Max	IGAF2	Min-Max	IGAF2	Min-Max	IGAF2
SYN-Flood	0,7575	0,7897	0,0888	0,0925	3,14	3,6
Fraggle	0,7254	0,6445	0,1332	0,2068	2,1	2,2
Smurf	0,584	0,7004	0,1861	0,1417	3,9	4,2
Login-bomb	0,4662	0,5416	0,2483	0,2275	6,8	7,2

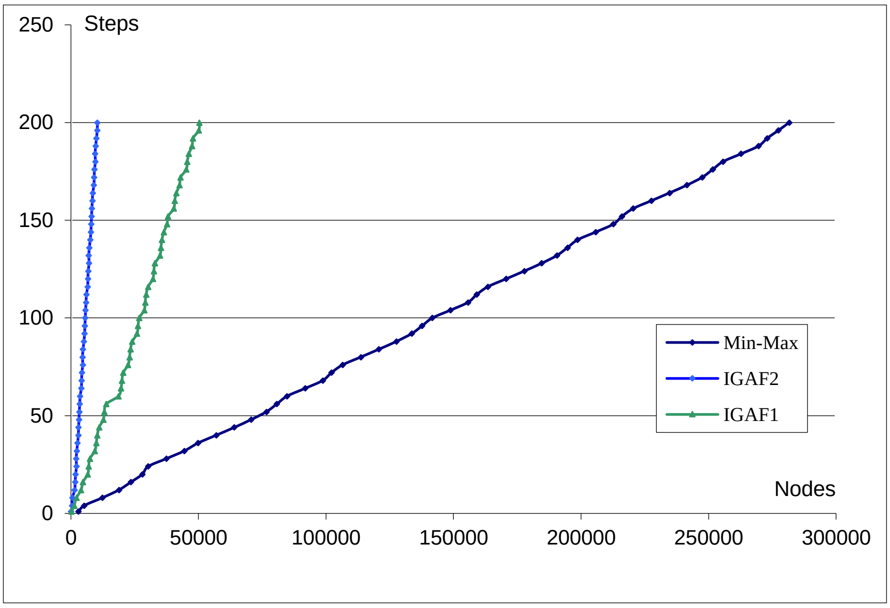


Fig. 8. Number of nodes searched by the IGAF2 algorithm with all expert rules and subgoals are decreasing compared with the IGAF1 algorithm or the minimax one. [experiments against SYN- Flood attack with the depth of search 5 and 200 steps of the defense actions].

6. Conclusion

A version of Botvinnik’s Intermediate Goals at First (IGAF2) algorithm is developed able to acquire a range of expert knowledge in form of goals or rules and to increase the efficiency of strategy formation with increasing the amount of expert knowledge available to the algorithm.

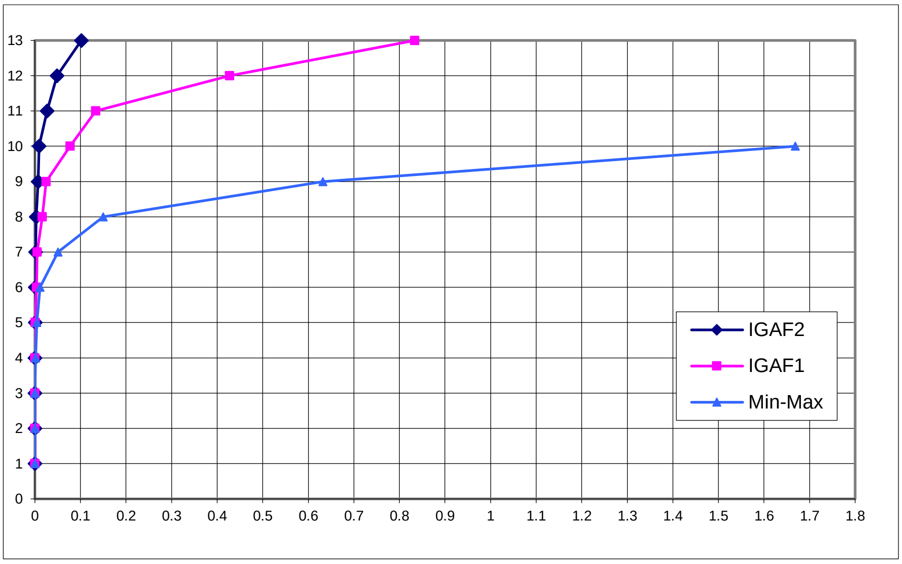


Fig. 9. Number of nodes searched by the IGAF2 algorithm with all expert rules and subgoals is the smallest compared with the IGAF1 algorithm or the minimax one when the depth of search is increasing up to 13 in experiments against the SYN- Flood attack.

Table 4. Number of nodes searched by the IGAF2 algorithm and time spent (in milliseconds) with all expert rules and subgoals is the smallest with the IGAF1 algorithm or the minimax one when the depth of search is increasing up to 13 in experiments against the SYN- Flood attack.

Depth of search	IGAF2		IGAF1		Min-Max	
	Node Count	Time	Node Count	Time	Node Count	Time
1	19	269	19	270	21	480
2	42	271	42	290	64	490
3	70	300	70	305	388	580
4	254	529	406	570	1162	771
5	562	561	1030	750	4231	1541
6	943	921	3580	1692	11512	3473
7	1775	1060	5083	1982	50797	13717
8	2987	1611	16493	5636	150096	43365
9	7389	2994	24502	8110	631960	172436
10	9503	4546	77573	25192	1668643	590242
11	27076	10822	133448	43655	--	--
12	48428	20675	427218	141689	--	--
13	101905	41932	833876	284948	--	--

A viability of the IGAF2 algorithm was successfully tested in the network intrusion protection problems against representatives of four classes of attacks: SYN-Flood, Fraggle, Smurf and Login-bomb, allowing to formulate the following statements:

- Sampling means for Distance to Safety and Productivity of the IGAF2 and minmax algorithms are compatible.
- Number of nodes searched by the IGAF2 algorithm with all expert rules and subgoals are decreasing compared with the IGAF1 algorithm or with the minimax one.

- Number of nodes searched by the IGAF2 algorithm with all expert rules and subgoals is the smallest compared with the IGAF1 algorithm or with the minimax one when the depth of search is increasing up to 13
- The time spent by the IGAF2 algorithm with all expert rules and subgoals is the smallest compared with the IGAF1 algorithm or with the minimax one when the depth of search is increasing up to 13.
- The recommended version of the IGAF2 algorithm with all expert rules and subgoals, for the depth of search 5 and 200 defending steps is overperforming the Productivity of minmax algorithm in 14%, using for that 6 times less computing time and searching 27 times less nodes of the tree.

The future plans include, particularly, the following lines:

1. Expanding the alphabet of attack and defense actions including hidden ones
2. Developing our approach to increase the strength of the IGAF algorithms by systematic enrichment their knowledge base by new IP goals and rules.
3. Developing on the job performance versions of the IGAF algorithm for actual applications, particularly completing its Cluster IP implementation [1].

References

1. H. V. Astsatryan, Yu. H. Shoukourian,, V. G. Sahakyan. The ArmCluster1 Project: Creation of High-Performance Computation Cluster and Databases in Armenia Proceedings of Confer-ence. Computer Science and Information Technologies, 2001, pp. 376-379
2. M.M. Botvinnik, About solving approximate problems, S. Radio, Moscow, 1979(Russian)
3. Botvinnik, M.M., Computers in Chess: Solving Inexact Search Problems. Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag: New York , 1984.
4. Botvinnik, M. M., Stilman, B., Yudin, A. D., Reznitskiy , A. I., Tsfasman, M.A., Thinking of Man and Computer, Proc. of the Second International Meeting on Artificial Intelligence, Re-pino, Leningrad, Russia, Oct. 1980, pp. 1-9.
5. Chi, S.-D., Park, J.S., Jung, K.-C., Lee, J.-S.: Network Security Modeling and Cyber Attack Simulation Methodology. Lecture Notes in Computer Science, Vol.2119 (2001)
6. V. Gorodetski, I. Kotenko: Attacks against Computer Network: Formal Grammar Based Framework and Simulation Tool. Proc. of the 5 Intern. Conf. "Recent Advances in Intrusion Detection", Lecture Notes in Computer Science, v.2516, Springer Verlag, pp.219-238, 2002.
7. V. Gorodetski, I. Kotenko, O.Karsaev.Framework for ontology-based representation of dis-tributed knowledge in multiagent network security system. Proc. of the 4-th World Multi-conf. on Systems, Cybernetics and Informatics (SCI-2000), vol. III, "Virtual Engineering and Emergent Computing", Orlando, USA, July 2000. pp. 52-58.
8. I.Kotenko, A.Alexeev E., Man'kov. Formal Framework for Modeling and Simulation of DDoS Attacks Based on Teamwork of Hackers-Agents. Proc. of 2003 IEEE/WIC Intern. Conf. on Intelligent Agent Technology, Halifax, Canada, Oct. 13-16, 2003, IEEE Computer Society. 2003, pp.507-510.
9. U. Lindqvist, P.A. Porras, Detecting Computer and Network Misuse with the Production-Based Expert System Toolset, IEEE Symp. Security and Privacy, IEEE CS Press, Los Alami-tos, Calif., 1999
10. A. Lukacki, Attack Detection, BXB-Petersburg, St. Petersburg, 2001 (in Russian)
11. N. G. Miloslavskaya, A. I. Tolstoj, Intranetwork: Intrusion Detection ,Unity, Moscow, 2001 (in Russian)
12. K. Ilgun, R.A. Kemmerer, P.A. Porras, State Transition Analysis: A Rule-Based Intrusion Detection System, IEEE Trans. Software Eng. vol. 21, no. 3, Mar. 1995
13. V. Paxson, Bro: A System for Detecting Network Intruders in Real-Time, Proc. Seventh Usenix Security Symp., Usenix Assoc., Berkeley.
14. E.Pogossian Focusing Management Strategy Provision Simulation. Proceedings of the CSIT2001, 3d International Conference in Computer Science and Information Technologies, Yerevan, 2001
15. E.Pogossian. Adaptation of Combinatorial Algorithms.(inRussian), Yerevan., 1983, 293 pp.

16.Pogossian E. Javadyan A. A Game Model And Effective Counteraction Strategies Against Network Intrusion. 4th International Conference in Computer Science and Information Tech-nologies, CSIT2003, Yerevan, 2003, pp.5
17.Pogossian E. Javadyan A. A Game Model For Effective Counteraction Against Computer At-tacks In Intrusion Detection Systems, NATO ASI 2003, Data Fusion for Situation Monitor-ing, Incident Detection, Alert and Response Management", Tsahkadzor, Armenia, August 19-30, pp.30.
18.Pogossian E. Javadyan A.,Ivanyan E. Toward a Toolkit for Modeling Attacks and Evaluation Methods of Intrusion Protection, Annual Conference of the State Engineering University of Armenia, 2004, pp.5 (in Russian).
19.Pogossian E.. Focusing Management Strategy Provision Simulation. Proceedings of the CSIT2001, 3d International Conference in Computer Science and Information Technologies, Yerevan, 2001, 5p.
20.Phillips C., Swiler L.. A Graph-Based System for Network-Vulnerability Analysis, New Se-curity Paradigms Workshop ,Proceedings of the 1998 workshop on New security paradigm
21.Russell S., Norvig. P. Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs, NJ, 2002
22.Sheyner O., Jha S., Haines J.,Lippmann R., Wing J., Automated Generation and Analysis of Attack Graphs .Proceed. of the IEEE Symposium on Security and Privacy, Oakland, 2002.
23.Sheyner O., Wing J., Tools for Generating and Analyzing Attack Graphs, to appear in Pro-ceed. of Formal Methods for Components and Objects, Lecture Notes in Computer Science, 2005.
24.Stilman, B., Linguistic Geometry: From Search to Construction, Kluwer Academic Publishers, Feb.2000, 416 pp.

Appendix 5: OJP in Defense

1.E. Pogossian, D. Dionne, A. Grigoryan, J. Couture and E. Shahbazian, "**Developing Goals Directed Search Models Empowering Strategies Against Single Ownership Air Threats**," International Conference in Computer Sciences and Information Technologies, pp. 155-163, Yerevan, 2009.

ABSTRACT

This paper studies the construction of expert knowledge-based game models with the objective of increasing effectiveness of naval defense strategies against threats.
Game trees are generated for a simplified single ownship scenario that includes: uncertainty in the output of defense actions, varying time intervals for actions, and irregular undertaking of the actions.
Experiments with deliberative planning with the game model demonstrated increased probabilities of survival of the ownship with respect to the worst-case situation.

1. INTRODUCTION

1.1. The objective of the paper is to develop effective decision-making systems for both defense and terrorism problems. The paper demonstrates increased effectiveness of strategies against threats elaborated from expert knowledge.
The work was realized in framework of the NATO Program of Security Through Science for Defense Against Terrorism, by IPIA, Academy of Sciences of Armenia and Centre for Research in Mathematics at the University of Montreal, Canada.
Pertaining to countermeasure against terrorism, the NATO Program identifies the following tasks:
a) Increasing timeliness and completeness of detection and identification of threat,
b) Providing an estimate of severity of threats,
c) Selecting strategies to act against the threats, and
d) Building a decision tree of all possible countermeasures against all threats and chose the most effective path in the three.
We study these tasks through the framework of games where hypotheses about solutions are known as *strategies,* where the *search space* is specified by *reproducible game trees* (SSRGT) and target solutions have to be discovered by methods able to systematic acquired knowledge about them.
For SSRGT problems it is typical to require acting according to optimal strategies, i.e. requiring that the solutions have to deliver recommendations on how to interpret the real world and how to act in it in the best way. For example, in the Intrusion Protection (IP) problem an agent - a decision-making system - stands against the intrusion by analyzing the possible strategies throughout a game tree and searching the best protection strategy.
Despite the fact, that the protection of networks has been becoming more effective, the detection of intrusions will remain the integral part of each serious secure system. There are two main categories of intrusion detection methods: the detection of anomalies [Chi, S.-D., Park, J.S., Jung, K.-C., Lee, J.-S.2001] and the detection of abuses [Ilgun, Kemmerer, Porras95; Paxson98].
1.2. In our approach to solve the IP problem [PogossianJavadyan2003] in addition to the known approaches of static identification, during the process of analyzing anomalies, the model of dynamic protection allows performing possible sceneries of intrusions and recommends the best way of protecting from them.
We define a class of *combating agents*, based on the following models and procedures:

- a game tree model for the target competition, including the sub-models of the states, actions and contractions, the rules to apply (contra)actions to the states and transform them to the new ones, descriptors of the goal states

- the optimal strategy search procedure, including the strategy planning unit, aimed to narrow the search area in the game tree, the plans quantification, their game tree based dynamic testing and the best actions selection units.

A Common Planning and Dynamic Testing methodology for combating agents is developed allowing constructing agents with the best, in the framework of corresponding game tree models, strategies. For example, for the IT problem it was outperforming system administrators and known standard protection systems in about 60% in experiments on fighting against 12 different types of known network attacks [Pogossian, Javadyan 2003].

To increase the efficiency of the IGAF1 algorithm we developed its more advanced version able to acquire a range of expert knowledge in form of goals or rules and to increase the efficiency of strategy formation with increasing the amount of expert knowledge available to the algorithm. The new IGAF2 algorithm for a range of types of knowledge in form of goals and rules demonstrates strong tendency to increase the efficiency of strategy formation with increasing the amount of knowledge available to the system [Pogossian, Javadyan, and Ivanyan2005].

1.3. Defense strategies developed by Lockheed Martin Canada consist of a twofold approach:

1. *Evaluating and ranking the threats based on their opportunity, capabilities and intent* [Menard, Couture] [Couture, Menard].
2. *Make the threat ineffective and/or remove the threats by producing reactive or deliberative engagement planning.*

The ultimate goal for establishing a careful evaluation of threats is to optimize the efficiency of engagement planning that will neutralize or remove the threats. Lockheed Martin Canada has studied over the years different schemes to build engagement plans. In general, a compromise between the times allowed computing the plan and the quality of the plan must be made. Tabu search and genetic algorithms have been examined and have provided some interesting planning results but required significant amount of time when the number of threats is large, i.e. typically larger than 5 or 6.

The SSRGT methodology presented in the current project offers a promising alternate way to investigate planning. At first glance, the current engagement situation can easily be mapped for example to a chess game: ships and planes for both threats and own force represent chess pieces, which move and attack according to specific rules (e.g. speed, maneuverability, weaponry) and pursue the goal of destroying the other force's assets or main asset.

1.4. The objectives of the project are to develop new expert knowledge-based models for increasing effectiveness of defense strategies against threats, including

1. *models of defense problems where class of hypotheses about solutions are strategies and space of their search is specified by game trees*
2. *IGAF strategy search algorithms for defense problems*
3. *models of expert knowledge empowering IGAF strategies against threats.*

The paper presents results of adequate modeling of the one ownship air threats defense problem by SSRGT games.

2. SCENARIO

The scenario involves two parties designated "defense" and "threats", respectively. Each party contains players. Each player responds to the actions taken by the opposite party.

The defense party has a single player, i.e., the ownship.

The threats party may have several players in the form of missiles and aircrafts. The types of threat players can be regrouped into categories, e.g., missile of type xxx, aircrafts of type yyy. An additional category can be defined for threat players whose type is uncertain.

In the simplified scenario, all the threat players belong to a single category of missiles. Several threat players may attack concurrently.

The threat players are generated as follows:

d) All threat players are created at the start of the scenario.

e) The maximum number of threat players is $N_{\text{threats}}^{\max} = 8$.

f) The initial position of each threat is uniformly and randomly selected in an area of space satisfying the conditions:
   - Initial range of 5 to 80 km from ownship,
   - Polar angle between 0° and 90° (i..e, angle in the vertical plane),
   - Any azimuthal angle (i.e., angle in the horizontal plane).

Assumptions:

i. It is assumed that the threats are ranked by the defense player. In the simplified scenario, the ranking function is the range: the closer the threat, the higher the rank of the threat.

ii. It is assumed that the defense player may bundle up concurrent defense actions. The admissible bundles must satisfy the engagement rules.

iii. A bundle of defense actions must ensure that only one defense action per threat is undertaken at any given time.

Each action results into a transformation in the scenario. The sets of defense actions, defense bundles, threat actions, and their associated transformation rules are assumed finite and known.

3. DEFENSE AND THREAT STRATEGIES

The objective of a defense strategy, P_D, is to prescribe a unique defense bundles for every admissible threat action. Since the defense strategy is in general not unique, one must be selected from the known set of admissible defense strategies, S_D. The selection process requires the formulation of a defense objective and of a corresponding utility function to be maximized. The selected defense strategy is denoted $P_D^* \in S_D$.

Similarly, a threat strategy, P_T, is composed of a set that prescribes a threat action bundle for every admissible defense action. The threat strategy belongs to a known set of admissible threat strategies, S_T. A threat objective and a threat utility function are required to select a unique threat strategy.

Example:

Let S_D = {D1, D2, D3} be the set of admissible defense bundles, and let S_T = {T1, T2, T3} be the set of admissible threat actions. One admissible defense strategy, P_D, could be

P_D = {T1->D1, T2->D2, T3->D1}

where T1->D1 means that if the threat action T1 is undertaken, the defense bundle D1 is prescribed by the defense strategy.

3.1 Objectives, utility functions, and formulation of the game

In the simplified scenario, the ownship has for primary defense objective to survive, while the primary objective of the threat party is to damage the opponent. The defense and threat objectives are described by utility functions denoted U_D and U_T, respectively.

Two utility functions are involved, one for the defense player and one for the threat players. The purpose of the defense utility function is to weight each defense strategy with respect to the defense objectives. Similarly, the threat utility function weights the threat strategies with respect to the threat objectives.

The defense utility function $U_D : P_D \times P_T \to \mathrm{R}^1$ is selected to be the probability of survival of the ownship in the worst-case scenario. The worst-case scenario is the one in which the threats and the ownship always survive to the defense and threat actions.

In the simplified scenario with a single threat action, the probability of survival is calculated as:

$$U_D\left(P_D, P_T\right) = f\left(P_D\right) \prod_{i=1}^{N_{\text{engaged}}^{\text{threats}}} \left(1 - P_{\text{kill}}^{i}\left(P_D\right) P_{\text{kill}}^{\text{threat}}\right)$$

where

$$P_{\text{kill}}^{i}\left(P_D\right) = \prod_{j=1}^{N_{\text{actions}}^{i}} \left(1 - P_{\text{kill}}^{\text{actions}}(i, j)\right)$$

$$f\left(P_D\right) = \begin{cases} 1, & \text{if } P_D \text{ engages the } N_{\text{engaged}}^{\text{threats}} \text{ closest threats} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{\text{engaged}}^{\text{threats}} = \begin{cases} N^{\text{threats}}, & \text{if } N^{\text{threats}} \leq 3 \\ 3, & \text{otherwise} \end{cases}$$

with $N_{engaged}^{threats}$ is the number of threat that can be engaged, the function $f\left(P_D\right)$ ensures that only defense strategies that engage the closest threats first are considered, $P_{\text{kill}}^{i}\left(P_D\right)$ is the probability that threat *i* is destroyed by the defense strategy P_D, and $P_{\text{kill}}^{\text{threat}}$ is the probability that the threat action destroy the ownship. The value of $P_{\text{kill}}^{\text{threat}}$ is provided later in section 5.2. The value of $P_{\text{kill}}^{\text{actions}}(i, j)$ is the probability that the threat *i* is destroyed by the defense action *j*; this value is provided later in section 5.1 for each type of action.

The threat utility function is selected to be the opposite functions $U_T = -U_D$ due to the choice of defense and threat. The utility functions being opposite functions, a zero-sum game is generated. The solution of the zero-sum game guarantees that the utility of the defense player has at least the value U_D^*

$$U_D^* = \max_{P_D \in S_D} \min_{P_T \in S_T} U\left(P_D, P_T\right), \qquad U\left(P_D, P_T\right) \equiv U_D\left(P_D, P_T\right)'$$

provided that the defense player adopts the game optimal defense strategy P_D^*

$$P_D^* = \arg\max_{P_D \in S_D} \left\{ \min_{P_T \in S_T} U\left(P_D, P_T\right) \right\}$$

4. GAME TREE GENERATION

The naval game tree (NGT) describes all the admissible sequence of threat responses and defense responses. The branching in the NGT is generated by the capabilities of the players and the uncertainties in the game, i.e., branches are created for:

a) each category of threat players and each category of defense players,
b) each admissible defense and threat bundle of actions, and
c) each possible outcomes of the transformation rules (when the outcome of a response is uncertain).

The set of admissible defense actions depends on the current situation and how this situation is transformed by the actions. For example, some defense actions are available only when a threat is between threshold ranges. The value of these threshold ranges may vary dynamically, differ for each type of defense action, and depend on the weather conditions, the weapon setup and status, the ownship course, etc.

4.1. List of actions and their characteristics

The set of actions of NGT is defined below. The ownship can apply several defense actions concurrently in presence of several threats. These concurrent defense actions are called "bundles". The set of admissible bundle of actions will be defined in section 8 with respect to specific nominal situations.

4.2. Defense actions of ownship and Time period for completion of the defense actions

It is assumed that all the defense actions of the ownship involve a sequence of three steps:

a) search and lock on target,
b) fire and target intercept, and

c) kill assessment.

It is also assumed that fixed time periods are required for completion of the search and lock step, and of the kill assessment step.

Three types of defense action are available: i) launch a long range surface-air missile (SAM), ii) shoot the medium range gun, and iii) shoot the short range gun.

Each defense action requires a period of time to elapse before completion. This time period, Δt_c, is calculated as

$$\Delta t_c = \Delta t_{\text{search}} + \Delta t_{\text{fire}} + \Delta t_{\text{kill}}$$

The characteristics of the defense actions and time periods are provided by the experts.

4.3. Threat actions

It is assumed that there is a single category of threats: an anti-ship missile (ASM). This ASM has a single available threat action: directly incoming toward the ownship. The characteristics of this threat action are:

- speed : $v_{\text{ASM}} = 850$ m/s
- trajectory : straight line
- probability of kill : $P_{\text{ASM}}^{\text{kill}} = 0.50$

4.4. Rules of engagement

The rules of engagement are employed to define the set of admissible bundles of defense actions. These bundles are function of the current situation.

Several of the engagement rules stem from the fact that the SAM and the medium range gun share the same STIR. There are only two STIR available to the SAM and gun. Each STIR can track only one target at a time.

The CIWS has its own independent STIR.

4.5. List of the bundles of defense actions

The set of admissible defense action bundles, B_D, is to be constructed online in three steps:

a) select the set of possible defense action bundles, B_D^{possible}, based on the number of threats,
b) by pruning the selected set, get a reduced set of defense action bundles admissible with respect to the position of the threats, $B_D^{\text{position}} \subset B_D^{\text{possible}}$, and
c) get the set if admissible defense action bundles by pruning again with respect to the current status of the weapons and STIR, $B_D \subset B_D^{\text{position}}$.

Admissible bundles of ***defense actions*** are provided by the ***tables*** 1,2 and 3 provided by ***experts*** (EDA):

1. *List of possible defense action bundles with respect to the number of threats. Notice that there are six types of actions associated with the medium range gun, i.e., one for each type of salvo*
2. *List of possible defense actions with respect to the position of a single threat. There is a total of eight possible actions: SAM, CIWS, and six types of gun's salvo. The set of possible defense actions depends on the range of the threat and on its azimuthal ($\theta = 0$ looking forward) and polar angles ($\phi = 90$ looking upward)*
3. *Possible defense actions with respect to the weapons and STIR status.*

4.6. Transformation rules

The transformation rules describe the outcome of the defense and threat actions.

In this scenario, the outcome of all the defense and threat actions is uncertain. The outcome can be one of two possibilities: (i) the opponent is destroyed, or (ii) the opponent survived. Hence,

e. Transformation rules of defense actions
    a. the threat is destroyed with probability P_{kill}, or
    b. the threat survived with probability $1 - P_{\text{kill}}$

f. Transformation rules of the threats' action
   i) the ownship is destroyed with probability P_{kill}, or
   ii) the ownship survived with probability $1 - P_{\text{kill}}$

4.7. Time discretization of the scenario

The simplified scenario requires decisions about actions at discrete point in time. The time interval between decisions varies with respect to the situation.
A decision is required in the following situations:

a) at the moment a threat crosses one of the threshold ranges, and
b) at the instant a currently engaged defense action gets completed, see section 0.

5. GENERATION of ALL POSSIBLE STRATEGIES

5.1. Algorithm of generation of NGT strategies (GNGTS) works in the following 8 steps:

1. Create lists L1, L2 of subtrees of NGT
2. Add root node of NGT to L1
3. For any tree Ti from L1 find the most left node K with the depth not exceeding given max H and allowing to add new nodes from NGT. If K exist go to the step 4 otherwise to 8
4. Generate list DK = [DK1, …, DKn} of all nodes incidental to K by the **Procedure of Relevant Actions** for situations of NGT. If DK is not empty, go to 5 otherwise to 3
5. For any DKj, j=1, n, Ti and K create a new tree NTj by adding DKj to the node K in Ti and add NTj to the list L2
6. Copy L2 to L1
7. Clean L2
8. End the work.

The Procedure of Relevant Actions for situations of NGT is based on the EDA tables 1-3 which filter the actions in the following 3 steps:

1. table 1 filters the set BS1 of all bundles {SAM_F, GUN_4S_F, SAM_B} having more than one weapon and common illuminator
2. table 2 filters the set BS2 of all bundles for single threats satisfied to conditions of their actual applications
3. table 3 filters the set BS3 of all bundles in accordance with the state of motion and STIR status.

BS3 comprise the set of permit table defense actions.
Let's prove that the above algorithm GNGTS provides all possible strategies of NGT.
5.2. Proposition. Given situation S and max depth H of analysis of strategies of NGT algorithm GNGTS guarantees generation of all strategies rooted in S.
Proof.
Preposition. *Given situation S and max depth H of analysis of the strategies of NGT the above algorithm GNGTS guarantees the generation of all NGT strategies rooted in S.*
Proof.
Let's prove by contradiction.
Assume there is a strategy Ti not generated by the algorithm and Ck= S, S1, …, Sk is a chain in Ti from S to Sk.
In accord with the step 4 of the algorithm all incidental to S nodes including S2 will be generated due the edge (S,S1) couldn't exist in the game tree NGT if (S,S1) is not legally generated action from the node S in the NGT.
Thus, the algorithm will generate an intermediate tree T1 with the chain C1=S, S1.
Analogically, for the node S1 the tree T2 will be generated with the chain C2=S, S1, S2.

Eventually, this reasoning brings to generating by the algorithm of the tree Tk with the chain Ck. Therefore, there is no chains of any strategy not generated by the algorithm what proves the preposition.

6. EXPERIMENTS

The results of experiments with searching the best strategy in the NGT for a single ownship are provided in the table below.
It appears that for the case when waiting conditions are ignored search time in the NGT for the best strategies is significantly less compared with ones for TABU system [18]
.

Initial data for the threats	Utility of the optimal defense strategy/ Probability of survival of the ownship)	Maximal deepness of the game tree search
$r(0) = 60$ km $\theta = 90^o$ $\phi = 45^o$	U = 0.4844	H = 4
$r_1(0) = 60$ km $r_2(0) = 70$ km $\theta_1 = 30^o$ $\theta_2 = 20^o$ $\phi_1 = 45^o$ $\phi_2 = 45^o$	U = 0.4793	H = 6
$r_1(0) = 65$ km $r_2(0) = 70$ km $\theta_1 = 90^o$ $\theta_2 = 30^o$ $\phi_1 = 40^o$ $\phi_2 = 150^o$	U = 0.4691	H = 6

Here is the listing of calculations for the second example of the table:
Navy system builds 1.0.1
Enter Treats count 2
Enter Treat1 distance in meters 60000
Enter Treat1 Fi angle degrees 45
Enter Treat1 Teta angle degrees 30
Enter Treat2 distance in meters 70000
Enter Treat2 Fi angle degrees 45
Enter Treat2 Teta angle degrees 20

Best Strategy Mark = 0.480043

Current Knot Util = 1
Start at time = 0
isa destroyed Treat1 = 0;
isa destroyed Treat2 = 0;
{SAM_F, NO_WEAPON}
Current Knot Util = 0.1875
Start at time = 55.1972
isa destroyed Treat1 = 0;
isa destroyed Treat2 = 0;

```
{SAM_F, NO_WEAPON}
      Current Knot Util = 0.
      Start at time = 69.808
      isa destroyed Treat1 =
      isa destroyed Treat2 =
      {NO_WEAPON, GUN_4S_F
            Current BRANCH
      isa destroyed Treat1 =
      isa destroyed Treat2 =
      Current Knot Util = 0.
      Start at time = 69.808
      isa destroyed Treat1 =
      isa destroyed Treat2 =
      {NO_WEAPON, GUN_4S_F
            Current BRANCH
      isa destroyed Treat1 =
      isa destroyed Treat2 =
Current Knot Util = 0.1875
Start at time = 55.1972
isa destroyed Treat1 = 0;
isa destroyed Treat2 = 0;
{SAM_F, NO_WEAPON}
      Current Knot Util = 0.
      Start at time = 69.808
      isa destroyed Treat1 =
      isa destroyed Treat2 =
      {NO_WEAPON, GUN_4S_F
            Current BRANCH
      isa destroyed Treat1 =
```

lic, 20-22 October 2003

7. CONCLUSION

Definition of a simplified naval defense scenario with multiple threats is considered.

Generation of the game three for the simplified single ownship scenario is provided overcoming the following main difficulties:

- The output of the defense actions is uncertain. There are two possible outputs with known realization probabilities,
- The time interval for completion of a defense action is different from one defense action to another,
- Several defense actions can be undertaken simultaneously, and
- A defense action can be undertaken while other defense actions are still ongoing.

The reactive planning solution is implemented to serve as a comparison baseline (to assess the benefits of deliberative planning).

It is shown that solution of the game tree for deliberative planning maximizes the probability of survival of the ownship with respect to the worst-case situation.

4. Scenarios with up to 8 threats are considered. Monte Carlo simulations are employed to statistically assess the benefits of the proposed deliberative planning.

References

[2] Chi, S.-D., Park, J.S., Jung, K.-C., Lee, J.-S.: Network "Security Modeling and Cyber Attack Simulation Methodology", Lecture Notes in Computer Science, Vol.2119 2001.

[3] V. Gorodetski, I. Kotenko, "Attacks against Computer Network: Formal Grammar Based Framework and Simulation" Tool. Proc. of the 5 Intern. Conf. "Recent Advances in Intrusion Detection", Lecture Notes in Computer Science, vol. 2516, Springer Verlag, pp.219-238, 2002.

[6] K. Ilgun, R.A. Kemmerer, P.A. Porras, "State Transition Analysis: A Rule-Based Intrusion Detection System", IEEE Trans. Software Eng. vol. 21, no. 3, Mar. 1995

[7] V. Paxson, Bro, "A System for Detecting Network Intruders in Real-Time, Proc", Seventh Usenix Security Symp., Usenix Assoc., Berkeley,

[12] E. Pogossian, A. Javadyan, "A Game Model For Effective Counteraction Against Computer Attacks In Intrusion Detection Systems", NATO ASI 2003, Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management, Tsahkadzor, Armenia, August 19-30, 2003, pp.30.

[13] E. Pogossian, A. Javadyan, E. Ivanyan, "Effective Discovery of Intrusion Protection Strategies" The International Workshop on Agents and Data Mining, St. Petersburg, Russia, 2005, Lecture Notes in Computer Science, Vol. 3505, pp. 263-274

[15] J. Couture, E. Menard, "Issues with Developing Situation and Threat Assessment Capabilities", Data Fusion Technologies for Harbour Protection, NATO Advanced Research Workshop, Tallinn, Estonia, June 27-July 1, 2005.

[16] E. Menard, J. Couture, "Application of Improved Threat Evaluation for Threat Assessment, Multisensor Data and Information Processing for Rapid and Robust Situation and Threat Assessment", NATO Advanced Study Institute, Albena, Bulgaria, 16-27 May, 2005.

[17] D.Boily, "Agent Based Resource Management Simulation", NATO Advanced Study Institute on "Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management", Tsakhkadzor, Armenia, 18-29 August 2003 (to be published by Kluwer Academic Publishers).

[18] D. Blodgett, M. Gendreau, F. Guertin, J.-Y. Potvin, Y. Seguin, “A Tabu Search Heuristic for Resource Management in Naval Warfare”, Centre for Research on Transportation, internal publication No.98-63.

[19] P. Bergeron, J. Couture, J.-R. Duquet, M. Macieszczak, M. Mayrand, "A New Knowledge-Based System for the Study of Situation and Threat Assessment in the Context of Naval Warfare" FUSION 98, Las Vegas, Vol II, pp.926-933, 6-9 July 1998.

[20] E. Shahbazian, B. Chalmers, P. Bergeron, J.-R. Duquet, "Situational Awareness in the Tactical, Air Environment", Paul Hall Center, Piney Point, Maryland, pp. 57-77, June 8 & 9, 1999,

[21] E. Shahbazian, L. Baril, E. Menard, G. Michaud, D. Turgeon, "Analysis of Adaptive Data Fusion Approaches within LM Canada’s Technology Demonstrator", NATO RTO Symposium in Prague, Czech Repub

Appendix 6: OJP in Communication

1.Arakelova E. and Pogossian E., "**Tools for testing and correction of the completeness of knowledge acquisition by autistic children**," pp. 159-165, Yerevan, Armenia, 2011.

ABSTRACT

While knowledge acquisition by children with ordinary abilities follows to traditions, personal experiment of parents and well-known standard methodologies development of unordinary children with positive and negative declines need extremely personalized to each child means.

We consider peculiarities and patterns of positive tutoring of autistic children (autistics) in knowledge acquisition and find that tools providing the complete tree of constituents required to acquire certain communalized knowledge followed by providing recommendations on learning the missed constituents can be supportive for any children.

The SSRGT Solver we develop provides advanced instruments to acquire expert knowledge and elaborate effective strategies for a spacious class of games. We introduce modules of the Solver relevant to tutoring and analyze ways to use them for knowledge acquisition by children.

Keywords: Tutoring, autism, chess, knowledge, meaning, acquisition.

1. INTRODUCTION

1.1. Autism became the object of research only 60 years ago, when the incidence of illness was 1:5000. The statistics was most perhaps underestimated due to health system's imperfect records, but the disease was considered extremely rare. Now, approximately 1 in 200 suffers from autism (90% of them are boys), and in some countries the statistics is even worse - in Great Britain it is 1 in 80. The growth of autism in the world is already comparable to an epidemic.

There are numerous researches on autism being now conducted in the world. The centre of these investigations, Autism Research Institute , http://en.wikipedia.org/wiki/Autism_Research_Institute, founded in 1967. The Institute annually holds conferences of scholars and doctors from all over the world, aimed at discussing the most recent achievements in this area. At one of the Conferences, there was a protocol signed "Defeat Autism Now!", which obligates doctors working on the problem, to apply all the known methods to cure autism, without awaiting the final results of their research, since they do not presuppose the use of medicines and are innocuous. In such cases, postponing of psychological and methodological aid aggravates the problem and makes it irreversible. **1.2.1.** Just 10 years ago, autism was considered an incurable disease, but now the "stories of success" appear from time to time. However, each of such victories is a sensation, proving to be rather an exception to the rule. And each victory requires a lot of efforts. Among these stories there are no memories like "we did nothing, and it was spontaneous recovery". All the researchers emphasize: the more effort parents and experts apply, the sooner the problem is identified, the better is the result. This brings us to the idea that one of effective ways to overcome "congenital asociality" is to develop new efficient methods, presupposing that a tutor (a parent or a teacher) thinks in the same way as his disciple, i.e. in the case of autism, teaching should not be based on conventional motivations, but rather on searching a stimulus significant for a child, be it even unclear for the rest.

1.2.2. According to another common opinions, the essence of the problem, is the dysfunction of "mirror" neurons, which play a key role in the processes of imitation and language learning [from the third culture: http://www.edge.org/3rd_culture/rama/rama_p1.html].

According to this theory, the dysfunction of "mirror" neurons results in the prevention from understanding the formula *"I am being my body"*, i.e. from the identification of oneself with his body.
1.2.3. Motivation is an integral part of human's social nature; this is social adaptation, based on social values; this is also one's desire to attract attention to him, to express himself, to "hear and be heard by others", i.e. to create interpersonal relations. All this can be based exclusively on the adaptive abilities of the psychics, on the ability to perceive and to analyze, determined by "motivators-correctors." Thus, the real problem of teaching autistic children is not their low cognitive abilities, but lack of motivation for the process of cognition.

Prima facie, parallels with children suffering from other disorders of sensory areas - like blind and deaf children - may seem to be relevant, since their training is also very problematic. However, on closer examination we see that the isolation of latter children is forced, they have motivations, and only complete isolation can bring to the decay of psychics and degradation.

1.3. Our work is based on the experience of forming certain skills sufficient for the study in the secondary school, with a girl, T., with the diagnose of "early infantile autism".

At an early age she was diagnosed with Kanner Syndrome classified as F 84.0. according to the ICD-10 scale (ICD-10) (http://apps.who.int/classifications/apps/icd/icd10online/). Kanner's syndrome is one of the five common (Pervasive) developmental disorders, sometimes called a form of highly functional autism (that is, a mild form of autism, in which the ability to socialize is relatively preserved). Often, persons with Kanner syndrome have ordinary or even high order of intelligence, but they are marked with non-standard or poorly developed social skills, which results in late emotional and social development, as well as problematic integration.

Through years of training, a girl has been brought up to a certain level of socialization and learning; now she attends the 3d level of the secondary school. Her mother language is Russian. At school, her achievement grades are satisfactory, but behavior is asocial.

1.4. The analysis of obstacles one of the authors faced and overcome during10 years of training of T., as well as those on the agenda yet, united with the experience in computer aided knowledge acquisition [8-15] motivated us to design a tool assisting in overcoming of some regular omissions in training of autistic children. First, we present a brief description of the stages of the chronological development of T. versus those of the development of an ordinary child, with an interpretation of reasons of the deviations. Then we discuss our view on knowledge hierarchies and their presentation. Finally, the tools aimed to accumulate stabilized views and methodologies on knowledge acquisition and assisting in revealing and correcting the constituents required to acquire certain communalized knowledge are presented.

2. PECULARITIES OF TUTORING AUTISTICS

Below we are trying to briefly present the behavioral deviations of T. from her early age to the present, as well as to analyze their causes and to establish a causal connection between them and learning abilities.

Here is the description of the child's development (from birth to the age of 10 years).

1-3 months: with normal physical development, the child is somewhat tense, often holding hands clenched in fists; she does not express her emotions aloud, avoids eye contact, does not use facial expressions, does not imitate the mood of the elders, does not like to be in somebody's hand, unless being strongly shaken , does not like rattles; toys attracts her solely because of bright colors; she is irritated with talks, soothed by casteless melodies; overexcitable, she has a dog-sleep, screams a lot (but no crying!), cannot stand loud vibrating sounds, as well as "synthetic" sounds of electronic toys; outdoor she behaves more quiet. She likes to take a bath.

4-6 months: physical development is normal, tension persists; the girl avoids all contacts, does not react or reacts negatively to strangers, but on the other hand, can easily stays with any person. She never tries to draw attention to herself. In the presence of mother, she prefers to stay with her. There are problems with additional feeding - the child denies everything new including food, while they try to feed her from a bottle or a spoon; a strange taste is rejected in every way, up to vomiting. Growing fear of sounds, "synthetic" tunes, voices. She can weep/scream all day long. Her look is for a long time concentrated at one and the same subject, often at the street-lamp. She likes long walks. From early age, the child is accustomed to swimming in open water.

7- 9 months: a little behind in growth due to malnutrition, does not drink water, does not accept anything but the breast milk; she recognizes close relatives, but does not react to their presence, nor does she try to talk or attract the adults' attention; she is not interested in toys; never asks for anything; the pointing gesture is absent; the child hardly smiles, and cries almost around-the-clock. She likes to go outdoors and to stay in water.

months 10-12: the girl in behind in growth and weight due to malnutrition, does not drink water, eats only breast milk; she feels weak, at the same time stress persists; the child stands, supported by somebody's hand and walks holding furniture; she mainly walks on her toes. She neither talks, nor plays with toys; while playing with blocks she does not builds them vertically, but horizontally, and sometimes some cubes are replaced by small items or toys. The child often screams or no apparent reason; she sleeps light, and sometimes screams even during sleep. She likes to walk outdoors. She demonstrates no preferences. She is (physiologically) attached to her mother, and she goes to her grandfather's hands with obvious pleasure. At this period, fear is growing, "a street-lamp" becomes a fixed-idea. She looks through book without interest, as if she does not understand pictures. The girl pronounces the word "mother".

months 13-15: by the 14th month, breast-milk is replaced with alternative feeding; the child begins to walk with confidence; her movements are faster and clear. There is no emotional background, she does not copy adults' mimics. Entranced, she listens to boost advertisements or children's songs. She still has phobia for objects, an attempt to play hide and seek results in another fear, her attitude to sounds is even more painful. Electrical gadgets are never used, since even their names ("a vacuum cleaner, a hair dryer") provokes psychosis, crying, closing ears with hands. She can stay alone at her room for a long time, sometimes covering herself with a blanket over her head. She looks through pictures and books without much interest, but she does not mind listening to somebody's reading at bedtime (particularly poetry). She likes to walk outdoors and to be in the water. She has no self-service skills: she cannot hold a spoon, as well as cannot stay dry. Finally, she is able to hold a bottle with cereal and to drink from a cup, but she holds them with diffidence.

months 16-18: the physical state is normal, the child walks with confidence. She does not realize the danger of height, standing on the edge, cars, etc. She does not talk, nor does she play social games. She can stay alone in her room for a long time (avoid!). Trying to point to an object, she uses her mother's hand instead of her own. She does not communicate with anyone. She knows her name, but does not react – neither verbally, nor with her look. She likes to walk outdoors and to swim. She does not follow instructions. She stays on somebody's hands without pleasure, struggling out of hands without realizing the threat to fall down.

months 19-21: hereinafter – physical development and macro-motorics are normal. No significant changes. Panic before the elevator (persists up to nowadays).

months 22-24: it becomes noticeable that the child does not like small enclosed rooms (without windows); she shows no interest towards children, games, as well as animals; she neither follows instruction, nor knuckle under to the adults' demands. But it does not look like a protest, but rather as a result of incomprehension of what is happening around (due to incomprehension of speech).

months 25-30: unconcealed anxiety becomes apparent if the daily route or the established order of actions is changed. General anxiety scales up. She does not like to draw, does not take a pencil by herself. The girl forwards a pencil to mother so that she would draw, but at the same time, the girl does not point to the desired object of drawing. She easily distinguishes colors. She can show an object of a certain color if asked to, but almost does not use a pointing gesture. Fears persist. There are some elements of unconscious speech (echolalia). It is obvious that the child understands the names of many items, but not verbs. She practically does not play with children. She does not perform mirror movements (she waves with her hand, holding it with its back side to herself), cannot repeat a simple physical exercise in a reflecting manner.

months 31-36: the child does not understand human behavior, the logics of what is happening; children's cry is fearsome for her, she covers his ears, shivering all over. She can listen to the same music for many times. Speech is not progressing. She answers a question with the same question and intonation: "[Do you] want to drink?" - "Want to drink?"). The child never complains. There are problems with public transport (small and crowded space); she prefers big buses and definitely likes trolleybuses and metro.

From the age of 2 to 6 years the child attends kindergarten. She does not participate in games. She is scared by some songs at music lessons. However, at the same time, during the first weeks in the kindergartens, the child acquires certain skills of self-service. She stops using diapers, and learns to carry a spoon. Two years later she recites her first poem at the children's show. The kindergartener notes that along with the difficulties of upbringing (asocial behavior, aggression - though rare), the child demonstrates good memory, learns poems better than the rest of the children, can count, etc. By the age of 6, T. can read and write in block letters. She can add small numbers, but subtraction is problematic. The actions are performed only with illustrative material.

At the age of 6, T. entered a secondary school, where the teaching language was Armenian. This fact balked the progress of the Russian-speaking girl, and she had to leave the school four months later. The next year, she entered the school, in which the teaching was in Russian, where she has studies for three years with varying success, but rather steadily (primarily thanks to the professionalism of the teacher.)

3. REVEALING TIPS OF TUTORING

3.1. Typifying Difficulties.

• The child does not feel responsibility (obligations), but during the 1st year of study, following the example of classmates, and to be more precise, as a result of praises and reproaches, the child gets used to the fact of the compulsory every day work. Finally comes the first ideas of responsibility, as well as of knowledge assessment.

- Difficulties in performing elementary mathematic actions. Addition is relatively easy. Subtraction is problematic since it requires the understanding of such a notion as the composition of a number. Since the verb "consists of" rather belongs to the abstract verbs, which do not denote an action in its full sense, subtraction as an action represents certain difficulties. Probably, it is difficult for T. to manipulate with symbols, but she relatively easily remembers schemes. That is why we have introduced the notion of a compound as "the difference between the minuend and the second compound". It is quite obvious that this looks like a vicious circle. Another possibility to expand a number in compounds is to take any number less than the initial one, and to look for the second component unless their sum gives the initial number.
- Simultaneously, the problem of understanding of "the whole and a part" can be discussed.

3.2. On the Causes of Difficulties.

The initial causes of such problems, particularly the dispersion of the attention of autistics, appear to be the violations of certain factors of perception [3, 4, 7].

Those factors include, particularly, the following external and internal ones.

External: size, the intensity (on the physical or emotional levels), contrast (contradictions with the environment), movement, recurrence, novelty and recognition.

Internal

- The directive of perception – expectation to see what is supposed to be seen a posteriori
- Needs / motivation – a person sees what he needs or what he considers important. (In the case of T., certain basic needs and relevant motivations are obviously expressed (hunger, thirst), whereas the other (the need for contacts, etc.) are weak or absent. This is the reason why the development of all the subsequent levels of motivations occurs at least with deviations.
- Experience; a person perceives that aspect of the stimulus, which he has been taught by experience.
- *Self*-concept – the perception of the world is built up around self-perception. (T. reacts to a child's cry in a strange manner: she covers her ears with her arms, and tries to escape. She cannot relate somebody's crying with her own feelings, and approaches it as an unclear and thus, dangerous event. Approximately in the same way she reacts to any loud speech, even if it does not sound aggressive.

Three mechanisms of the selectivity of perception are explained as the following:

- The principle of resonance – individuals corresponding to a person's needs and values, are accepted more quickly than those who do not correspond.
- The principle of protection – something resisting a persons' expectations is perceived as worse.
- The principle of caution – something which is threatening to a person's psychics is more quickly recognized.

From the above we see that the internal factors of perception of autistics can form a biased or distorted final image was built as a result of perception. The reasons are, we think, in the absence or deformation of the self-concept, and also due to violations of the needs and motivations.

3.3. The following **tips have a positive impact in tutoring** T. and are consistent with ones recommended by Temple Grandin, Ph.D., former autistic and these days Assistant Professor of Colorado State University [3]:

- Steady recurrence of the exercises in acquiring any knowledge
- Awareness of the methodologies of training of ordinary children along with their continuous creative adaptation to a particular autistic
- Awareness on the permanent dispersion of the attention of autistics for regular return it back to the subject of teaching, using in parallel additional activities like swinging [3].

3.4. Tips of positive tutoring induce certain **niches for computer aid** and corresponding requirements to
the performance of software including the following ones:

- accumulating communalized knowledge, following the stages by Piaget [5], needed for development of children both ordinary and with declines, in general and for selected areas of human activity
- learning personalized knowledge of experts
- testing completeness of knowledge of selected types acquisition
- preserving steadiness in application of tutoring tips including the recurrence of the exercises
- adapting acquisition to the peculiarities of children
- coping with the dispersion of attention, particularly by visualizing interactions and using attractions.

Software that meets these requirements can support tutoring of children both ordinary and with declines. In what follows the progress in applications of a kind of knowledge processing software to tutoring is reported.

4. KNOWLEDGE PROCESSING BY SSRGT SOLVER

4.1. Problems. We study optimal strategy provision problems for a class where the *space* of hypothesis of *solutions* can be specified by *reproducible game trees* (SSRGT) which meet the following requirements [8, 12, 13]:

- to be given (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions in the (c) specified moments of time and (d) specified types of situations
- to be identified benefits for each of the actors

- to be a positional game, i.e. the game tree be reproducible from any given position.

Many competitive decision making and dialogue problems are presented by the class including combinatorial chess like games, diagnostic, marketing, intrusion protection, etc.[8, 12] **4.2.The Solver** of the SSRGT problems is a package [8, 14] aimed to acquire strategic expert knowledge to become comparable with a human in solving hard problems. In fact, the following three tasks of knowledge acquisition can be identified in the process:

- construction of the package of programs *sufficient* to acquire the meanings of units of problems vocabulary (UV)
- construction of procedures for *regular* acquisition of the meanings of UV by the package
- provision of means *measuring the effectivenes* of solutions of SSRGT problems.

We formulate the limitations in designing effective package as following:

- be able to store typical categories of communalized knowledge as well as the personalized one and depend on them in strategy formation
- be able to test approximate knowledge-based hypothesis on strategies in questioned situation by reliable means, for example, using game tree search techniques.

The second task of *complex expert knowledge acquisition* was planned to be solved in the following two stages:

- *to prove the sufficiency*, i.e. that the package, in principle, can acquire the meanings of UV used by chess experts and allow to tune them properly to solve expert knowledge intensive chess problems
- *to ensure regularity,* i.e. to develop procedures for regular acquisition of the meanings of UV as well as any SSRGT problem.

Fig.1 presents the flow chart of the pilot version of the Solver implemented in Java [13, 14].

The main modules of the Solver are:

- *Personalized Planning and Integrated Testing* (PPIT) program is the central module of the Solver and is composed from the following basic units: Knowledge Base (KB), Reducing Hopeless Plans (RHP) , Choosing Plans with Max Utility (CPMU) , Generating Moves by a Plan (GMP). Given a questioned position P1 and a store of plans in KB, RHP recommends to CPMU a list L1 of plans promising by some not necessary proved reasons to be analyzed in P1. RHP relies on the hierarchies, classifications, etc. of knowledge in KB and provides the most perspective plans for further analysis of P1. If KB is rich and P1 is identified properly it can provide a ready-to-use portion of knowledge to direct further game playing process by GMP unit. Otherwise, RHP, realizing a reduced version of CPMU, identifies L1 and passes the control to CPMU.
- *Concepts Manager* is responsible for storing, organizing and managing expert knowledge represented in the form of concepts.
- *Percepts Manager* is responsible for storing, organizing and managing the percepts, inputs of the preceptors of the Solver [13].
- *Graphical User Interface* provides a web based graphical interface that allows users of the package visually configure the package, insert an arbitrary problem from the SSRGT class and organize knowledge insertion process in the regular way by using visual instruments including the concepts representation graphical language.

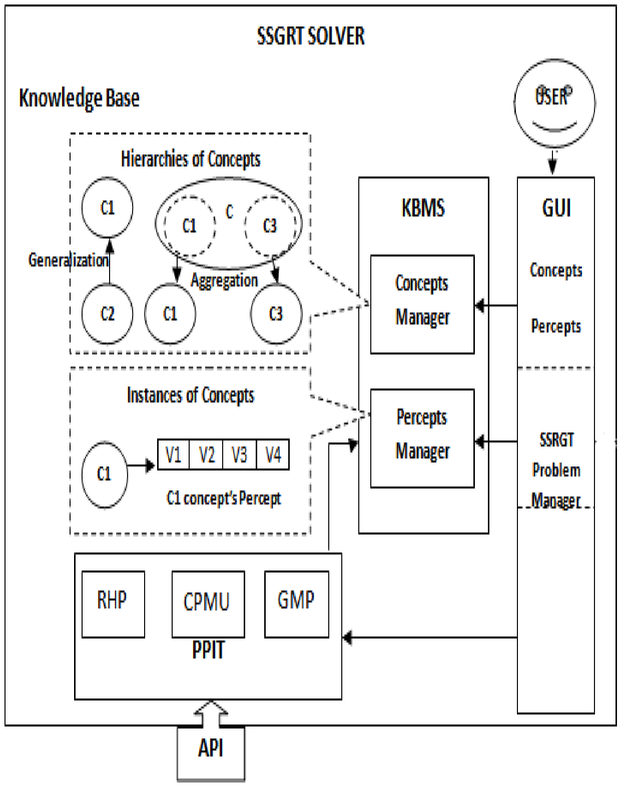


Fig.1.The Solver.

4.3.Sufficiency of the Solver for acquiring and effectively involving complex expert knowledge in decisions making was demonstrated by experiments in acquisition of knowledge for two etudes which are intractable for common chess programs -- the Reti and Nadareishvili etudes suggested by Botvinnik to test the quality of knowledge intensive solutions in computer chess [8]. Particularly, Nodareshvili etude can be solved only in the case of 36 depth of search in the game tree while chess programs in average search for the depth 6. The assertions on sufficiency of Solver for acquisition of communalized and personalized strategy knowledge are given in the next chapter.

4.4. Regularity of knowledge acquisition was ensured by developing the graphical language interface [14, 15]. Instructions for operations with problems of the SSRGT class are formulated in a unified, not problem depended form allowing the specification and communalized knowledge of particular problems step by step to input into the Solver.

4.5. Effectiveness of the Solver was successfully tested for the network intrusion protection problems by using corresponding expert rules and goals [12]. Experiments on protection against representatives of four classes of attacks: SYN-Flood, Fraggle, Smurf and Login-bomb allow formulating, in particular, the following statements:

- The number of nodes searched by the IGAF2 version of the Solver for making decisions with all expert rules and subgoals is the smallest compared with the IGAF1 version of the Solver with the minimax algorithm in which the depth of the search is increasing up to 13
- The recommended version of the IGAF2 with all expert rules and subgoals, for the depth of search 5 and 200 defending steps, is outperforming the productivity of the minmax algorithm by 14% while using 6 times less computing time and searching 27 times fewer nodes of the tree.

Local tournament-based methodology for measuring effectiveness of Solver for SSRGT problems are presented in [9, 19].

5.TUTORING KNOWLEDGE ACUSITION BY SOLVER

5.1. Chess Concepts. The consequent approach to chess-based training is induced by the models of cognition in [8, 9, 15, 16] and their modules realized by Solver.

The approach starts from the following two **assertions**:

- Communalized meanings of units of chess vocabulary (chess UV) can be equally presented by Solver
- Solver is able to learn personalized expert meanings of chess UV.

5.1.1. Assertion 1 is argued as the following.

Following the model in [16], *knowledge of the Solver* comprises the *abstracts*, or abstract classes, say in Java, the *percepts*, or the outputs of preceptors, realized in the Solver by the user interface, the constituents and compositions of the abstracts along with any presentation of the above units.

The *meaning* of an abstract A unites the abstracts and percepts able to contribute to activation of A united along with activation the abstracts of which A in turn contributes.

Percepts activate abstracts of the first layer and hence ground them to the realities [16, 17], which in our case are the positions on the chess board. Abstracts of each next higher layer are activated by the abstracts of the lower ones.

Chess UV correspond to *classifiers,* or concepts*,* which are a kind of abstracts. Articulation of some chess UV x by the interface causes activation of corresponding classifier x which, in turn, activates the meaning mx of x. Hence, the Solver can address down (implement parsing) to any consequent layers of abstracts that contribute to the activation of x or address up (parenting) to activate abstracts that could have been contributed by x.

By necessity Solver can address to the percepts which activate x as well as ground those percepts to the board.

Assuming the above multi-layer *stratification*, i.e. parsing and parenting, of meanings of classifiers is modeling corresponding stratification of meanings by experts the question of equal stratification of a chess UV x by Solver is reduced to the question of equal presentation of mx in Solver.

The question is positively resolved for communalized meanings which, as a rule, can be specified and therefore modeled.

Let's emphasize that communalized meanings of chess UV include the possibility of their grounding which is intractable problem, yet, for the majority of known today models of the fragments of natural languages but chess ontology differs from them providing advantages with respect to others.

5.1.2. Justification of the **assertion 2** succeeds from the following findings in [11]:

- real implementation of the meanings of chess UV both for experts and computers, in principle, can only approximate the specification of winning game tree structures due to irresistible complexity of computations to prove correctness of the models. Thus, it follows that
- both chess players and computers form, as a rule, different meanings on realities essentially dependent from their personal experience
- any preferences between meanings of chess players cannot be completely justified
- personalized constituents in the meanings of UV chess are, as a rule, inevitable.

Knowledge by [16] comprise regularities in achieving certain utilities. Particularly, the utilities of chess classifiers are induced by the utility to be winning. If a position P* in chess, has a certain utility u(P*) the new positions P with similar utilities can be united to P* by abstraction or other learning procedures.

Learning procedures look for regularities causing u(P*), find positions where those regularities take place and can preserve in P utilities similar with u(P*). The positions P of the first circle can be the base for the second circle expansion, etc.

Typical rules in forming chains of positions with similar utilities use, particularly, the following ideas:
- any P has utility u(P) as close to u(P*) as many values of attributes of P coincide with the ones for P*

- as many positions of the ongoing classifier of P* have the same values of attributes of P* so great is the likelihood that a new position with the same values will belong to the same class.

5.1.3. There are several methods to form meanings of realities by their training sets which include methods to form partial Boolean functions, machine learning and statistics-based methods, the Beckon-Mile induction, the G. Polya's plausible reasoning, etc. If the training realities are unique it is appropriate to use methods which generate new realities while preserving the base utilities.

While communalized meanings can be specified and hence tested in hard but common ways the personalized ones need additional efforts to test that expert meanings were learned properly.

We develop the following tests:

- generation by the models of meanings positions on the board to be commented by the experts for possible corrections of the models
- using complementary means to minimize the threat that only a part of the meanings of the expert classifiers were "caught" by the models like the following tests:

* experts generate examples which are tested by the models
* known positions that meet certain criteria of completeness are taken from some repository to compare expert responses with the model ones.

5.1.4. Testing of acquisition of concepts can be done by a variety of strategies like the one where tutors sequentially decompose given concept A, for each decomposition reveal not leaned units, each of them decompose to find new unlearned components, etc. and, finally, composing the map of acquisition of A layer by layer move up while achieving the learning of each missed component of the layers.

5.1.5. The experiments in knowledge acquisition are based on the repository of chess knowledge, a comprehensive chess ontology of the communalized meanings of UV chess [9, 10].

About 300 units of the repository are classified in accordance with chess traditions for systematic analysis and modeling. The languages of the set theory and finite logic of predicate calculus with type variables are used for specification the UV chess.

Starting from basic concepts of chess definition the ontology is built as a hierarchy where each consequent layer of concepts includes concepts composed from subordinate ones supplemented bysome new attributes and procedures.

Up to 4 layers including about 100 concepts were realized. For example, the 4th layer includes, particularly the concepts "Threats to take a piece", "Defense of X by piece Y", "The square X is overloaded", "Double attack ", etc.

Successive formation of layers revealed an essential growth of the role of personalized constituents in the meanings with rising the layers as well as possible increasing influence of the factors like belief, commitments, traditions both parties have, etc. [18]. The ongoing development of the Solver is aimed to resolve those questions.

6. EXPANDING THE DIMENSIONS OF TUTORING

6.1. We consider the ways of transition of our chess knowledge acquisition experience to **tutoring basics of Math.** By analogy with the chess the following questions are on the agenda:

- UNL base ontology for pre and basics of Math
- orderings of Math concepts
- programs for diagnosis of completeness of Math concepts acquisition
- programs for processing tutors' recommendations and corrections.

We are going to rely on UNL instruments for presenting basics of Math knowledge since at present they acceptably cover the essentials in English and are available to be used.

Math basics is supposed to be presented by the tree of development divided into stages and states of the progress which by values of certain attributes specify for each state the required knowledge for normal development, recommendations and means coming from psychologists and other experts for correcting the declines from the normal one.

The attributes and their values for normal development and declines are defined by experts.
The tree has to be able to grow dynamically with providing new knowledge on the child development, attributes of declines, methods for corrections, etc. without deterioration of the former structures. The corresponding tool is supposed to be implemented by the client server principle.

6.2. Tutoring Pre-Chess Knowledge. Early chess playing accelerates the development and essentially contributes to the characters of children. It is acknowledged that chess players become goal oriented and persistent in achieving them, attentive and competitive, disciplined, impartial and assiduous, can compare the options, plan and foresee the results of the planning. They have a good memory and imagination, are quick-witted, resourceful and can operate in time limits. Players enjoy creativeness and, cetera par bum, are excellent in learning school subjects, specially, Math [2].

Germany, Czech Republic, Hungary, these days and Armenia, teach chess in schools.
Unfortunately, ordinary children are able to start to learn chess when they are about 7 years old. Only in a few cases they learn earlier and, as a rule, become extraordinary players. So, Casablanca, Karpov, Keres and few others had learned chess at the age of 4 years, Stainec, Eave in 5, Kasparov, Fisher, Spasski in 6, Alechin, Tal, Smislov started learning chess when were about 7 years old.

One of the essential reasons of late learning of chess rules can be explained by their being of the abstracts of high layers. Indeed, recall that even for starting operations on the chess board kids need to be acquitted, at least, with the concepts of colors, variety of lines: horizontal, vertical, diagonal, squires and relationships between them, pieces and their legal moves, etc. Each of these concepts has its own line of learning and kids come to them only when certain concepts of the lower layers that have been already learned.

Many models aim to explain the process of learning of those early concepts but difficulties in pattern formation and recognition indicate that the problem is far from proper solution, yet.
Pre chess knowledge greatly intersects with one for the basics of Math what allows bringing certain elements of experience of its tutoring to chess.

Particularly, chess tutoring like for Math is possible only for kids with proper level of knowledge on space, quantity, magnitude, is size based, especially with respect to development of a proper level of sensory-motor, orientation in space, coordination the motions, language etc. abilities. The main aspects of that step by step development, specially the aspects of cognition, are analyzed to be included into diagnostic schemes and software.

6.3. Strategies for testing the completeness of acquisition of some amount of knowledge can be produced by the Solver as the following. Let's consider some knowledge K that can be split into classes and selected members of the classes, say concepts, A= {A1, …, Ak}, acceptably present those classes in the way that the measures of completeness of acquisition of A properly estimate acquisition of K.

Tutors test how well students learn A in a time restricted dialogue by asking a concept from A , grading the answer, choosing the new question from A , grading it, etc. and depended from sum of the grades and the time spent for answering put the final grade, say in the scale 0, ½, 1.

Assuming students' possible answers comprise a set B of certain units the dialogue induces certain game tree where particular games correspond to the question-answer sequences while the terminal nodes can get 0,1/2,1 value. Tutors aim to estimate the amount of not learned knowledge of the students while the latter try to get max grades. Thus, strategies of tutors can be valued, for example, by the sum of final grades of terminal nodes and the best tutor strategy minimize that sum.
Assuming the questions and answers reciprocally determine each other, the testing game can be classified as a positional game of the SSRGT class and, therefore, relevant to application of knowledge-based strategy processing by the Solver [8-16].

6.4. The Solver provides an advanced platform for **tutoring SRGT strategies** which is convenient to present in the context of an advanced software in the field, UMRAO, developed in the Laboratory of Advanced Educational Systems [1]. The authors find that while most research in computer chess is focused on creating an excellent chess player a relatively little concern is given to modeling of how humans create strategies to play chess. They investigate knowledge-based chess in the context of building a prototype chess tutor, UMRAO, which helps students learn how to play bishop-pawn endgames. They argue that it is essential to follow in tutoring to a knowledge-based approach, since students must learn how to manipulate strategic concepts and not how to carry out minimax search. UMRAO uses an extension of Michic's advice language to represent expert and novice chess plans. For any given endgame the system is able to compile the plans into a strategy graph, which elaborates strategies (both well-formed and ill-formed) that students might use as they solve the endgame problem. Strategy graphs can be compiled "off-line" so that they can be used in real time tutoring. It was shown that the normally rigid "model tracing" tutoring paradigm can be used in a flexible way in this domain. Like the Solver UMRAO presents chess knowledge and use high level concepts such as plans and goals for elaboration of strategies. Comparing strategies of students with the expert ones it consults the players how to improve the performance. The explanations are given in chess terminology and are personalized.

The Solver can extend the above tutoring at least in the following:

- considering not only endgames but arbitrary positions including the complex middle game ones
- creating plans automatically like the ones demonstrated in solving Reti and Nadareishvili etudes
- tutoring learning the strategies of important for applications SSRGT problems, particularly the intrusion protection, defense, management ones.

7.CONCLUSION

The peculiarities of 10 years of successful training of an autistic child who is now the 4th year student of an ordinary school are considered and some patterns of positive tutoring of children both autistic and ordinary are extracted to be realized by the SSRGT Solver.

The architecture of SSRGT Solver and functions of its units are presented to make comprehended the advances in formation of knowledge-based strategies, their applications to solving chess like problems of the SSRGT class and perspectives of their use for tutoring.

The successive approaches to chess-based tutoring and testing are discussed induced by the models of cognition and modules of their realization by the Solver are presented.

The assertions on conditions of positive outcomes of tutoring of the communalized and personalized meanings of chess concepts are argued and the ways of their implementation by the Solver are considered. Finally, the perspectives of development of the Solver for tutoring of basics of Math, pre chess, testing and tutoring strategy knowledge are discussed.

ACKNOWLEDGEMENT

We would like to express our gratitude to V. Arakelova and N. Ajabyan for valuable advises and fruitful collaboration in the process of our work and preparation of this paper. We thank S. Aghababyan and V. Tadevosyan for involvement in our research and wish to continue it in their master degree diplomas.

REFERENCES

[1] D. Gadwal, J. Greer, G. McCalla. Tutoring Bishop-Pawns Endgames: An Experiment of Using Knowledge-Based Chess as a Domain for Intelligent Tutoring. University of Saskatchewan, Canada, 1992, preprint S7N0W0, p.34

[2] V. Grishin. Kids Play Chess, Moscow, "Prosveschenie", 1991 (in Russian).

[3] Temple Grandin, Tutoring Tips for Autistics, http://www.cdrcp.com/autism/teaching-tips

[4] O.N. Gromov, G.R. Latfullin. Organizational Behavior, St. Petersburg. Peter, 2008. - 432.

[5] J. Flavell. The Developmental Psychology of Jean Piaget, D. VanNostrand Comp. Inc., Princeton, N.J., 1962

[6] J. Furnkranz. Machine Learning in Games: A Survey in “Machines that Learn to Play Games”, Nova Sci. ,2001
[7] O.S. Nikolskaya, E.P. Babenskaya, M.M. Libling. Autistic children: ways to help. Moscow, 1997.
[8] E. Pogossian, V. Vahradyan and A. Grigoryan. On Competing Agents Consistent with Expert Knowledge, Lecture Notes in Computer Science, AIS-ADM-07: The International Workshop on Autonomous Intelligent Systems - Agents and Data Mining, St. Petersburg, Russia, 229-241pp, June 6-7, 2007.
[9] E. Pogossian. Adaptation of Combinatorial Algorithms. Acad.emy of Sci. of Armenia, Y., 293, 1983 (in Russian)
[10] Pogossian E., Hambartsumyan M., Harutunyan Y.: A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations. Academy of Sciences of Armenia, IPIA, reports 1974-1980, 1-55, in Russian
[11] E. Pogossian. Specifying Personalized Expertise. International Association for Development of the Information Society (IADIS): International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006), 8-10 Dec., Barcelona, Spain (2006) 151-159
[12] E. Pogossian, A. Javadyan, E. Ivanyan: Effective Discovery of Intrusion Protection Strategies. The Intern. Workshop on Agents and Data Mining, St. Petersburg, Russia, Lecture Notes in Computer Science, Vol. 3505 (2005) 263-274
[13] K. Chachatryan, V. Vagradyan. Graphical Language Interpreter Unified for SSRGT Problems and Relevant Complex Knowledge, see the Proceedings of CSIT2011, 5p
[14] Z. Naghashyan, E. Pogossian. Developing Java Software for Representation, Acquisition and Management of Strategy Knowledge", Mathematical Problems of Computer Sciences, Proceedings of IPIA, Yerevan, 187-195, 2010
[15] E. Pogossian. On Measurable Models of Promotion of Negentroping Strategies by Cognition, New Trends in Information Technology, Sofia, 161-169,2010
[16] E. Pogossian. On Modeling Cognition. See theProceedings of CSIT 2011
[17] D. Roy. Grounding Language in the World: Signs, Schemas, and Meaning Cognitive Machines Group, The Media Laboratory, MIT http://www.media.mit.edu/cogmac / projects. html, 1-27, 2005
[18] T. Winograd, F. Flores. Understanding Computers and Cognition. A new foundation for design Publishers, Huntington, NY, 1986.
[19] E. Danielyan, E. Pogossian E. On Expanding of Method of Local Tournaments on Evaluation of Management Strategies, Proceedings of SEUA, 6p 2009.

2. Pogossian E, Grigoryan S. and Berberyan L., **Personalized Interactive Tutoring in Chess**, Didactics in Chess, London Chess Conference, December 10-13, 2016, pp.

1. **Tutoring in Chess**. With tremendous advances in computers personalized interactive tutoring of students and testing their advances in learning new knowledge becomes possible.

Particularly, while knowledge acquisition by children with ordinary abilities follows to traditions, personal experiment of parents and well-known standard methodologies development of unordinary children with positive and negative declines need extremely personalized to each child means.

In [1] were considered the peculiarities and patterns of RGT Solvers [2-4] based positive tutoring of autistic children (autistics) in knowledge acquisition and it was found that tools providing the complete tree of constituents required to acquire certain communalized knowledge followed by providing recommendations on learning the missed constituents can be supportive for any children.

RGT, Reproducible Game Trees, class includes chess and chess-like games, certain types of teaching and communication problems, applications, such as computer networks intrusion protection, marketing strategy elaboration in competitive environments and defense of military units from a variety types of attacks [2, 4].

Since expert approaches in solving RGT problems stay the most effective, we develop knowledge-based solvers of RGT problems comparable with experts by their effectiveness. And because testing of chess knowledge of students can be interpreted as RGT problems their Solvers can be used as advanced instruments for acquisition of chess expert knowledge and elaboration of effective strategies of its testing. In what follows we present our experiments in explaining classifiers and strategies of chess experts to students.

2. Explanation of Chess Classifiers. We present chess knowledge by graphs with nodes corresponded to the units of chess knowledge. The edges of graphs present relationships and regularities. Thus, “CheckMate” concept is presented by a node that identifies the concept and is linked with “King can’t escape”, “King under check”, and “King has no defense” nodes as the attributes. The algorithm brings up the name of the concept, relations and regularities, also certain examples when tutoring. For each of explained node the relations, regularities can be extracted and explained.

3. Explanation of Strategy Searching. Plans, which are certain common descriptions of strategies, are explained by their constituent goals sorted by priorities, where each goal is a composition of chess concepts definition precondition and postcondition of the goal, the depth of tree to search for the goal from the initial situation and criteria to evaluate how good the goal is achieved. Description of plan is done by explaining each goal of plan and stating its priority, for each goal its preconditional and postconditional chess concept nodes are explained as described above.

4. Testing of Learning. For each strategy related chess knowledge Solver suggests ways for testing the learning. Solutions of students are compared with ones of Solvers and if they do not match to each other, what means students did not understand the explained plans or concepts correctly, the correct solution is demonstrated, the tutoring knowledge is explained again and a new problem for the same knowledge is suggested to students until satisfactory learning.

5. For Measuring the Progress in Learning Solvers use tournaments-based scales similar to ones of Elo ratings. Corresponding tool lets to handle chess games between specific person as user and chess engines, between two users and between two chess engines, thus making possible rate the students according to their performance in chess games and tournaments organized by the tool.

6. Adequacy of Tutoring was confirmed by experimenting in endgames. For example, the winning strategy in Rock against King includes the following classifiers: 1. Put mate 2.Avoid stalemate 3.Escape rook from attack 4.Push king to the edge (without putting rook under attack) 5.Make a waiting move when pre Opposition appears 6.Bring white king closer to the opponent king.

After the tutoring, students were tested by comparing their moves with the ones suggested by the Solver. If the students suggest wrong moves Solver suggests correct ones and explained in details corresponding plans. Explanations iterate until students learn the strategy.

7. Conclustion

2. Algorithms, methods and software for tutoring are developed within RGT Solver, where certain implementation and integration of chess interface and explanation mechanisms are designed and developed, measuring the performance and providing interface for the student is developed. The approach gives the following advantages:

a. The mechanism of tutoring is personalized for each student by their levels, including genius and autistic students.

b. Level by level tutoring, testing, feedback provision and correction by detailed description are provided in the interactive environment.

c. Students’ performance measurement means are provided in the developed interface tool.

The designed algorithms and software adequacy are confirmed by experiments in chess endgames tutoring, particularly rook endgame tutoring is tested.

References

[1] Arakelova E. and Pogossian E., "Tools for testing and correction of the completeness of knowledge acquisition by autistic children," pp. 159-165, Yerevan, Armenia, 2011.

[2] Pogossian E., Javadyan. A. Ivanyan. E. "Effective Discovery of Intrusion Protection Strategies.," The International Workshop on Agents and Data Mining, Lecture Notes in Computer Science, pp. 263-274, St. Petersburg, Russia, 2005.

[3] K. Khachatryan and S. Grigoryan, “Java programs for presentation and acquisition of meanings in SSRGT games”, Proceedings of SEUA Annual conference, pp. 127-135, Yerevan, Armenia, 2013.

[4] Pogossian E, Vahradyan V., Grigoryan A. "On Competing Agents Consistent with Expert Knowledge," Lecture Notes in Computer Science, AIS-ADM-07: The Intern. Workshop on Autonomous Intelligent Systems - Agents and Data Mining, St. Petersburg, 2007.

Appendix 7: The Repository of Chess Classifiers

Hambartsumyan M., Harutunyan Y., Pogossian E., **A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations**, National Academy of Sciences of Armenia, IPIA, 1974-1980 research reports**.**

The list of provided chess classifiers.

Hambartsumyan M., Harutunyan Y., Pogossian E., **A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations**, National Academy of Sciences of Armenia, IPIA, 1974-1980 research reports**.**

Описание «Фортран» программ, реализующих шахматные дескрипторы.

Введение

1.Как следует из [1], использование описаний в алгоритмах управления связано с построением алгоритмов, вычисляющих компоненты этих описаний. Напомним, что д е с к р и п т о р о м ф о р м у л ы F (или ф у н к ц и и f), заданной на универсуме А, мы назвали произвольный алгоритм, вычисляющий F(f).

Описываемые ниже дескрипторы составляют только часть из практически реализованных нами. Этот список будет расширен в ..... дескрипторами, включёнными в схемы правдоподобия целей.

Относительно отношения θ3 «часть-целое» [1] все дескрипторы классифицируются по уровням. К первому уровню относятся дескрипторы, соответствующие концевым вершинам графа θ3 или атомарным формулам: к n-ому уровню – объединение покрытий первого порядка для всех вершин (n-1)-го уровня.

Напомним, что если Pn – n-местный предикатный символ, а **a1,….an –** термы , то атомарной формулой называется запись Pn(**a1,….an).** Дескрипторы атомарных формул мы называем также элементарными дескрипторами.

Внутри каждого уровня дескрипторы разбиты на классы сходных в определённом (как правило, в шахматном) смысле дескрипторов. В частности, объединение в классы производилось по цвету или составу фигур., по геометрическим характеристикам позиции и др.

Если для этого класса построена единая обращающаяся программа, то она называется **блоком**.

Принцип, по которому дескрипторы объединяются в классы, описываются в общем описании, предшествующем каждому классу.

[1] E. Pogossian, “Adaptation of combinatorial algorithms”, *Academy of Sci. of Armenia,* p. 293, 1983.

Для удобства каждый дескриптор данного раздела заиндексирован трёхзначным числом. Первая цифра указывает номер уровня, вторая – номер класса на этом уровне, и третья – номер программы в соответствующем классе.

Программы, реализующие дескрипторы, мы будем называть ФОРТРАН – программами или для краткости, просто Ф – программами. Некоторые Ф – программы являются объединением нескольких дескрипторов. При этом каждый конкретный дескриптор, вклюяённый в эту программу, определяется заданием входных параметров.

Описаниеконкретных дескрипторов и реализующих их Ф-программ построено по следующей схеме.

1. Определение соответствующих понятий лежащих в основе данных дескрипторов на естественном языке.
2. Рисунок.
3. Програмное описание, т.е. описание реализующих данные дескрипторы Ф-программ, которое состоит из следующих частей:

   а) вход(параметры и т.п.),

   б) описание работы программы,

   в) выход (выдачи результатов, их вид и т.п.).

Подпрограммами данной Ф-программы мы назовём те Ф – программы, к которым следуют обращения непосредственно из данной

Выходные данные описанных ниже Ф-программ могут быть либо значениями переменной, либо одним или двумя массивами.

В формируемых массивах в настоящее время хранится либо содержимое некоторых клеток доски, либо координаты некоторого подмножества клеток. При формировании массивов переменной длины, рахмерность (чиосло элементов) массива берётся максимально возможной. После записи в массив конкретной информации элементы, оставшиеся незаполненными, приравниваются к 15. Отметим, что при формировании массивов координат – характеристик анализируемой фигуры, координаты клетки, не принадлежащей даске, в массивы не заносятся. Это достигается граничными условиями, определяющими размеры доски.

Заметим также, что в описании конкретной Ф-программы указываются выходные значения только основной программы, реализующей рассматриваемый дескриптор.

2. В заключение приведём список используемых в описаниях дескрипторов терминов, обозначений и выражений.

1. «Противник», «другой цвет» - определяет игрока противоположного цвета, относительно выбранного.

2. «Пункт» - клетка доски, занятая какой-либо фигурой.

3. «Поле» - свободная клетка доски.

4. «Фланг» - половина доски, содержащая вертикали от 1-ой до 4-ой, либо 5-ой до 8-ой включительно.

5. «Соседние» или «расположенные рядом с клеткой (A, B) клетки» - это клетки (A-1, B) и (A+1,B). Рис.3

6. «Клетка, расположенная непосредственно впереди клетки (A,B)» - это клетка (A,B+1) или (A,B-1) соответственно тому, стоит ли на (A,B) белая или чёрная фигура. Рис.4

7. «Соседние с клеткой (A,B) вертикали» - это вертикали (A-1) и (A+1). Рис.7

8. «Длина линий или отрезков» - количество входящих в них клеток.

9. «Длина пешечной цепи» - количество пешек, составляющих цепь.

10. «Цвет пешечной цепи» - цвет клеток, на которых стоят пешки цепи.

11. «Нижняя пешка цепм» - пешка или пешки цепи с наименьшими для белых, и наибольшими для чёрных ординатами.

12. «Пояс короля» - пояс клетки (см. 1.5.1), на которой стоит король.

13. «Фигура (A,B) – краткая запись выражения фигура, стоящая на клетке с координатами (A,B).
14. «Лёгкие фигуры» - слон и конь.
15. «Тяжёлые фигуры» - ладьи и ферзи.
16. Пешки типа «слабая» - изолированные, отсталые, слабые или сдвоенные пешки.
17. Произвольная клетка шахматной доски задаётся парой координат (A,B), 1<=A,B<=8.

Так, например, записанные в обычной шахматной нотации клетки E3 и G2 будут парами (5,3) и (7,2), соответственно.

Шахматные фигуры закодированы в соответствии со следующей таблицей:

Белые фигуры	Код(десят.)	Чёрные фигуры	Код (десят.)
ПЕШКА	6	ПЕШКА	14
СЛОН	5	СЛОН	13
КОНЬ	4	КОНЬ	12
ЛАДЬЯ	3	ЛАДЬЯ	11
ФЕРЗЬ	2	ФЕРЗЬ	10
КОРОЛЬ	1	КОРОЛЬ	9

Ценность фигур определяется также, как и в шахматной теории. А именно, если принять за условную единицу ценность пешки, то **ценности** остальных фигур будут выражаться через эту единицу следующим образом:

Фигуры	Ценность
СЛОН	3 ед
КОНЬ	3 ед
ЛАДЬЯ	5 ед
ФЕРЗЬ	9 ед

Король а этот список не включён как фигура, которую нельзя взять. В этом смысле ценность короля условно принимаемся равной сумме всех возможных материальных и позиционных преимуществ.

I. УРОВЕНЬ

Стр.

I. ДЕСКРИПТОРЫ ПЕРВОГО УРОВНЯ

1.1. **Линии**

Определяются три типа линий: вертикали, горизонтали и диагонали, причем различаются диагонали двух типов: правые и левые.

1.1.1. Вертикаль с клеткой (A,B); ВЕРТ (A)
1.1.2. Горизонталь с клеткой (A,B); ГОРИЗ (B)

Вертикалью, содержащей клетку (A,B), назовём подмножество клеток доски

$$\{(L, K) \mid L=A \,\&\, K \in \{1,2,\ldots,8\}\}$$

т.е. подмножество клеток с одинаковыми абсциссами A (рис. 8), а **горизонталью**, содержащей клетку (A,B) – подмножество

$$\{(L, K) \mid L \in \{1,2,\ldots,8\} \,\&\, K = B\}$$

т.е. подмножество клеток с одинаковыми ординатами (рис.9).

Соответствующие ф - программы ВЕРТ(А) и ГОРИЗ(В) выделяют эти подмножества* Поскольку опроделяющей для вертикали являются абсцисса A « то она и указывается в качество параметра Ф- программы. Для горизонтали по той же причине указываемся ордината B .

Работа программ ВЕРТ(А) и ГОРЙЗ(В) осуществляется следующим образом. Выделяется конкретное значение A или B, которым был осуществлен вход в соответствующую программу, а затем строятся координаты клеток указанных подмножеств. В результате работы программ формируются одномерные массивы ВЕРА(9) и ГОРВ(9), первые элементы которых равны номеру вертикали A и номеру горизонтали B соответственно, а остальные равны 1,2,...,8.

Координаты клеток, принадлежащих искомым подмножествам, описываются парами ВЕРА(1), ВЕРА(К) и ГОРВ(1), ГОРВ(К) соответственно, где $K \in \{2,3,\ldots,9\}$.

Программы содержат по 18 операторов и занимают по 98 ячеек памяти.

1.1.3. Девая диагональ о клеткой (A. B); ЛЕДИ (A,B)
1.1.4. Правая диагональ о клеткой (A.B): ПРАДИ (A,B)

Левой диагональю, содержащей клетку (A,B), назовем подмножество клеток доски

$$\{(L, K) \mid L+K=A+B\}$$

Правой диагональю, содержащей клетку (А, В), назовем подмножество клоток доски

{(L, K)| L-K=A-B }

Соответствующие Ф - программы ЛЕДИ(А,В) и ПРАДИ(А,В) выделяют указанные подмножества клеток. Поскольку для определения принадлежности клетки той или иной диагонали необходимы обе ее координаты, то они и указываются в качестве параметров рассматриваемых Ф - программ.

Программы работают следующим образом. Исходя из конкретных значений координат рассматриваемой клетки, которыми был осуществлен вход в соответствующую программу, определяется крайняя правая клетка левой диагонали либо крайняя левая клетка правой диагонали. Затем, начиная о этой клетки, формируются координаты всех остальных клеток данной линии и записываются в соответствующие массивы.

Результатом работы программ являются массивы IЛД(8), JЛА(8) и IПД(8), JПА(8) соответственно, в которых хранятся абсциссы и ординаты клеток, принадлежащих искомой линии. Произвольная клетка этой линий задается парой [IЛА (К), JЛД(К)],либо парой [I ПД(К), JПА(К)], где К = 1,...,8. Поскольку длина диагонали может быть меньше максимальной, т.е. меньше 8, то лишние элементы указанных выше массивов заполняются числом 15.

Программы содержат по 36 операторов и занимают по 148 ячеек памяти.

1.2 Списки фигур.

Этот блок объединяет дескрипторы, описывающие наличие фигур на доске. Списки сосавляются за один просмотр доски по группам фигур одинакового достоинства.

1.2.1 Список белых фигур; СПИБФ

1.2.2.Список черных фигур; СПИЧФ

Ф - программы СПИБФ и СПИЧФ составляют списки имеющихся в наличии фигур соответствующих цветов. Программы параметров не имеют.

Максимально допустимое количество фигур одного цвета по группам одинакового достоинства принято следующим:

пешек - 8,
слонов - 3,
коней - 3,
ладей - 3,
ферзей - 3.

Программы работают следующим образом.

Просматриваются в порядке очерёдности элементы множества клеток доски. Если на рассматриваемой клетке стоит фигура, то её координаты заносятся в определенные элементы массивов в зависимости от кода фигуры. Результатом работы программ являются массивы IСПБ (21), JСПБ(21), и IСЧП(21), JСЧП(21) соответственно.

Структура массивов приведена на рисунке.

Координаты пешек	Слонов	Коней	ладей	ферзей	Короля

Таким образом, координаты, например, коня, если он имеется в наличии, определяются парой [IСПБ(К), JСПБ(К)] (либо [IСЧП(К), JСЧП(К)]), где К=12,13,14 в зависимостиот количества коней.

Поскольку фигур может быть меньше, чем максимально допустимое количество, то лишние элементы массивов заполняются числом 15.

Если при просмотре списка выясняется, что очередной его элемент равен 15, это говорит об отсутствии на доске соответствующей фигуры.

Программы содержат по 77 операторов и занимают по 305 ячеек памяти.

1.3. Конфигурации «конь-пешка».

В этот блок включены дескрипторы, описывающие взаимное расположение одноцветных коня и пешки.

1.3.1. Фаланга из цепки (A,B) и коня (C,D); ФАПЕК(A,B,C,D).
1.3.2. Конь (C.D) - сбоку от пешки (A,B); КОСПЕ (A,B,C,D).
1.3.3. Конь (C.D) - перед пешкой (A,B); КОППЕ (A,B,C,D).

Фалангой одноцветных пешки и коня, стоящих соответственно на клетках (A,B) и (C,D), назовем расположение этих фигур, удовлетворяющее следующему условию:

$(D=B-1)\&((C=A-1)v(C=A+1))$ - для белых,

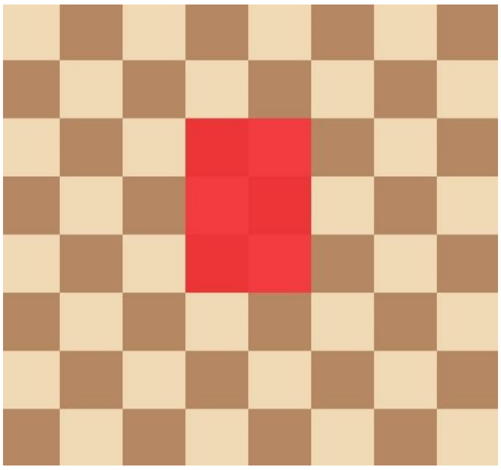


Fig.1.Центр доски белых

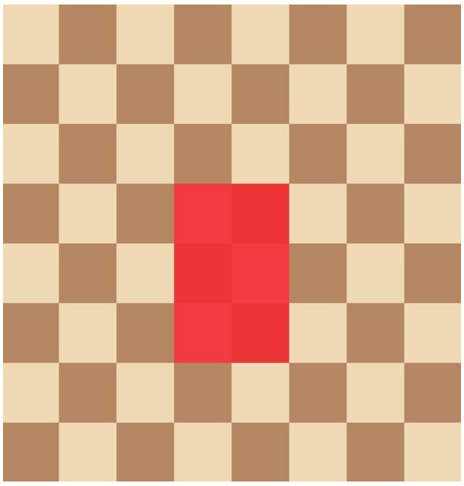


Fig. 2. Центр доски черных

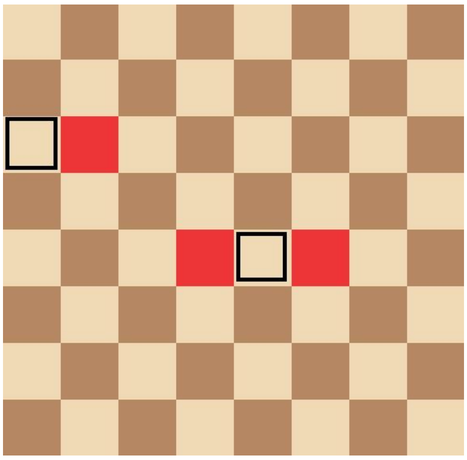


Fig.3. Соседние клетки

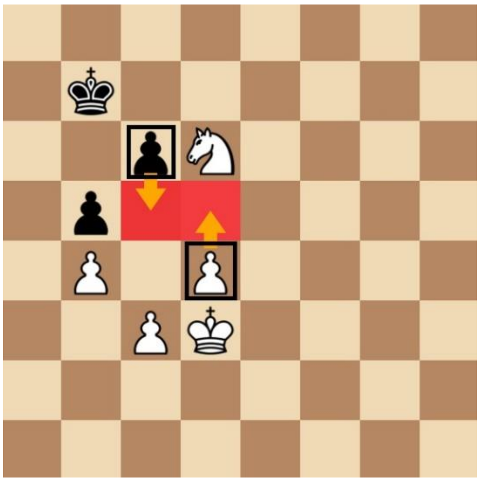


Fig.4. Клетка впереди

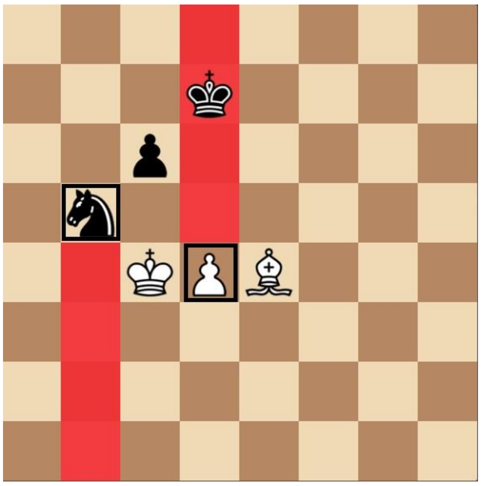


Fig.5. Клетка выше рассматриваемой

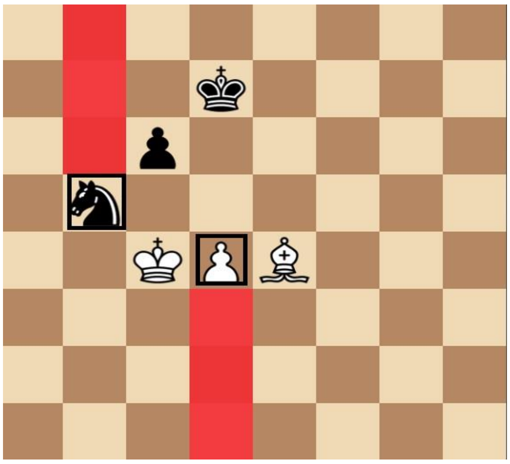


Fig.6. Клетка ниже рассматриваемой

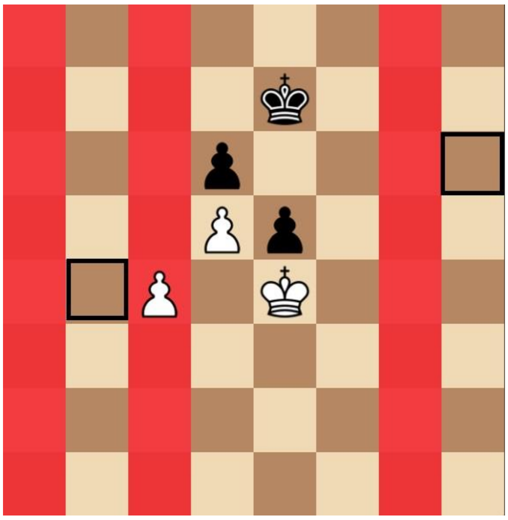


Fig.7. Соседние вертикали

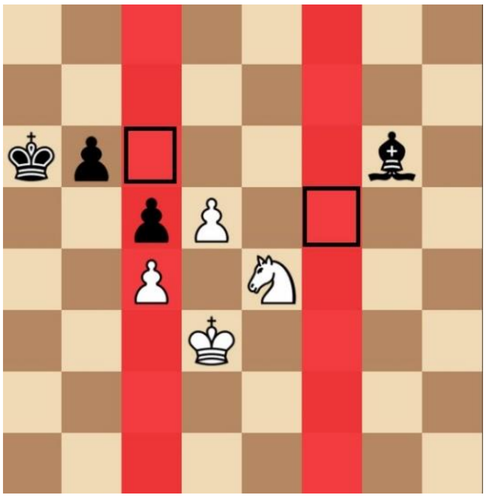


Fig. 8. ВЕРТ (1.1.1)

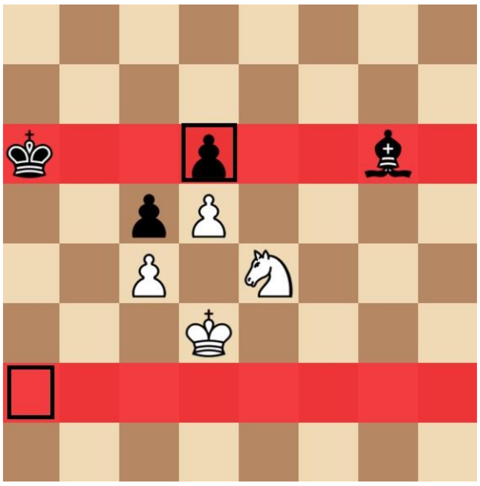


Fig.9. ГОРИЗ (1.1.2.)

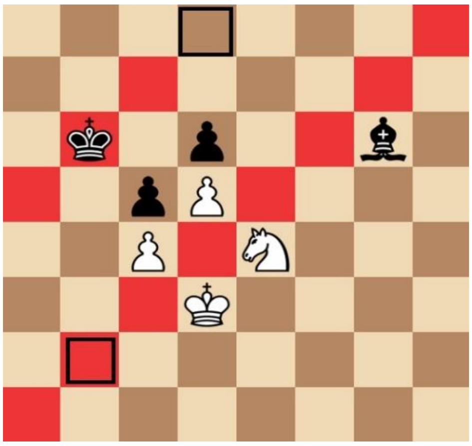


Fig.10. ПРАДИ (1.1.4.)

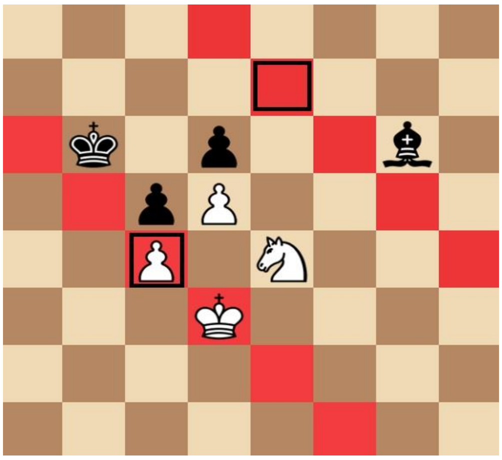


Fig.11. ЛЕДИ (1.1.3.)

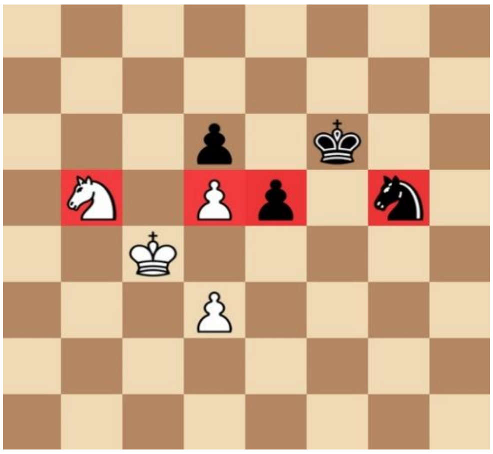


Fig.12. КОСПЕ (1.3.2.)

И **(D=B+1)&((C=A-1)v(C=A+1))** - для чёрных.

Таким образом, конь и пешка создают фалангу, если они стоят на соседних клетках одной диагонали, причем пешка выше (ниже для черных) коня. (Рис. 14).

Конь, стоящий на клетке (C,D), **находится сбоку** от одноцветной пошки, стоящей на клетке (A,B), если

(D=B)&((C=A-2)v(C=A+2))

т.е. конь стоит сбоку от пешки, если они стоят на одной горизонтали, и между ними находится одна свободная клетка доски. (Рис.12).

Конь, стоящий на клетке (C,Д), **стоит перед одноцветной пешкой**, стоящей на клетке (A,B), если выполняется следующее условие:

(C=A)&(D=B+1)) для белых,

и (C=A)&(D=B-1)) для черных.

Таким образом конь стоит перед пошкой, если он находится на клетке, расположенной непосредственно впереди пешки (Рис.13).

Поскольку положение фигуры на доске определяется обеими координатами, а указанные дескрипторы рассматривают взаимное расположение двух фигур, то координаты обеих фигур указываются в качестве параметров соответствующих Ф - программ ФАПЕК (A,B,C,D), КОСПЕ (A,B,C,D) и КОППЕ (A,B,C,D).

Программы идентифицируют факт наличия на доске указанных фигур и их конфигураций. Результатом работы программ являются значения переменной ФПК, которые определяются в зависимости от ситуации на доске следующим образом:

ФПК = 0,если фаланга из пешки и коня белых; для ФАПЕК(если белый конь - сбоку от белой пешки; или если белый конь - перед белой пешкой для КОСПЕ и КОППЕ, соответственно),
ФПК = 1, если фаланга из пешки и коня черных для ФАПЕК, (если черный конь - сбоку от черной пешки или если черный конь - перед черной пешкой для КОСПЕ и КОППЕ, соответственно),
ФПК = 2, если на (A,B) не пешка;
ФПК = 3, если на (C,D) не конь;
ФПК = 4, если нет соответствующей конфигурации.
Программы содержат по 29 операторов и занимают по 173 ячейки памяти.

1.4.Характеристики клетки.

1.4.1. Цвет клетки (A,B), G(A,B).

1.4.2. Занятость клетки (A,B) и цвет фигуры на ней; F(A,B).

Клетка (A,B) **белая,** если сумма (A+B) нечетна, и **черная** – в противном случае.

Клетка (A,B) **занята,** если на ней стоит фигура произвольного цвета.

Соответствующая Ф - программа G(A,B) и F(A,B) предназначены для выявления цвета клетки, её занятости и цвета фигуры, стоящей на ней. В качестве параметров указываются координаты рассматриваемой клетки.

Результатом работы программ являются следующие значения:
G(A,B) = 0 – если клетка (A,B) – белая,
1 – елси клетка (A,B) – черная.

F(A,B) = 0 – если на (A,B) нет фигуры,
1 – если на (A,B) – белая фигура,
2 – если на (A,B) – черная фигура.

Указанные выше Ф-программы определены как внешние функции. Тип функций – целый.

Программа G(A,B) содержит 15 операторов и занимает 81 ячейки памяти.

Программа F(A,B) содержит 15 операторов и занимает 92 ячейки памяти.

1.4.3. Клетка (A,B) – дырка; ДЫР(A,B)

Клетку (A,B) назовем **дыркой белых** (при условии 2<=B<=5), если выполненны приведенные ниже условия вместе с одним из условий G1 и H1.

M1. Хотя бы на одной клетке подмножества {(L,B+1)| L є{A-1,A+1}} стоит пешка черных.

G1. Ни на одной клетке из подмножества R={(L,K)| (L є {A-1,A+1})&(K є {B-1,B-2})} нет белой пешки.

H1. на клетках того же подмножества R стоят белые пешки, но непосредственно впереди (выше) этих пешек стоят какие-либо фигуры.

Клетку (A,B) назовем **дыркой черных** (при условии 4<=B<=7), если выполняется условие M2 вместе с одним из условий G1 и H1.

Таким образом, поле является дыркой, если оно находится под ударом пешки противника и нет возможности за один ход защитить его своей пешкой (Рис 18).

Соответствующая Ф - программа ДЫР (A,B) устанавливает факт, является ли рассматриваемая клетка дыркой. Координаты клетки указываются в качестве параметров программы.
Результатом работы программы является переменная ДАН» определяемая следующим образом:

0, если (А,Б) - дырка белых,
ДАН= 1, если (A,B) - дырка черных,
2, если (A,B) - не дырка.

Программа содержит 68 операторов и занимает 331 ячейку памяад.

1.5. Подмножества клеток.

Входящие в этот блок дескрипторы объединены по близости принципов их работы и являются вспомогательными для дескрипторов более высоких уровней.

I.5.I. Пояс кладки (A.B); ПОЯК(A,B)

Поясом клетки (A,B) назовем следующее подмножество клеток доски:
{{(L,K)| (L є {A-1,A,A+1} & K є {B-1,B,B+1}}}\(A,B)}
т.е. поясом клетки (A,B) является множество клеток, отличных от (A,B) и имеющих с ней общее ребро (Рис.15). Максимальное число клеток пояса по определению равно 8.

Соответствующая программа ПОЯК(A,B) выделяет указанное подмножество клеток. Координаты рассматриваемой клетки задаются в качестве параметров программы. Программа работает следующим образом. Строится указанное выше подмножество, начиная с его левой угловой клотки (с минимальными координатами). Найденная клетка сравнивается покоординатно о клеткой (A,B). В случае несовпадения, если клетка принадлежит доске, координаты этой клетки заносятся в предназначенные для этого массива. В противном случав соответствующие элементы массивов приравниваются к 15. Затем одна из кординат увеличивается на единицу и цикл повторяется.

Программе заканчивает работу после построения всего подмножества.

Результатом работы программы являются одномерные массивы IПФК(8), JПФК(8), в которых записаны, соответственно, абсциссы и ординаты клеток искомого подмножества. Таким образом, пара [I ПФК(N), JПФК(N)], где N={1,2,…,8}, задает некоторую клетку пояса.

Если при просмотре массивов, выясняется, что очередной элемент равен 15, это означает, что соответствующая (по номеру) клетка пояса лежит за пределами доски.

Программа содержит 40 операторов и занимает 176 ячеек памяти.

1.5.2. Пояс коня, стоящего на клетке (A.B); ПКОНЬ (A,B)

Поясом коня, стоящего на клетке (A,B), назовем еле-дающее подмножество клеток доски:

{(L,K)|(L є {A-2,A-1,A,A+1,A+2} & K є {B-2,B-1,B,B+1,B+2}})\{(M,N)|M є{A-1,A,A+1} & N є {B-1,B,B+1})}

т. о, поясом коня на (A, B) является множество клеток, имеющих общее ребро с клетками пояса клетки (A,B).
По определению максимальное число клеток пояса коня равно 16 (Рис.)

Соответствующая Ф - программа ПКОНЬ(A,B) используется как вспомогательная подпрограмма для выделения полей доступности коня, т.к. все поля, куда может пойти конь, попадают в этот пояс. Программа работает следующим образом. Поочередно порождаются все клетки указанного подмножества, начиная с левой угловой с координатами (А-2,В-2).Если рассматриваемая клетка принадлежит доске, то в очередные элементы соответствующих массивов заносятся координаты этой клетки. В противном случав рассматривается следующая клетка, и так до последней. Оставшиеся незаполненными элементы массивов приравниваются к 15.

Результатом работы программы являются массивы ПФI (16), ПФJ(16), в которое соответственно записаны абсциссы и ординаты клеток пояса коня. Координаты поля, куда может пойти конь, задаются, таким образом, парой [ПФI (К), ПФJ(К)], где К є {2,4,…,16}.
Если при просмотре массивов выясняется, что очередной элемент равен 15, это означает, что все остальные клетки пояса лежит за пределами шахматной доски.

Программа содержит 72 оператора и занимает 318 ячеек памяти.

1.6. «Разное»

1.6.1. Клетки (А,В) и (С,D) – на одной линии; НТЖЕ(А,В,С,D)

Клетки (А,В) и (С,D) считаются принадлежащими
одной горизонтали, если B=D,
одной вертикали, если A=C,
одной левой диагонали, если A+ B = C + D,
одной правой диагонали, если A - B = C – D.

Соответствующая Ф - программа НТЖЕ (A,B,C,D) устанавливает один иа этих фактов. Координаты клеток указываются в качестве параметров программы.

Программа работает следующим образом. Для рассматриваемых клеток проверяются поочередно все указанные выше условия. Если какое-либо из них выполняется, то соответствующий идентификатор получает конкретное значение.

Если не выполняется ни одно из условий, то идентификатору присваивается другое конкретное значение и программа завершает работу.

Результатом работы программы являются значения переменных НФRIZ,VERT, DIAGL, DIAGR и ФФ, которые принимают значение “O”, если рассматриваемые клетки на одной линии, и "1" в противном случае.
Программа содержит 32 оператора и занимает 127 ячеек памяти.

1.6.3. Конь (А,В) находится в центре доски; КВЦ(А,В)

Конь (А,В) находится в центре доски, если клетка (А,В) принадлежит подмножеству, называемому центром доски соответствующего цвета. Определение понятия «центр доски» было дано во введении (Рис. 16,17).

Соответствующая Ф - программа КВЦ (А,В) проверяет атот факт.

Проверка производится покоординатным сравнениям. Координаты поля, на котором стоит конь, задаются в качестве параметров программы.

Результатом работы программы является идентификатор ДАН, определенный следующим образом;

0, если белый конь в центре,
ДАН = 1, если черный конь в центре
2, если конь не в центре.

3,если на (A,B) не конь.

Программа содержит 36 операторов и занимает 145 ячеек памяти.

1.6.3. Сравнительная ценность фигур (A,B) и (C,D); H(A,B,C,D)

Фигура стоящая на (A,B) считается **ценнее** фигуры, стоящей на (C,D), если она в обычной шахматной классификации имеет больший вес.

Соответствующая ф-программа **H(A,B,C,D)** предназначена для определения, является ли фигура (A,B) ценнее фигуры (C,D). В качестве параметров указываются координаты клеток, на которых стоят рассматриваемые фигуры. Результатом работа программы являются следующие значения:

0 - если фигура (A,B) ценнее фигуры (C,D),

H(A,B,C,D) = 1 - в противном случае.

Программа содеожит 21 оператор и занимает 142 ячейки памяти.

II. УРОВЕНЬ

Стр.

2.ДЕСКРИПТОРЫ ВТОРОГО УРОВНЯ

2.1. Открытые линии

Нижеследующие определения открытых линий несколько отличаются от аналогичных шахматных понятий. Открытыми мы называем линии, на клетках которых нет не только пешек, но и вообще каких-либо фигур.

Определяются четыре типа открытых линий: открытые вертикали и горизонтали, левые и правые диагонали.

2.1.1 Открытая вертикаль с клеткой (А,В); ОТВЕР (А,В).
2.1.2 Открытая горизонталь с клеткой (А,В); ОТГОР (А,В).
2.1.3 Открытая левая диагональ с клеткой (А,В); ОТЛЕД (А,В).
2.1.4.Открытая правая диагональ с клеткой (А,В); ОТПРД(А,В).

Линия с клеткой (А,В) является **открытой**, если на клетках этой линии нет фигур (Рис. 19,20,21,22).

Ф-программа ОТВЕР(А,В), ОТГОР(А,В)Б ОТЛЕД(А,В), ОТПРД(А,В) предназначены для проверки, является ли соответствующая линия открытой. Координаты клетки, через которую проходит рассматриваемая линия, задаются в качестве парамтров соответствующей Ф-программы.

Программа вызывает подпрограмму, формирующую соответствующую линию с клеткой, координатами которой был осуществлён вход в программу. Затем проверяется, стоят ли на клетках выделенного подмножества какие-либо фигуры, и в зависимости от результатов проверки, выходной переменной присваивается то или иное значение. Результатом работы программ является следующая переменная ДАНТ:

ДАНТ = 0, если соответствующая линия открыта,
1, в противном случае.

Программы содержат обращение к подпрограммам ВЕРТ(А,В), ГОРИЗ (А,Б), ЛЕДИ (А,В), ПРАДИ (А,В) соответственно.

Программы 1.2 занимают по 153 ячейки памяти и содержат по 26 операторов, программы 3,4 занимают по 182 ячейки памяти и содержат по 30 операторов.

2.2. Отрезки линий

Рассматриваемое понятие отрезка линии не имеет прямого эквивалента в шахматах, однако очень часто используется неявным образом. В атом смысле дескрипторы данного блока можно считать вспомогательными.

Определяются четыре типа отрезков: отрезки вертикалей, горизонталей, левых и правых диагоналей.

2.2.1. Отрезок вертикали между (A,B) и (C,D); ОВЕРТ(A,B,C,D)
2.2.2. Отрезок горизонтали между (A,B) и (C,D); ОГОРЗ(A,B,C,D)
2.2.3. Отрезок левой диагонали между (A,B) и (C,D); ОЛЕДИ(A,B,C,D)
2.2.4. Отрезок правой диагонали между (A,B) и (C,D); ОПРДИ(A,B,C,D)

Отрезком (точнее - интервалом) линии между двумя данными клетками (A,B) и (C,D) этой линии клетки линии, лежащие строго между (A,B) и (C,D), т.е. клетки (A,B) и (C,D) в отрезок не включаются (Рис. 23,24,25,26).

Соответствующие Ф - программы ОВЕРТ(A,B,C,D), ОГОРЗ(A,B,C,D), ОЛЕДИ (A,B,C,D) и ОПРДИ (A,B,C,D) выделяют указанные подмножества.

Поскольку отрезок определяется двумя концевыми точками, то координаты рассматриваемых клеток указываются в качестве параметров программ.

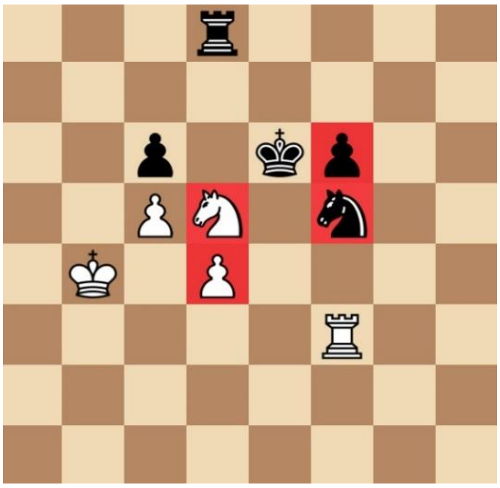


Fig.13. КОППЕ (1.3.3.)

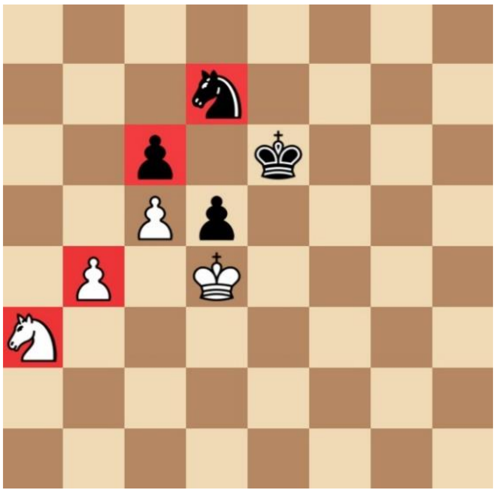


Fig. 14. ФАПЕК (1.3.1.)

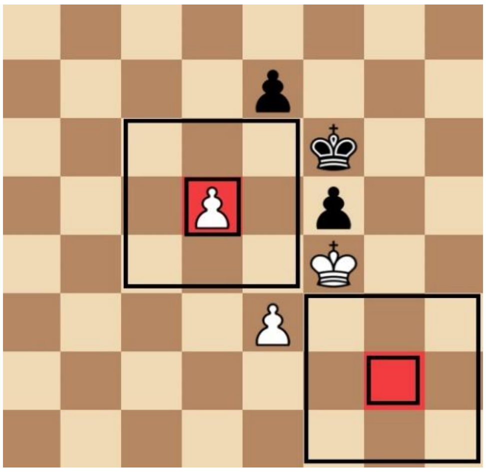


Fig.15. ПРЯК (1.5.1.)

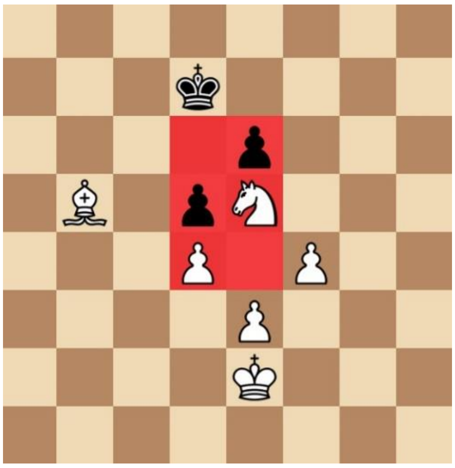


Fig.16. КВЦ (1.6.3.)

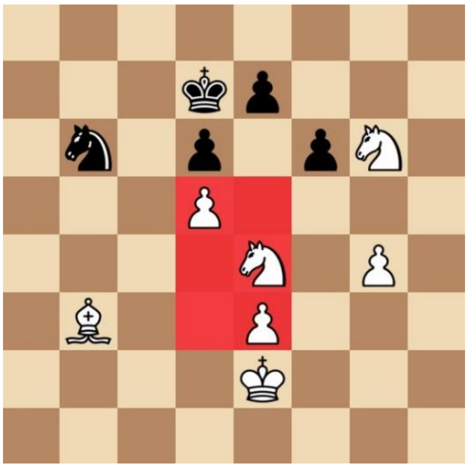


Fig.17. КВЦ (1.6.3.)

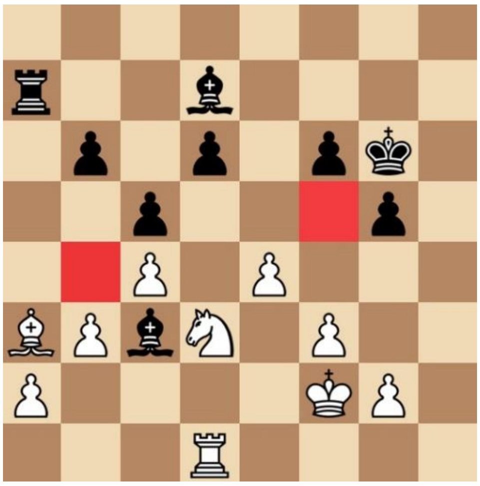


Fig.18. ДЫРКА (1.6.2.)

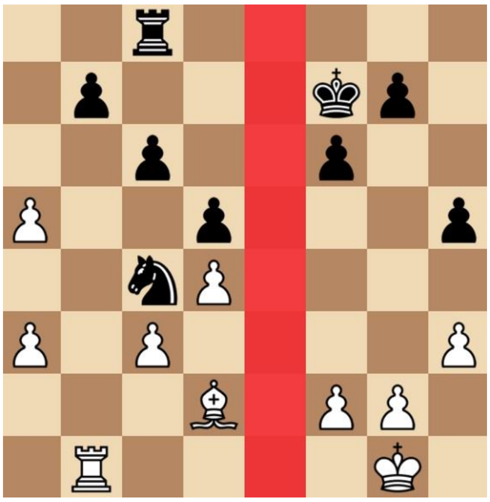


Fig.19. ОТВЕР (2.1.1.)

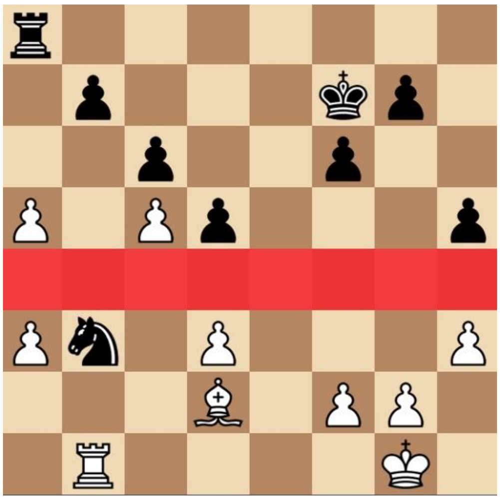


Fig. 20. ОТГОР (2.1.2.)

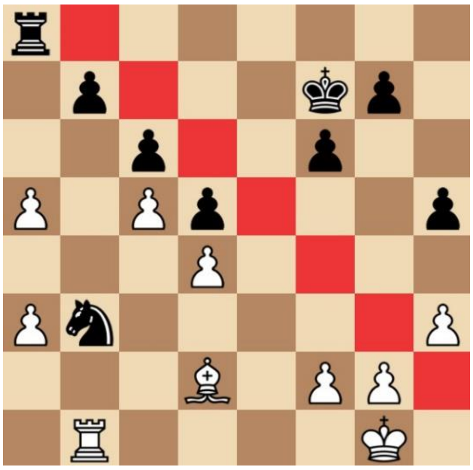


Fig.21. ОТЛЕД (2.1.3.)

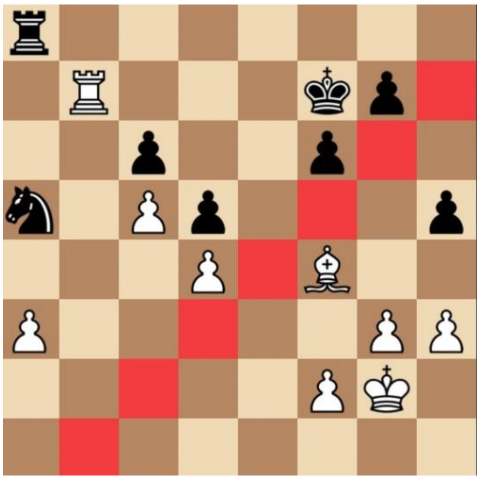


Fig.22. ОТПРД (2.1.4.)

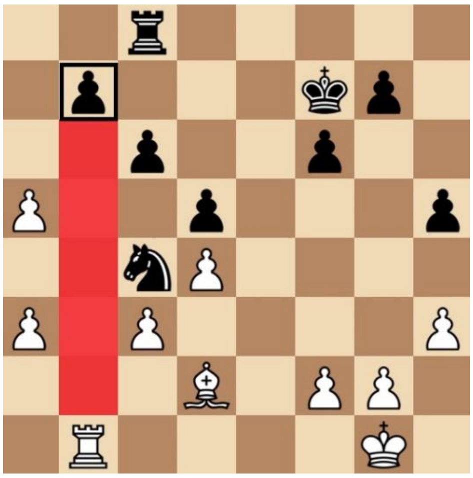


Fig.23. ОВЕРТ (2.2.1.)

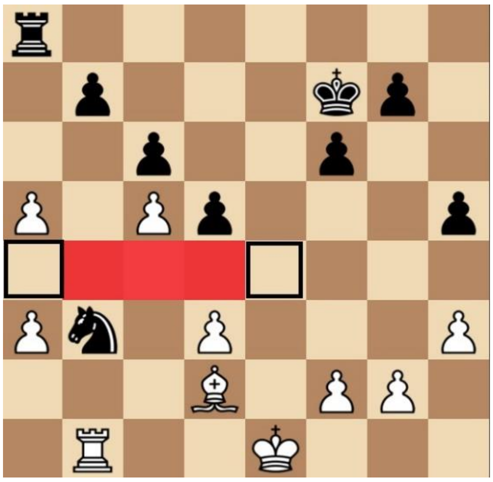


Fig.24. ОГОРЗ (2.2.2.)

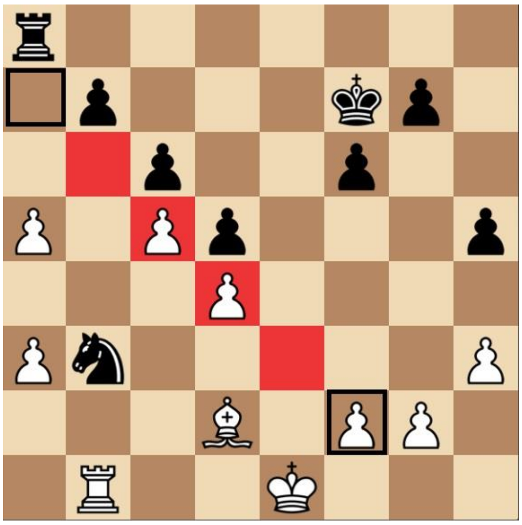


Fig.25. ОЛЕДИ (2.2.3.)

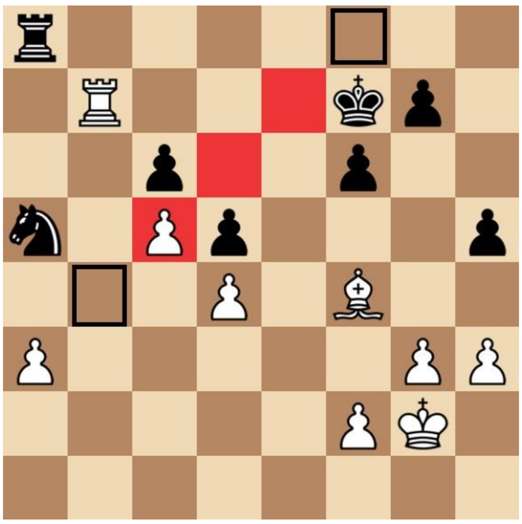


Fig.26. ОПРДИ (2.2.4.)

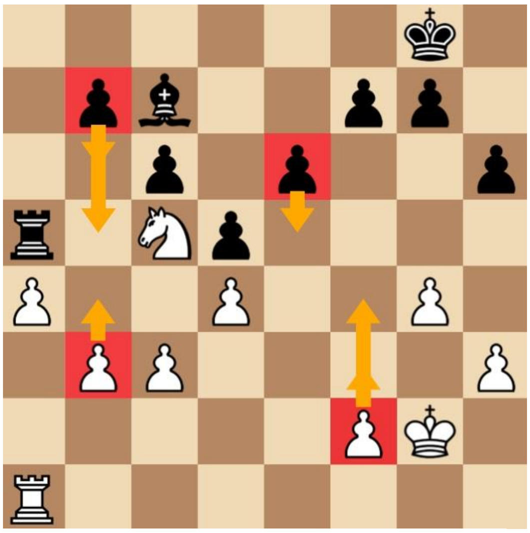


Fig.27. ПДБП, ПДЧП (2.3.1., 2.3.2.)

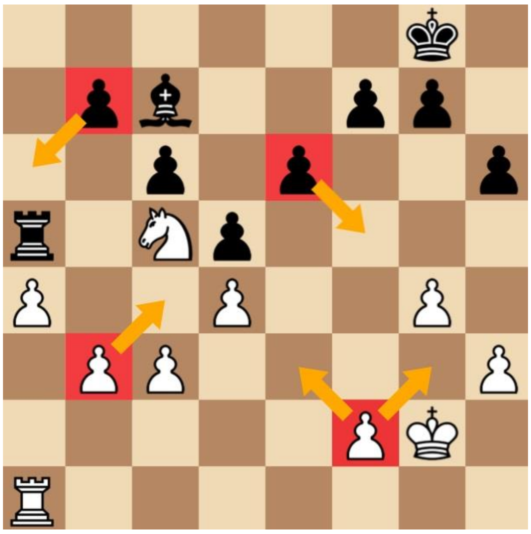


Fig.28. ПУБП, ПУЧП (2.3.3., 2.3.4.)

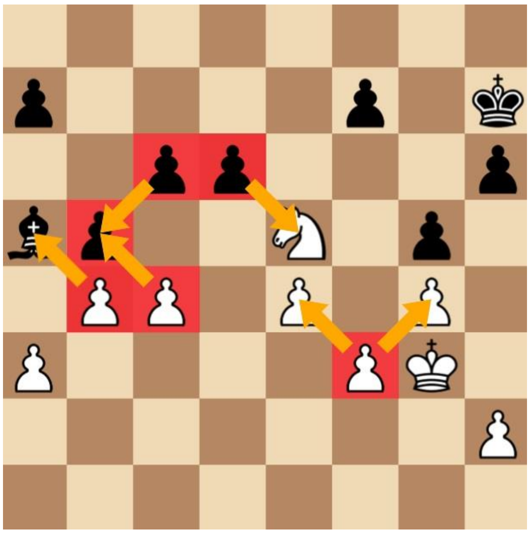


Fig.29. ВИЗВП, ВИЗЧП (2.5.1., 2.5.2.)

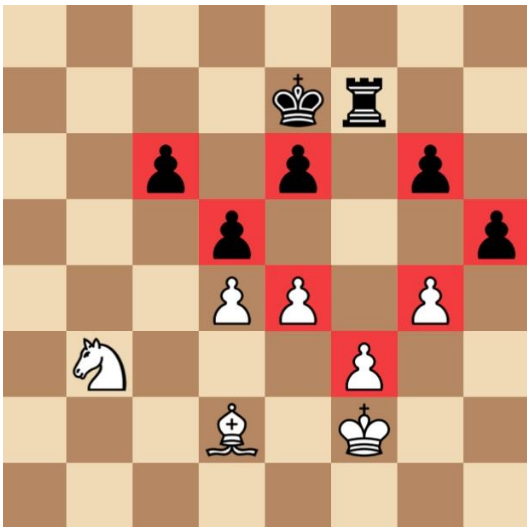


Fig.30. ЦБЕП, ЦЧЕП (2.9.1., 2.9.2.)

Каждая программа вначале проверяет, лежат ли рассматриваемые клетки на одной прямой. Проверка осуществляется вызовом программы НТЖЕ. В зависимости от результатов проверки выходной переменной присваиваются соответствующие значения, и если клетки лежат на одной линии, переходим к формированию массивов координат отрезка.
Если клетки не лежат на одной линии или совпадают, программа завершает работу.

Результатом работы программ являются, соответственно, массивы ОБЕ (7), ОГО(7), IОЛ(6) и J0Л(6), I0ТП(6) и JOТП6), а также переменная ОЛ, определяемая следующим образом:

ОЛ = 0, если клетки на одной линии,
1, если нет отрезка,
2, если щетки совпадают.

Координаты произвольной клетки отрезка вертикали, горизонтали, правой и левой диагоналей задаются следующими парами:
(ОВЕ(1), ОВЕ(К))- отрезок вертикали, где К є {2,…,7}.
(ОГО(1), ОГО(К))- отрезок горизонтали, где К є {2,…,7}.
(I0Л(К), JОЛ(К))- отрезок лэвой диагонали, где К є {1,…,6}.
(IОТП(К), JОТП(К) - отрезок правой диагонали, где К є {1,…,6}.

Указанные Ф - программы в процесса работы обращаются в подпрограммам ВЕРТ, ГОРИЗ, ЛЕДИ, ПРАДИ соответственно и к подпрограмме НТЖЕ.
Программы содержат:
ОВЕРТ - 57 операторов, 270 ячеек ОЗУ
ОГОРЗ - 57 операторов, 270 ячеек ОЗУ
ОЛЕДИ - 61 операторов, 303 ячейки ОЗУ
ОПРДИ - 61 операторов, 303 ячейки ОЗУ.

2.3. Возможности пешек.

В этот блок включены дескрипторы, описывающие поля доступности и поля, находящиеся под ударом пешки.

2.3.1. Клетка (C,D) – поле доступности белой пешки на(А,В); ПДБП(А,В,С,D)
2.3.2. Клетка (C,D) – поле доступности чёрной пешки на(А,В); ПДЧП(А,В,С,D)

Клетка (C,D) является **полем доступности** пешки, стоящей на (А,В), если пешка имеет возможность в один ход достичь клетки (С,D). (Рис.27).

Соответствующие ф - программы ПДБП (А,В,С,D) и ПДЧП (А,В,С,D) проверяют этот факт. Координаты обеих клеток задаются з качестве параметров программ.

Программы последовательно проверяют, является ли клетка (С,D) полем доступности пешки (А.В) и свободна ли клетка (С,D). В зависимости от результатов проверки определенной переменной присваиваются некоторые конкретные значения.

В результате работы программ вырабатывается следующая переменная ДН:

ДН = 0,если (С,D) - поле доступности,
1, в противном случае.

В процессе работы программы обращаются к подпрограмме - функции F(A,B).
Программы содержат по 18 операторов и занимают по 100 ячеек памяти.

2.3.3. Клетка (C,D) – под ударом белой пешки(А,В); ПУБП(А,В,С,D)
2.3.4. Клетка (C,D) – под ударом черной пешки(А,В); ПУЧП(А,В,С,D).

Полями, находящимися **под ударом пешки (А,В),** называются поля (А-I, В+I) и (А+I, В+I), если пешка белая, и поля (А+I, В-I) и (А-I, В-I), если пешка черная (Рис .28).

Соответствующие Ф - программы ПУБП (А,В,С,D) и ПУЧП (А,В,С,D) проверяют эти условия. Координаты клеток указываются в качестве параметров программ.

Программы проверяют, свободна ли клетка (C,D), и если да, то совпадают ли координаты (C,D) с координатами одного из полей, указанных в условии. В зависимости от результатов проверки выходной переменной присваиваются конкретные значения.

Если клетка (C,D) занята, программы завершают работу.

Результатом работы программ является переменная ДН.

ДН = 0, если (C,D) – под ударом,

1 – в противном случае.

Программы содержат обращение к подпрограмме функции F(A,B). Программы содержат по 19 операторов и занимают по 105 ячеек памяти.

2.4.Блок «Возможности фигур»

Блок "Возможности фигуры" включает программы которые вычисляют ходы, возможности защиты своей фигуры или взятия фигуры противника рассматриваемой фигурой.

Дескрипторы "Возможности ферзя" и "Возможности короля" фактически принадлежат соответственно третьему и четвертому уровням, однако описаны в этом блоке для сохранения общности.

2.4.1.Возможности слона, стоящего на (A,B). ПДУЗС (A,B).

2.4.2.Возможности ладьи, стоящей на (A,B). ПДУЗЛ (A,B).

2.4.3.Возможности коня, стоящего на (A,B). ПДУЗК (A,B).

Под возможностями фигуры понимаются поля, на которые фигура может сделать ход, пункты, на которых возможно взятие фигуры противника данной фигурой, и пункты, на которых стоят свои фигуры, защищенные данной.Все указанные клетки определяются в соответствии с шахматными правилами.

Соответствующие Ф - программы ПДУЗС (A,B), ПДУЗ.л (A,B) и ПДУЗК (A,B) предназначены для выделения подмножеств клеток доски, соответствующих приведенному выше определению. Координаты клетки, на которой стоит рассматриваемая фигура, указываются в качестве параметров программ.

Программы работают следующим образом. Для программ возможностей слона и ладьи выделяются линии, по которым может ходить данная фигура в соответствии с шахматными правилами, линии выделяются путем обращения к соответствующим программам.

Затем, начиная с клетки, занятой данной фигурой, проверяются на занятость клетки данной линии, лежащие по одну сторону от рассматриваемой. В зависимости от результатов проверки координаты рассматриваемой клетки заносятся в определенные элементы соответствующих массивов. Если рассматриваемая клетка занята, то рассмотрение данного отрезка линии прекращается и программа переходит к рассмотрению клеток линии лежащих по другую сторону от рассматриваемой. По завершению анализа данной линии рассматривается следующая, либо, если таковой нет, программа зивершает работу.

Для программы возможностей коня вместо линии выделяется подмножество, называемое «поясом коня», которое включает все клетки, на которые может пойти конь. Подмножество выделяется обращением к программе ПКОНЬ(A,B). Возможности коня опредаляются анализом сфрмированных массивов.

Результатом работы программ являются, соответственно, массивы [IСЛ(44), JСЛ(44)] , [IЛА(44), JЛА(44)] , [IКО(44), JКО(44)] , имещие следующую структуру:
элементы 1-28 предназначены для записи абсцисс и ординат полей доступности;
элименты 29-36 - для абсцисс и ординат пунктов, на которых возможно взятие;
элементы 37-44 - для абсцисс и ординат пунктов на которых стоят защищенныее фигуры;
Элементы указанных подмножеств, оставшиеся незаполненными, приравниваются к 15.

Елси при просмотре массивов оказывается, что очередной элемент равен 15, это означает, что поля соответствующего типа для данной фигуры полностью просмотрены.

В процессе работы программы обращаются к подпрограммам ЛЕДИ(A,B), ПРАДИ(A,B), ВЕРТ(A), ГОРИЗ(B) и ПКОНЬ(A, B) соответственно.

Программы содержат операторов и занимают ячеек памяти :
1 - 122 и 439,
2 - 112 и 417,
3 - 4 и 320, соответственно.

2.4.4. Поля доступности, удары и защита ферзем; ПДУЗФ(A,B)

Соответствующая Ф - программа ПДУЗФ(A,B) предназначена для выделения из множества клеток шахматной доски полей доступности и пунктов, находящихся под ударом или защитой ферзя. Координаты ферзя задаются в качестве параметров программы (Рис.).

Работа программы заключается в следующем. Поочередно вызываются программы возможностей слона и ладьи по координатам ферзя. Полученные в результате работы вышеуказанные программы массива поочередно переписываются в массивы IФЕ (44), JФЕ (44). При атом запись соответствующей части второго массива начинается с элемента, следующего за тем, которым окончилась запись предыдущего. По окончании записи работа программы завершается.

Результатом работы программы является формирование массивов IФЕ(44), JФЕ(44), структура которых указана на рис. . Элементы массивов, оставшиеся незанятыми, заполняются числом 15.

Библиотека внутренних программ:
1. ПДУЗС 2. ПДУЗЛ

Программа ПДУЗФ содержит 54 оператора и занимает 157 ячеек памяти.

Координаты полей доступности 1..............................28	Координаты пунктов, на которых возможно взятие 29...........................36	Координаты защищаемых пунктов 37.......................44

2.4.6.Поля доступности, удары и защита королём: ПДУКР(A,B).

Соответствующая Ф - программа ПДУЗФ(A,B) предназначена для выделения из множества клеток шахматной доски полей доступности и пунктов, на которых стоят фигуры, находящихся под ударом или защитой короля. Координаты короля задаются в качестве параметров программы (Рис.).

Работа программы начинается обращением (по координатам короля) к программе ПОЯК, с помощью которой выделяются все поля, на которые может пойти король. Затем для клеток на которых стоят фигуры противника или вообще нет фигур, проверяется, находятся ли они под ударом противника, т.е. будет ли король на этом поле находится под шахом. Проверка производится обращением к программе КИПИН по координатам рассматриваемого поля.

Координаты полей, не находящихся под ударом фигур противника, заносятся в соответствующие элементы массива IKP(44), JKP(44). После этого выделяются пункты, на которых стоят фигуры своего цвета и их координаты также заносятся в соответствующие элементы указанных массивов.

Отдельно проверяются возможность рокировки. Если рокировка возможна, то соответствующим элементам указанных выше массивов (в подмассивах, предназначенных для полей доступности) присваивается значение 12 или 13, в зависимости от того, является ли рокировка короткой или длинной соответственно. Этим завершается работа программы.

Результатом программы являются массивы [IKP(44), JKP(44)] структура которых указана на рис.   . Элементы массивов, оставшиеся незаполненными, присваиваются к 15.

Библиотека внутренних првграьш:

1. ПОЯК 2. ШАХ.

Программа ПДУКР содержит 158 операторов и занимаем 640 ячеек памяти.

2.5. "Угроза взятия и защита пешкой".

В этот блок включены дескрипторы,описывающиеблизкие в шахматном смысле понятия.

Так как направление удара пешкой зависит от цвета пешки, то дескрипторы разделены по цвету. Понятия взятия и защиты определяются в соответствии с шахматными правилами.

2.5.1. Взятае и зашита белой пешкой; ВИЗБП (A,B,C,D).

2.5.2. Взятие и защита черной пешкой; ВИЗЧП (A,B,C,D).

Пешка имеет возможность **взятия (защиты),** если на пункте (C,D) находящемся под ударом пешки (A,B),стоит фигура противоположного (своего) цвета. (Рис. 29).

Соответствующие Ф - программы ВИЗБП (A,B,C,D) и ВИЗЧП (A,B,C,D) предназначены для выявления этих фактов. Координаты клеток задаются в виде параметров программ.

Каждая программа проверяет, находится ли пункт (C,D) под ударом пешки, стоящей на (A,B). Если да, то в зависимости от того, какого цвета фигура стоит на (C,D), выходной переменной присваивается конкретные значения.

Результатом работы программ является следующая переменная ДН:

ДН = 0, если возможно взятие или защита,

1, в противном случае.

В процессе работы программы используют подпрограмму - функцию F(A,B).

Программы содержат но 36 операторов и занимают по 157 ячеек памяти.

2.7. Блок «Пешечные признаки».

В этот блок объединены дескрипторы по типу рассматриваемых в них Фигур, а именно пешек. В каждом из них проверяется наличие на доске пешек о определенными свойствами. Эти свойства имеют свои эквиваленты в шахматах.

2.7.1. Пешка (A,B) – проходная; ПРОП(A,B)
2.7.2. Пешка (A,B) – блокированная; БЛОП(A,B)
2.7.3. Пешка (A,B) – слабая; СЛАП(A,B)
2.7.4. Пешка (A,B) – изолированная; ИЗОП(A,B)
2.7.5. Пешка (A,B) – отсталая; ОТП(A,B)

Пешка, стоящая на клетке (A,B), называется **проходной**, если впереди этой пошки по вертикали нет пешек противника, и ни одна из этих клеток не бьется пешками противника (Рис.35).

Пешка называется **блокированной**, если на клотке, находящейся непосредственно перед этой пешкой, стоит пешка противника (Рио.36)

Пешка называется **слабой**, если клетка, находящаяся непосредственно перед этой пешкой, является дыркой того же цвета, что и пешка (Рис.31).

Пешка называется **изолированной**, если на соседних с этой клеткой вертикалях нет пешек того же цвета (Рис.34).

Пешка называется **отсталой**, если на горизонтали с данной клеткой и на горизонталях, лежащих ниже данной, нет других пешек того же цвета, при условии, что на доске имеются хотя бы две пешки этого цвета (Рис. 33).

Соответствующие Ф - программы ПРОП(A,B), БЛОП(A,B), СЛАП(A,B), ИЗОП(A,B) и ОТП(A,B) предназначены для проверки этих фактов. Координаты пешки указываются в качестве параметров программы. Каждая программа проверяет условие принадлежности пешки к соответствующему типу, и в зависимости от результатов проверки выходной переменной присваиваются соответствующие значения.

Результатом работы программы L является следующая переменная ТП:

0, если на (A,B) стоит бедая пешка с признаком L,
ТП = 1, если на (A,B) стоит черная пешка с признаком L,
2, если на (A,B) не обладает L.

В качестве подпрограмм данной подблок использует Ф - программы ВЕРТ(A), ДЫР(A,B) и программу - функцию F(A,B). Количественные характеристики программ следующие:

Имя программы	Количество операторов	Количество ячеек памяти
ПРОП(A,B)	43	177
БЛОП(A,B)	20	116
СЛАП(A,B)	39	192
ИЗОП(A,B)	80	366
ОТП(A,B)	56	303

2.7.6. Пешка (A,B) – сдвоенная; СДВП(A,B).
2.7.7. Фаланга пешек; ФАПЕ(A,B).

Пешка (A,B) называется сдвоенной если на одной вертикали с ней стоит хотя бы одна пешка того же цвета (Рис. 32).

Пешка (A,B) образует фалангу с другой пешкой своего цвета, если последняя стоит на соседней с (A,B) по горизонтали клетке. (Рис. 41).

Соответствующая Ф – программа СДВП(A,B) и ФАПЕ(A,B) предназначены для выявления вышеуказанных фактов. Координаты пешки указываются в виде параметров программы.

Программы проверяют, соответственно, есть ли на вертикали другая пешка того же цвета, или стоит ли одноцветная пешка рядом с данной. В зависимости от результата проверки выходной переменной присваивается конкретное значение.

Результатом работы программы являются следующие переменные ТП и ФА:

ТП = 0, если на вертикали не меньше 2 б. пешек,
1, если на вертикали не меньше 2 ч. пешек,
2, если на вертикали одна пешка,
3, если на вертикали нет пешек.

ФА = 0, если рядом с б. пешкой стоит б. пешка,
1, если рядом с ч. пешкой стоит ч. пешка,
2, если рядом с пешкой нет одноцветных,
3, если на (A,B) – не пешка.

Программа использует в процессе работы подпрограмму – функцию F(A,B).

Программа СДВП содержит 30 операторов и занимает 134 ячейки памяти.

Программа ФАПЕ содержит 40 операторов и занимает 178 ячейки памяти.

2.8. Атака на важный пункт.

Дескрипторы этого блока сгруппированы по близости в шахматном смысле описываемых ими понятий. В основе каждого из них лежит одна из основных шахматных идей: занятие своими фигурами, хотя бы временно, важных со стратегической точки зрения полей.

2.8.1. Легкая Фигура стоит на дырке (A,B); ЛФДЫР(A,Б).

Легкими фигурами называют коней и слонов. Определение дырки дано в 1.6.2. (Рис.42).

Соответствующая Ф - программа ЛФДЫР(A,B) предназначена для выявления факта, стоит ли какая-либо легкая фигура на клетке, являющейся дыркой для противника. Координаты этой клетки задаются в качестве параметров программы.

Программа обращением к подпрограмме ДЫР(A,B) проверяет, является ли клетка (A,B) дыркой. Если да, то проверяется, стоит ли на (A,B) легкая фигура соответствующего цвета и в зависимости от результатов проверки выходной переменной присваиваются соответствующие значения. В противном случае программа завершает работу.

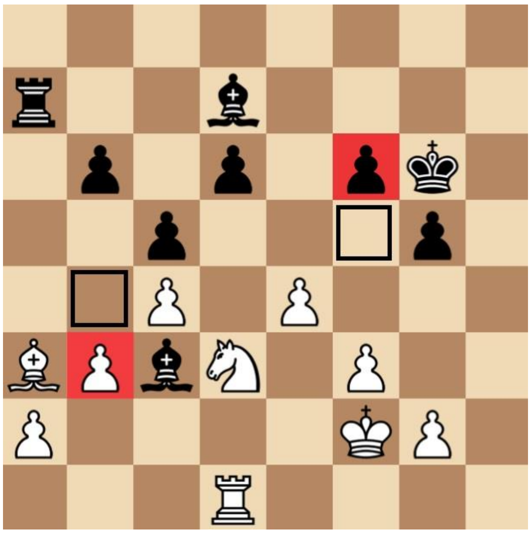


Fig.31. СЛАП (2.7.3.)

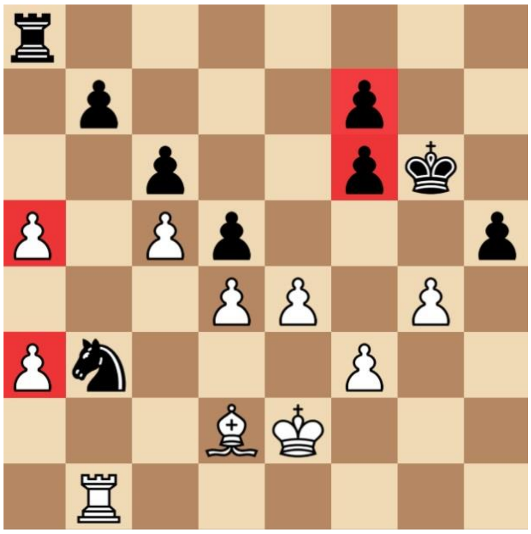


Fig.32. СДВП (2.7.6.)

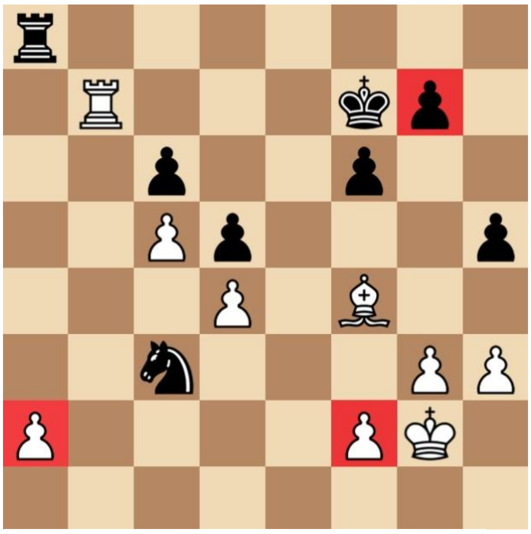


Fig.33. ОТП (2.7.5.)

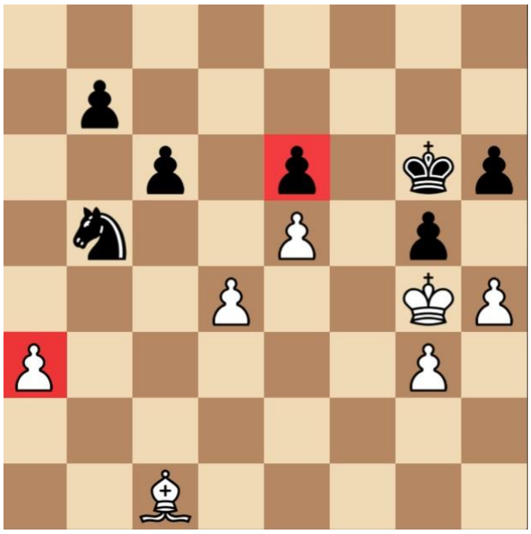


Fig.34. ИЗОП (2.7.4.)

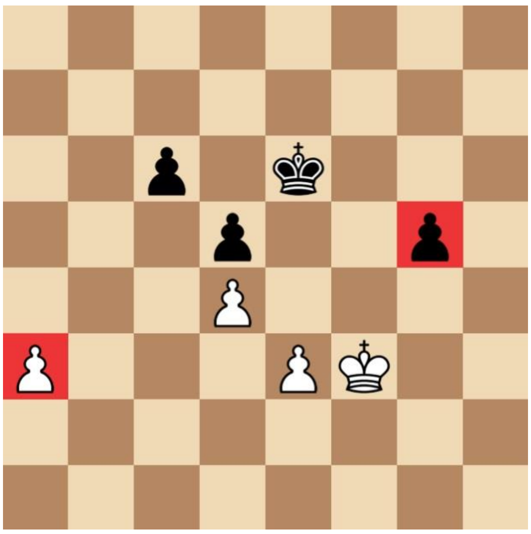


Fig.35. ПРОП (2.7.1.)

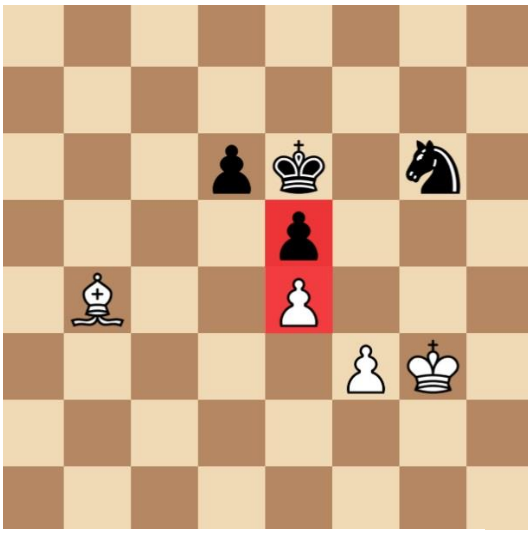


Fig.36. БЛОП (2.7.2.)

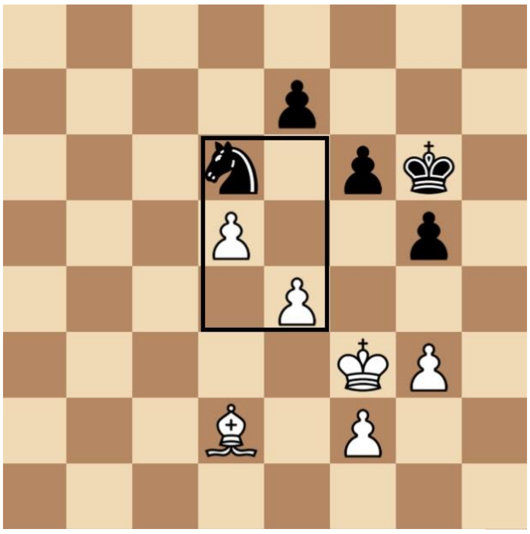


Fig.37. ЛЕВЦ (2.7.9.)

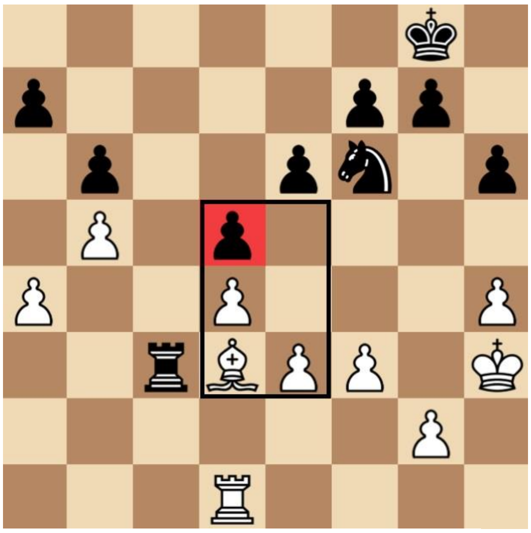


Fig.38. ПЕВЦ (2.7.9.)

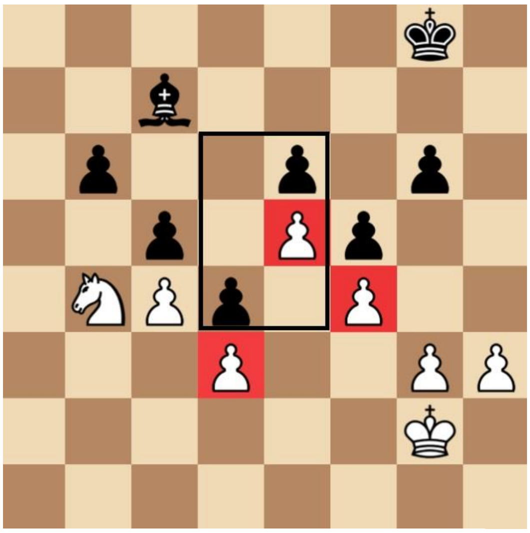


Fig.39. УНЦ (2.7.9.)

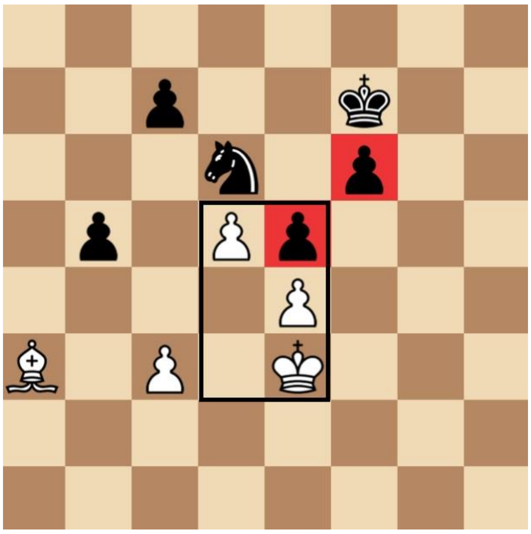


Fig.40. УНЦ (2.7.9.)

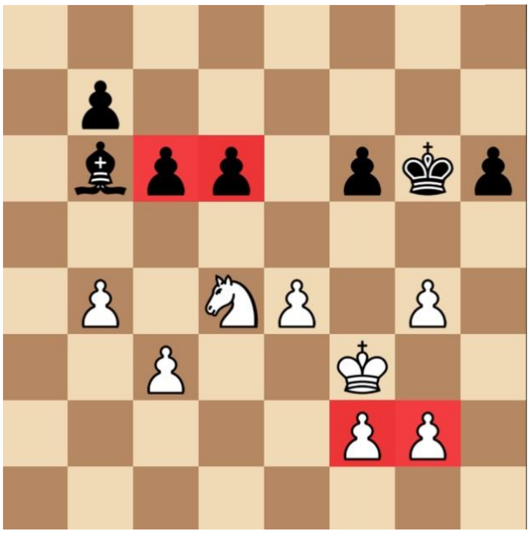


Fig.41. ФАПЕ (2.7.7.)

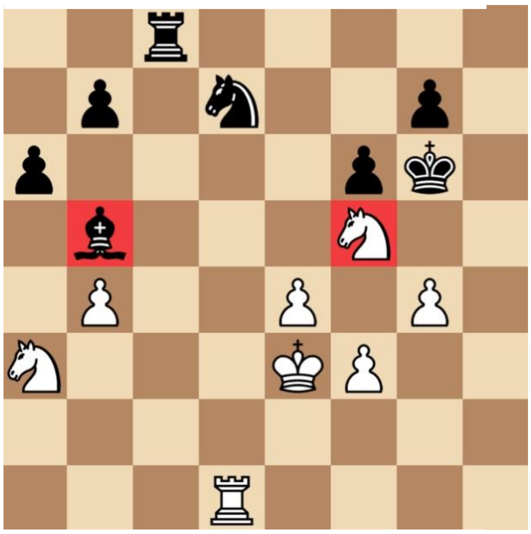


Fig.42. ЛФДЫР (2.8.7.)

Результатом работы программы является следующая переменная ДНТ:

$$\text{ДНТ} = \begin{cases} 0, \text{ если черная л.ф. стоит на дырке белых,} \\ 1, \text{ если белая л.ф. стоит на дырке черных,} \\ 2, \text{ если на дырке нет л.ф. противника.} \end{cases}$$

В процессе работы программа обращается к подпрограммам ДЫР(А,В).
Программа содержит 56 операторов и занимает 157 ячеек памяти.

<u>2.8.2. Пешки - вне атаки слона (А,В); ПВАС(А,В).</u>

Дескриптор вычисляется лишь при выполнении следующих условий:
1. на доске нет коней противника,
2. на доске находится не более одной тяжелой фигуры противника,
3, на доске имеются хотя бы три свои пешки и слон противника.

При выполнений этих условий пешки считаются находящимися вне атаки слона, если цвет клеток, на которых стоят пешки, отличается от цвета клетки, на которой стоят слон противника (Рис.43).

Соответствующая Ф - программа ПВАС(А,В) выявляет этот факт. Координаты слона задаются в качестве параметров программы.

Программа проверяет условия 1,2,3. Если хоть одно из них не выполнено, программа завершает работу. В противном случае сравнивает цвет клетки, на которой стоит слон, с цветом клеток, на которых стоят пешки. В зависимости от результата сравнения определенной переменной присваивается то или иное значение.

Результатом работы программы является следующая переменная ДОН:

$$\text{ДОН} = \begin{cases} 0, \text{ если б. пешки вне атаки слона,} \\ 1, \text{ если ч. пешки вне атаки слона,} \\ 2, \text{ в противном случае.} \end{cases}$$

В процессе работы программа обращается к следующим подпрограм F(A,B), G(A,B), СПИБФ, СПИЧФ.

Программа содержит 58 операторов и aammaef 170 ячеек памяти.

<u>2.8.3. Количество ударов на пункт иди поле (А,В); КИПИН(А,В).</u>

Под "количеством ударов" здесь понимается число недальнобойных фигур, под ударом которых находится рассматриваемая клетка, плюс число дальнобойных фигур, на линиях действия которых расположена рассматриваемая клетка.

Соответствующая Ф - программа КИПИН(А,В) предназначена для вычисления количества ударов обеих сторон на клетку (А,В). Координаты рассматриваемой клетки задаются в качестве параметров программы.

Работа программы начинается с обращения к программе СПИБФ. Затем проверяется, стоят ли на клетках (A-1,B-1) и (A+I.B-I) белые пешки. Если да, то значение переменной КУПБ увеличивается на I, После этого поочередно порождаются отрезки линий, проходящих через рассматриваемую клетку. Для каждого отрезка проверяется наличие на нем фигуры соответствующего типа, т.е. для диагонали - слонов или ферзей, для вертикали и горизонтали - ладей или ферзей. Если такая фигура есть, то значение соответствующей переменной увеличивается на I. Программой учитываются и "скрытые" удары, например, ферзь и слон на одной диагонали. После просмотра всех отрезков вычисляется количество ударов конем на данную клетку. Для этого вызывается программа ПКОНЬ по координатам рассматриваемой клетки, и проверяется наличие коней на клетках из сформированного множества. Цикл завершается вычислением количества ударов королем и суммированием количества всех ударов на рассматриваемую клетку.

Далее производится тот же цикл работ для черных, после чего программа завершает работу.

Результатом работы программы являются значения следующих переменных:

КУПБ – количество ударов пешек белых

КУПЧ - количество ударов пешек черных

КУЛФБ – легких фигур белых,

КУЛФЧ – легких фигур черных,

КУЛАБ – ладей белых,

КУЛАЧ – ладей черных,

КУТФБ – ферзей и короля белых,

КУТФЧ – ферзей и короля черных,

КУБПУ – белых на пункт,

КУЧПУ – черных на пункт.

В процессе работы программа обращается к следующим подпрограммам:

1.СПИБФ	4.ПОЯК
2.СПИЧФ	5.F
3.ПКОНЬ	6.НТЖЕ

Программа КИПИН содержит 432 оператора и занимает 2072 ячейки памяти.

III УРОВЕНЬ.

3.1. Блок «некоторые подмножества клеток доски, связанные с пешками».

Дескрипторы данного блока сгруппированы по близости выполняемых ими функций. Они служат для формирования и хранения координат некоторых подможеств клеток доски, связанных с определенными типами пешек (в данном случае - с проходной и блокированной пешками).

Понятия, лежащие в основе этих дескрипторов, имеют свои эквиваленты в шахматах.

3.1.1. Поде превращения проходной пешки (A,B).

Полем преврщения проходной пешки, стоящей на клетке (A,B), называется поле, на котором данная пешка имеет возможность превращения в какую-либо фигуру. (Рис. 46)

Это поля: - (A,8), если пешка белая,
- (A,1), если пешка черная.

Соответствующая Ф - программа ПППП(A,B) предназначена для нахождения на доске поля превращения произвольной проходной пешки. Координаты пешки задаются в качестве параметров программы.

Программа работает следующим образом:

Обращением к программе ПРОП(A,B) проверяется, является ли пешка проходной. Если да, то соответствующам переменным присваиваются координаты поля превращения данной пешки в соответствии с её цветом. В противном случае программа завершает работу.

Результатом работы программы являются значения переданных: IПФП, JПФП, равные соответственно абсциссе и ординате поля превращения.

В процессе работы программа обращается к подпрограммам ПРОП(A,B) и F(A,B).

Программа содержит 31 оператор и занимает 161 ячеек памяти.

3.1.2. Критические поля проходной пешки (A,B).

Критическими полями проходной пешки (A,B) являются:

1. Клетка, расположенная через одну выше (ниже) рассматриваемой и соседние с ней, если рассматриваемая клетка (A,B) находится на 2,3 или 4 горизонталях и пешка (A,B) – белая (на 5,6,7 горизонталях и пешка (A,B) – черная).
2. Клетка, расположенная непосредственно выше (ниже) рассматриваемой вместе с соседними, и указанные в пункте 1 клетки, если рассматриваемая клетка лежит на 5.6 или 7 горизонталях и пешка (A,B) – белая (на 2,3,4 горизонталях и пешка (A,B) – черная).
3. Кдетки (2,7) или (7,7), еслипешка (A,B) – белая и находится на 1 или 8 вертикалях соответственно, и клетки (2,2) или (7,2), если пешка (A,B) – черная, и находится на 1 или 8 вертикалях соответственно. (Рис. 47).

Таким образом, критическими полями проходной пешки, стоящей на клетке (A,B) назовем следующие подмножества клеток доски

{L,K| A не϶{1,8}& B ϵ {2,3,4} → (L ϵ{A-1,A,A+1}&K=B+2)V(A не ϵ{1,8}&B ϵ{5,6,7} → (L ϵ{A-1,A,A+1}&K ϵ{B+1,B+2})V(A=1 →(L=2&K=7))V(A=8→(L=7&K=7))} – если пешка белая, и
{L,K| A не϶{1,8}& B ϵ {5,6,7} → (L ϵ{A-1,A,A+1}&K=B-2)V(A не ϵ{1,8}&B ϵ{2,3,4} → (L ϵ{A-1,A,A+1}&K ϵ{B-1,B-2})V(A=1 →(L=2&K=2))V(A=8→(L=7&K=2))} – если пешка черная,

Соответствующая ф - програшда КРППП(A,B) предназначена для нахождения на доске критических полей какой-либо проходной пешки. Координаты пешки задаются в качестве параметров программы.

Программа работает следующим образом:

Обращением к программе ПРОП(A,B) проаеряется является ли пешка (A,B) проходной. Если да, то формируются массивы координат клеток той соответствующих подмножеств. Если рассматриваемая клетка из указанного подмножества лежит за пределами доски, то соответствующие элементы массивов приравниваются к 15. В противном случае программа завершает работу.

Результатом работы программы являются одномерные массивы IКПФ(6), JКПФ(6), в которых хранятся, соответственно, абсциссы и ординаты критических полей проходной пешки, Таким образом, пара [IКПФ(K), JКПФ(K)],где K ϵ{1,2,…,6}, задает координаты некоторого критического поля проходной пешки, стоящей на (A,B).

В процессе работы программа обращается к подпрошраммам ПРОП(A,B) и F(A,B).

Программа содержит 104 оператор и занимает 470 ячеек памяти.

3.1.3. Квадрат проходной пешки (A,B).

Понятие «квадрат проходной пешки» используется в некоторых эндшпилях, и указывает подмножество клеток, при выходе из которого король противника уже не сможет помешать прохождению пешки (см. рис. 48).

Квадратом проходной пешки, стоящей на (A,B) назовем подмножество клеток доски:

{(L,K)|(Lє{A+B-8,A+B-7,…,A-B+8}&((B не есть 2→Kє{B,B+1,…,8})V(B=2→Kє{B+1,B+2,…,8})))\(A,B) –если пешка белая, и

{(L,K)|(Lє{A-B+1,A-B+2,…,A+B-1}&((B не есть 7→Kє{1,2,…,B})V(B=7→Kє{1, 2,…,B-1})))\(A,B) – если пешка черная.

Среди определенных таким образом клеток могут оказаться клетки, расположенные вне доски. Программа имеет граничные условия, отсекающие клетки подобного рода.

Соответствующая программа КВПП(A,B) предназначена для выделения на доске квадрата какой-либо проходной пешки. Координаты пешки указываются в качестве параметров программы.

Программа работает следующим образом:

Обращением к программе ПРОП(A,B) выясняется, является ли пешка (A,B) проходной. Если да, то в соответствии с вышеуказанными определениями формируются массивы координат квадрата. В противном случае программа завершает работу. В результате работы программы формируются одномерные массивы IКВП(48), JКВП(48), содержащие, соответственно, абсциссы и ординаты клеток, составляющих квадрат проходной пешки. Если квадрат содержит меньше 48-ми клеток, то остальные элементы массивов заполняются числом 15. Таким образом, произвольная пара [IКВП(K),JКВП(K)], где Kє{l,2,...,48}, задает координаты некоторой клетки квадрата проходной пешки. Если при просмотре массивов выясняется, что очередной элемент равен 15, это означает, что соответствующая клетка квадрата лежит за пределами доски.

Программа обращается к подпрограммам ПРОП(A,B) и F(A,B).

Программа содержит 82 операторов и занимает 339 ячеек памяти.

3.1.4. Критические поля блокированной пешки(A,B).

Критическими полями блокированной пешки, стоящей на клетке (A,B), назовем следующее подмножество клеток доски:

{(L,K)|((Lє{A-3,A-2,A-1,A+1,A+2,A+3}) & (K=B))}

Таким образом, критическими полями блокированной пешки будут по три последовательно соседних клетки в каждую сторону на той же горизонтали. (Рис.49).

Соответствующая Ф-программа КРПБП(A,B) предназначена для нахождения на доске критических полей блокированной пешки. Координаты пешки задаются в качестве параметров программы:

Программа работает следующим образом:

Обращением к программе БЛОП(А,В) выявляется, является ли пешка (А,В) блокированной. Если да, то в соответствии с вышеуказанным определением формируется массив координат критических полей. В противном случае программа завершает работу.

В результате работы программы формируется одномерный массив КБП(7). Первый элемент массива равен В, а остальные - абсциссам критических полей блокированной пешки. Если критических полей меньше шести, то оставшиеся элементы массива заполняются числом 15. Таким образом, пара [IКПБ(К),JКПБ(К)], где К϶{2,7}, задает координатs некоторого критического поля блокированной пеilки (А,В). Если при просмотре массивов координат критических полей выясняется, что очередной элемент равен I5, это означает, что соответствующее поле находится вне доски.

Программа в процессе работы обращается к подпрограммам БЛОП(А,В) и F(А,В).

Програмш содержит 36 операторов, занимает 181 ячеек памяти.

3.2. Материальный перевес на фланге.

Включенные в этот блок дескрипторы сгруппированы по типу рассматриваемых в них фигур (в данном случае - пешек).

Материальный перевес на фланге понимается в том смысле, что у одной из сторон больше пешек на этом фланге, чем у противника.

3.2.1.Возможность пешечного прорыва белых.
3.2.2.Возможность пешечного прорыва черных.

Соответствующие Ф — программы В03БП и ВОЗЧП предназначены для выявления факта, что на одном из флангов доски количество пешек соответствующего цвета больше противоположного. Причем нет сдвоенных пешек данного цвета.

Программа работает следующим образом:

Подсчитываются количества белых и черных пешек на одном из флангов. Обращением к программе СДВП проверяется, есть ли среди них свободные. Если нет, то количество пешек сравнивается, и, в зависимости от результата сравнения определенной переменной присваивается некоторое конкретное значение. В противном случае аналогичный подсчет проводится для следующего фланга и программа завершет работу.

Результатом работы програшы являются значения переменной ВОП, определенной следующим образом:

ВОП = 0, если есть возможность белого или черного прорыва,
1, в противном случае.

Программ в процессе работы обращаются к программе СДВП(А,В).

Программа содержит по 70 операторов и занимает по 289 ячеек памяти.

3.3. Фаланга пешек в центре доски.

В данном блоке дескрипторов для фигур каждого цвета проверяемся факт, что на доске имеется фаланга пешек, стоящая на определенных клетках центра доски. Это понятие выделено ввиду того, что при таком расположении пешка в шахматной смысле относятся к одним из наиболее активных. (Рис.51,52).

3.3.1. Фаланга белых пешек в центре доски
3.3.2. Фаланга черных пешек в центре доски

Соответствующие Ф программы БП44 и БII55 предназначены для выявления факта, что на доске имеется фаланга пешек определенного цвета ,стоящих на клетках (4,4) и (5,4) или (4,5) и (5,5) соответственно.

Программы работают следующим образом:

Обращением к программе ФАПЕ(А,В) проверяется наличие фаланги на рассматриваемых клетках доски. В зависимости от результатов проверки определенной переменной присваиваются некоторые значения и программы завершают работу.

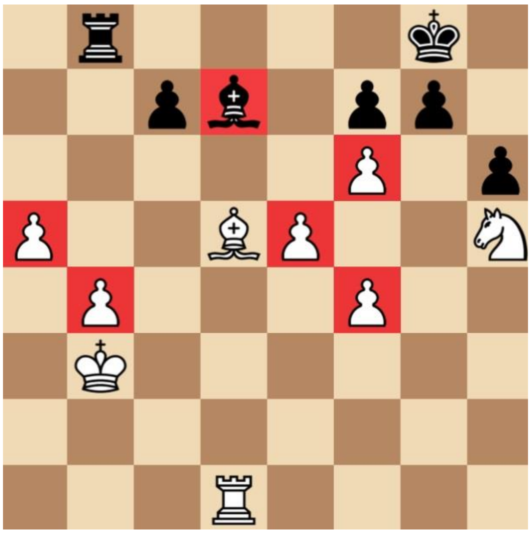


Fig.43. ПВАС (2.8.2.)

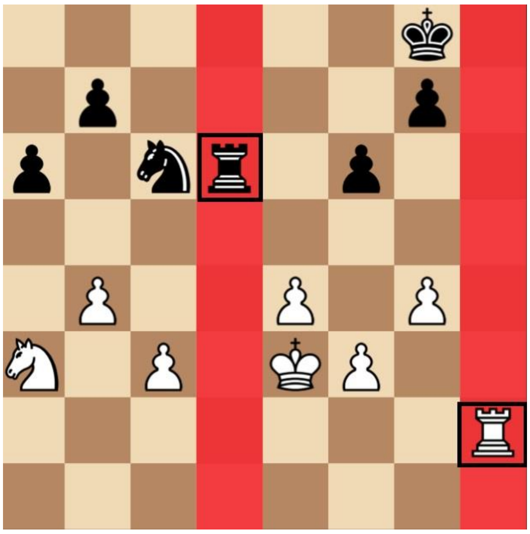


Fig.44. ЛАНОВ (3.6.3.)

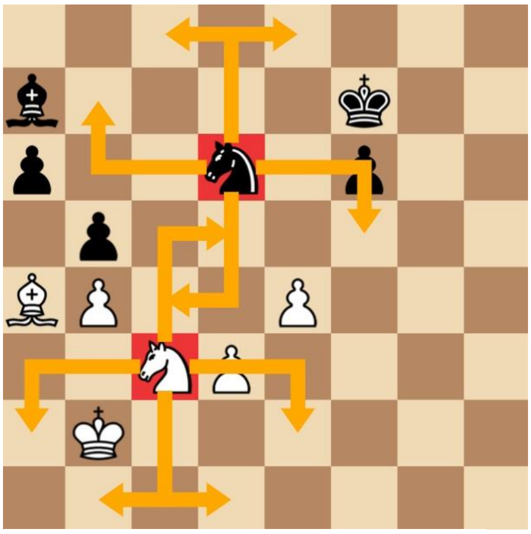


Fig.45. РРИУК (3.4.1.)

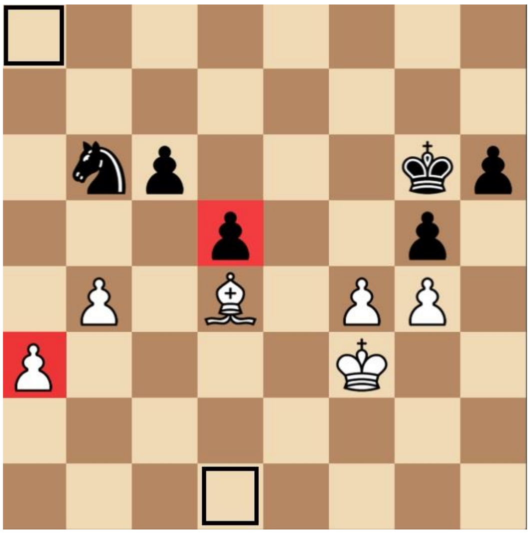


Fig.46. ПППП (3.1.1.)

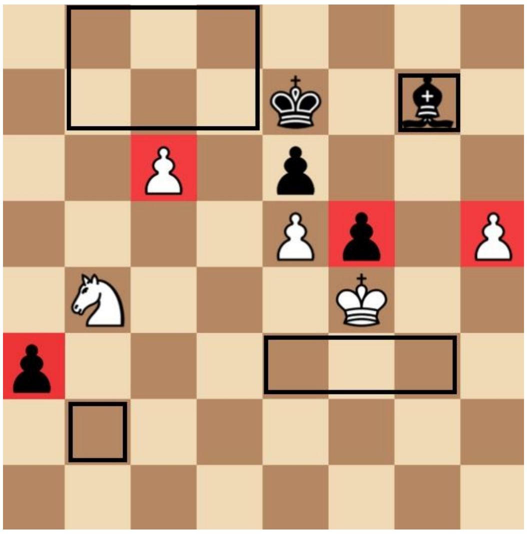


Fig.47. КРППП (3.1.2.)

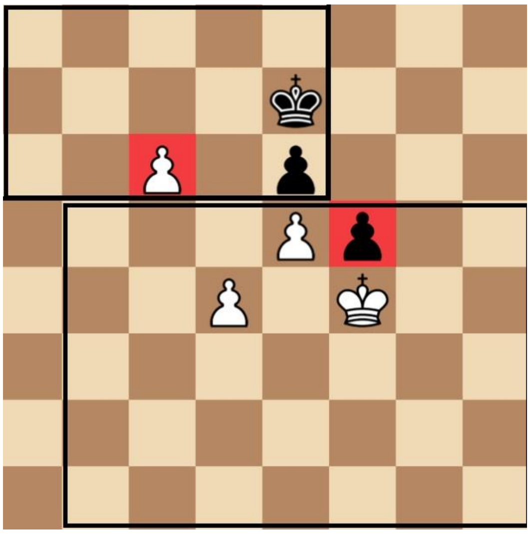


Fig.48. КВПП (3.1.3.)

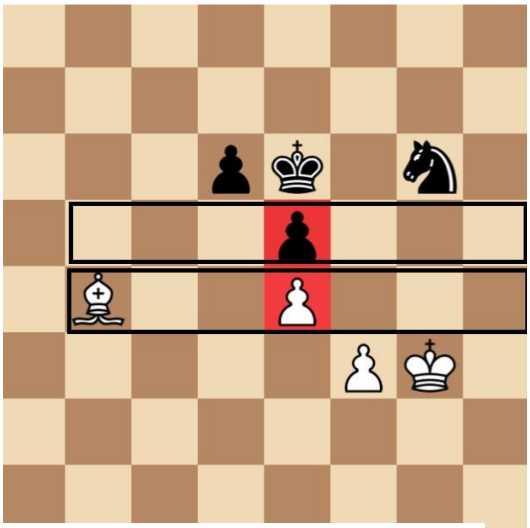


Fig.49. КРПБП (3.1.4.)

Fig.50. ВОЗБП, ВОЗЧП (3.2.1., 3.2.2.)

Fig.51. БП44 (3.3.1.)

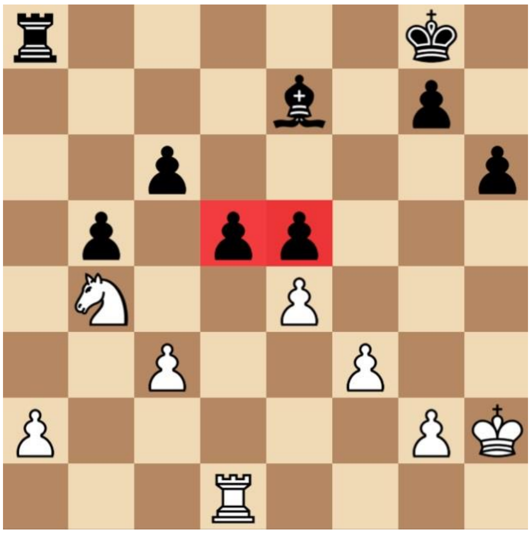


Fig.52. ЧП55 (3.3.2.)

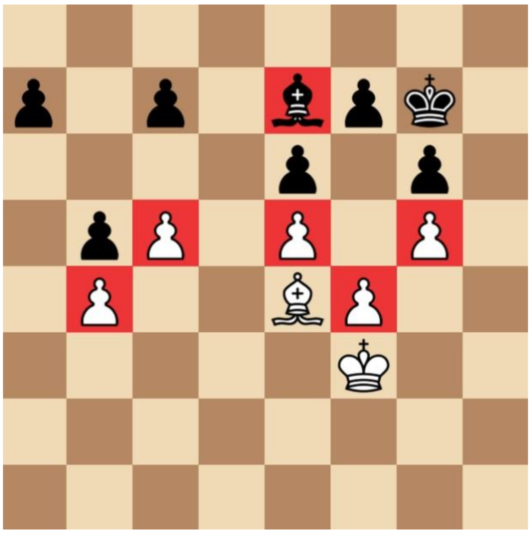


Fig.53. ПЗАС (3.6.2.)

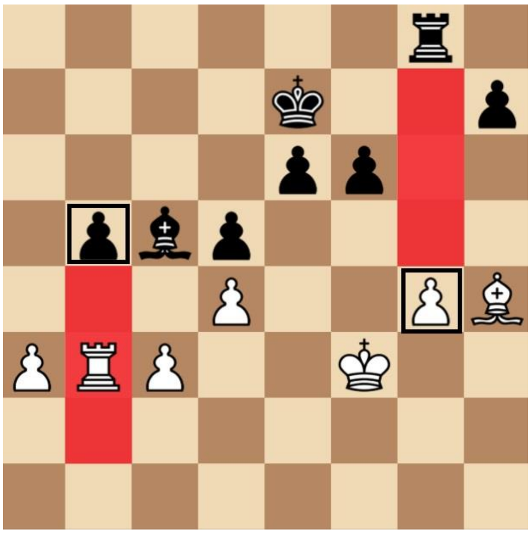


Fig.54. ИЗПОЛ (3.6.1.)

Результатом работы программ является переменная ДЕНТ, определенная следующим образом:

ДЕНТ = 0, при наличии фаланги на E4,D4 или на E5,D5,
1 - в противнда случае.

Программы обращаются к программе ФАПЕ(A,B).

Программы содержат по 23 операторов и занимают по 142 ячеек памяти.

3.4.Блок «Поле (C,D) является полем доступности или находится под ударом фигуры (A,B).

В дескрипторах данного блока проверяется факт, что рассматриваемое поле является полем доступности иди находится иод ударом конкретной фигуры, отличной от пешки.

Пешка выделяется по той причине, что для всех остальных фигур поля доступности и поля, находящегося под ударом, практически совпадают.

Дескриптор, выясняющий указанный факт для ферзя, принадлежит, в принципе, четвертому уровню, однако описывается в этом блоке для сохранения общности.

3.4.1 Поле (C,D) является полем допустимости или находится под ударом слона (A,B).

3.4.2 Поле (C,D) является полем допустимости или находится под ударом коня (A,B).

3.4.3. Поле (C,D) является полем допустимости или находится под ударом ладьи (A,B).

3.4.4. Поле (C,D) является полем допустимости или находится под ударом ферзя (A,B).

Поле (C,D) является полем допустимости или находится под ударом фигуры (А,Б), если рассматриваемая фигура имеет возможность в один ход достичь поля (C,D) (Рис. 58,45,59,60).

Соответствующие Ф-программы PPIUS(A,B,C,D), PDIUK(A,B,C,D), PDIUL(A,B,C,D), PDIUF(A,B,C,D) предназначены для определения, является ли поле (C,D) полем доступности, или находится под ударом коня, слона, ладьи иди ферзя, соответственно, стоящих на пункте (A,B).

Програшы работают следующим образом:

Вызывается соответствующая программа возможностей. Затем, проверяется, совпадают ли координаты клетки (C,D) с координатами хотя бы одной из клеток соответствующего массива. В зависимости от результатов проверки определенной переменной присваивается то или иное значение и программа завершает работу.

Результатом работы программ является переменная ДН.ю определенная следующим образом:

ДН = 0, если клетка (C,D) является полем доступности либо находится под ударом,
1, в противном случае.

Программы обращаются в процессе работы к программам ПДУЗС(C,D), ПДУЗК(C,D), ПДУЗЛ(C,D), ПДУЗФ(C,D) соответственно.

Программы содержат по 40 операторов и занимают по 242 ячеек памяти. (PDIUK – 30 операторов и 105 ячеек памяти).

3.5. Блок "Взятие или защита".

Включенные в данный блок дескрипторы объединены по близости в шахматном смысле описываемых ими понятий. В них проверяется факт, что данная фигура, отличается от пешки и короля, угрожает взятием или защищает свою фигуру, стоящую на определенной клетке.

Дескриптор "Взятие иди защита ферзем" фактически принадлежит четвертому уровню, однако описан в этом блоке, чтобы не нарушать общности. То же можно отнести к дескриптору "Взятие или защита королем", прияадлежащему пятому уровню. (Рис. 62,63,64,65).

3.5.1. Взятие или защита слоном.
3.5.2. Взятие или защита конем.
3.5.3. Взятие или защита ладьей.
3.5.4. Взятие или защита ферзем.

Соотвртствугагме Ф-прогршшы ВИЗС(A,B,C,D), ВИЗК(A,B,C,D), ВИЗЛ(A,B,C,D), ВИЗФ(A,B,C,D) предназначены для выявления факта, угрожает ли данная фигура, стоящая на (A,B) взятием фигуре, стоящей на (C,D) или защищает ее. Координаты пунктов, на которых стоят фигуры, указываются в качестве формальных параметров программ.

Программы работает следующим образом:

Вызывается по координатам (A,B) соответствующая программа из блока "Возможности фигур" (2.4). Затем координаты пункта (C,D) проверяются на совпадение с соответствующими элементами массивов, выработанных указанной выше программой.

В зависимости от результатов проверки определенной переменной присваиваются то или иное конкретное значение и программа завершает работу.

Результатом работы программ является переменная ДН, определенная следующим образом:

ДН = 0, если фигура (C,D) атакована или защищена фигурой (A,B),
1, в противном случае.

Программы содержат по 29 операторов и занимают по 188 ячеек памяти.

Программы в процессе работы обращаются к подпрограмме-функции F(A,B) и к одной из програш ПДУЗС, ПДУЗК, ПДУЗЛ, ПДУЗФ, соответственно.

3.6. Блок "Атака на важный пункт или поле".

В этом блоке сгруппированы дескрипторы, описывающие нападение на важный пункт или поле,

3.6.1. Легкая фигура (A,B) бьет на дырку противника (C,D).

Соответствующая Ф - программа ЛФБДР(A,B,C,D) предназначена для проверки факта, что легкая фигура (A,B) бьет на поле (C,D) , являющееся дыркой противника. (Рис. 56).

Работа программы начинается с проверки, является ли поле дыркой противоположного цвета. Проверка производится обращением к программе ДЫР(A,B) по координатам (C,D).

Если (C,D) не дырка, то программа завершает работу, в противном случае выписываются координаты клеток, находящихся под ударом легкой фигуры (A,B) и для каждой клетки проверяется совпадение ее координат с (C,D). Выписывание производится обращением к программам ПДУЗК или ПДУЗС в зависимости от кода фигуры.

В зависимости от результатов проверки определенной переменной присваивается то или иное значение и программа завершает работу.

Результатом работы программы является переменная ДН, определенная следующим образом:

0, если белая легкая фигура бьет на дырку черных,
ДН = 1, если черная легкая фигура бьет на дырку белых,
2, в противном случае

Библиотека внутренних программ:
1.ПДУЗК 2.ПДУЗС 3.ДЫР.
Программа ЛФБДР занимает 240 ячеек памяти и содержит 40 операторов.

3.6.2. Ладья (A,B) бьет на открытую вертикаль.

Соответствующая Ф -программа ЛАБФВ(A,B) проверяет, бьет ли ладья, стоящая на (A,B) какую-либо клетку открытой вертикали. (Рис. 57).

Программа работает следующим образом.

Выписываются координаты полей находящихся под ударом ладьи с помощью обращения к подпрограмме ПДУЗЛ по координатам ладьи. Затем для каждого поля, не лежащего на одной вертикали с (A,B) проверяется, принадлежит ли оно открытой вертикали. Проверка производится обращением к подпрограмме ОТВЕР по координатам рассматриваемого поля. В зависимости от результатов проверки определенной переменной присваивается то или иное значение и программа завершает работу.

Результатом работы программы является переменная ДУНТ, определенная следующим образом:

0, если белая ладья (A,B) бьет на открытую вертикаль,
ДУНТ = 1, если черная ладья (A,B) бьет на открытую вертикаль,
2, в противном случае.

Библиотека внутренних программ:
1. ПДУЗЛ. 2. ОТВЕР.

Программа ЛАБФВ(A,B) занимает 180 ячеек памяти и содержит 35 операторов.

3.7. Блок «Владение важными линиями».

Дескрипторы этого блока выявляют факт владения важными в определенном смысле линиями.

3.7.1. Владение краевыми горизонталями.

Соответствующая Ф - программа ФВГ81(А,В) проверяет следующий факт: если король одной из сторон, стоящий на (А,В), находится на 1-ой (2-ой) или 7-ой (8-ой) горизонталях (соотвотствонно, для белых и черных), то на этих же горизонталях стоит хотя бы одна неатакованная тяжелая фигура противника. В этом случае считаем, что противник владеет соответствующей краевой горизонталью. (Рис. 55).

Координаты короля задаются в виде параметров программы.

Программа работает следующим образом:

Обращощением к программе ГОРИЗ по соответствующей координате короля порождается нужная горизонталь и проверяется стоит ли на ней тяжелая фигура. Затем, обращением к программе КИПИН по координатам найденной фигуры, проверяется, атакована ли рассматриваемая фигура. В зависимости от результатов проверок определенной переменной присваивается то или иное значение и программа завершает работу.

Результатом работы программы является переменная ДОЛ, определенная следующим образом:

0, если белая тяжелая фигура на горизонтали,
ДОЛ= 1, если черная тяжелая фигура на горизонтали,
2, в противном случае.

Библиотека внутренних программ:
1. КИПИН. 2. ГОРИЗ.
Программа ФВГ81 содержит 42 операторов и занимает 237 ячеек памяти.

3.7.2. Ладья (А,В) на открытой вертикали.

Соответствующая Ф - программа ЛАНОВ (А,В) предназначена для выяснения, стоит ли ладья (А,В) на открытой вертикали. Координаты ладьи задаются в качестве параметров программы.

Работа программы заключается в следующем:

Обращением к программе ОТВЕР по абсциссе ладьи проверяется, является ли вертикаль открытой. Предварительно содержимое клетки доски, на которой стоит ладья приравнивается к нулю, а после проверки восстанавливается. В зависимости от результатов проверки определенной переменной присваиваются некоторые значения и программа завершает работу.

Результатом работы программы является переменная ДИНТ, определенная, следующим образом:

0, если белая ладья - на откр.вертикали
ДИНТ = 1, если черная ладья - на откр.вертикали
2, если на вертикали есть фигуры
3, если на клетке не ладья.

В качестве подпрограмм данная программа использует Ф -программы ОВЕРТ(А) и F(А,В).

Программа ЛАНОВ содержит 28 операторов и занимает 173 ячеек памяти.

3.8. Блок «Перегруженные фигуры».

Включенные в данный блок дескрипторы проверяют, что определенная фигура одной из сторон перегружена.

Понятие перегруженности фигур не имеет прямого эквивалента в шахматах, однако часто используется при оценках позиций. У нас сделана попытка уточнить катим-то образом сущность этого понятия учитывая силу фигур разных типов.

В основе понятия «перегруженности» отдельной фигуры лежат такие важные с точки зрения шахматной стратегии характеристики этой фигуры, как защита своих фигур, защита пешек, в особенности проходных и типа «слабых», защита своего короля, защита важных и слабых полей и пунктов на доске, атака на слабые поля и пункты противника.

3.8.1. Слон (A,B) перегружен.
3.8.2. Конь (A,B) перегружен.
3.8.3. Ладья (A,B) перегружена.
3.8.4. Ферзь (A,B) перегружен.

Следующие условия предназначены для проверки перегруженности фигуры:

1. Фигура (A,B) стоит на открытой линии.
2. Клетка (A,B) является клеткой из пояса своего короля.
3. Фигура стоит на одной линии со своим королем и клетки между ними свободны.
4. Фигура защищает хотя бы одни свою фигуру (кроме короля и пешек).
5. Фигура защищает свою проходную пешку.
6. Фигура защищает свою пешку типа «слабых».

Слон перегружен, если выполнены хотя бы две из условий 1, 2, 3, 4, 5. 6.
Конь перегружен, если выполнены хотя бы две из условий 1, 2, 3, 4, 5. 6.
Ладья перегружена, если выполнены хотя бы три из условий 1 – 6.

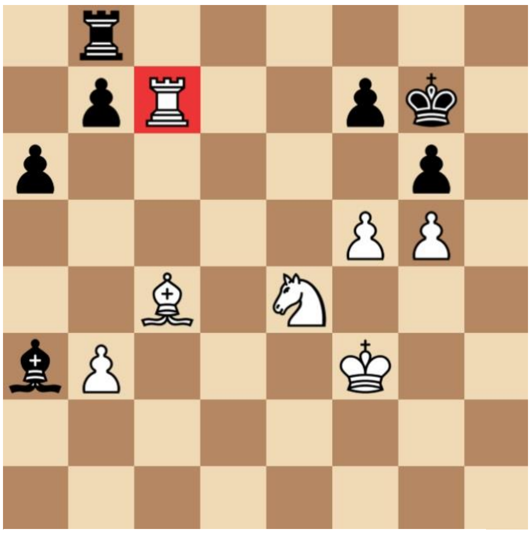


Fig.55. ОВГ81 (3.7.1.)

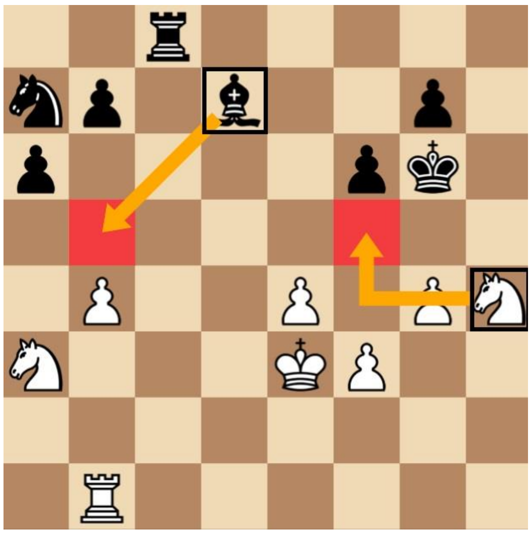


Fig.56. ЛФБДР (3.6.5.)

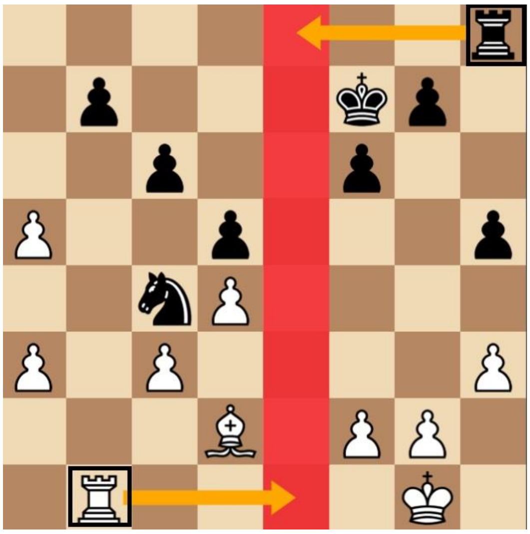


Fig.57. ЛАБОВ (3.6.4.)

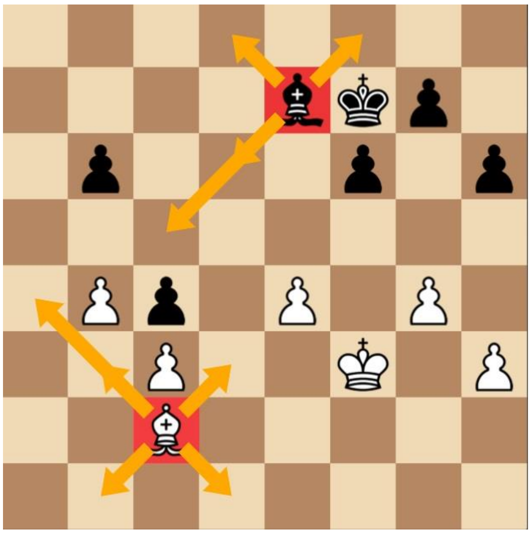


Fig.58. PDIUS (3.4.2.)

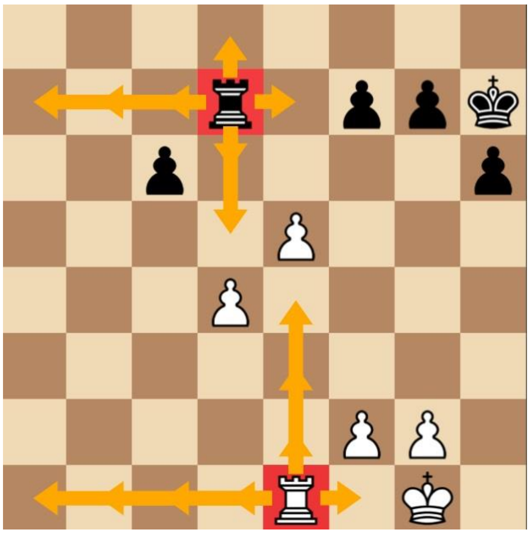


Fig.59. PDIUL (3.4.3.)

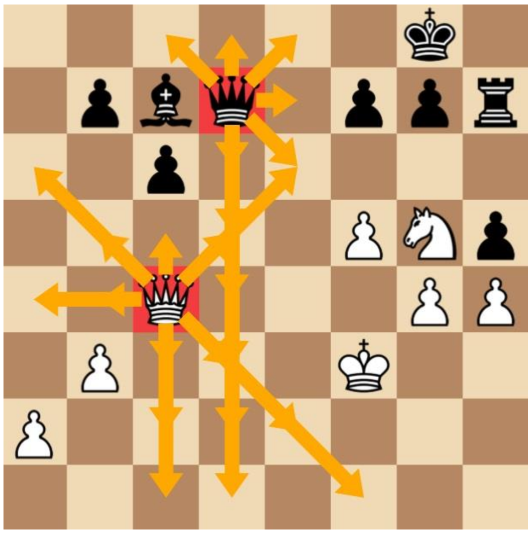


Fig.60. PDIUF (3.4.4.)

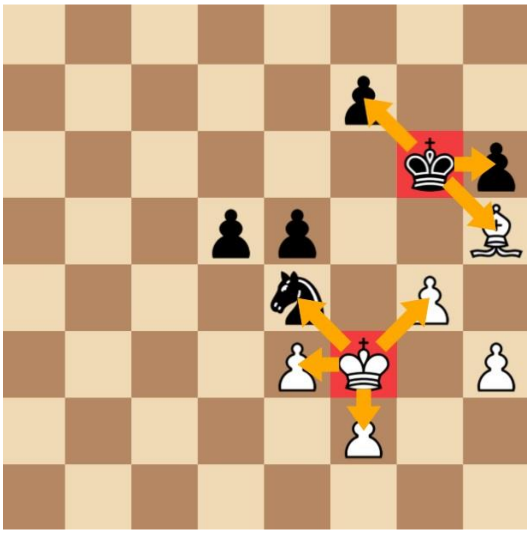


Fig.61. ВИЗКР (3.5.5.)

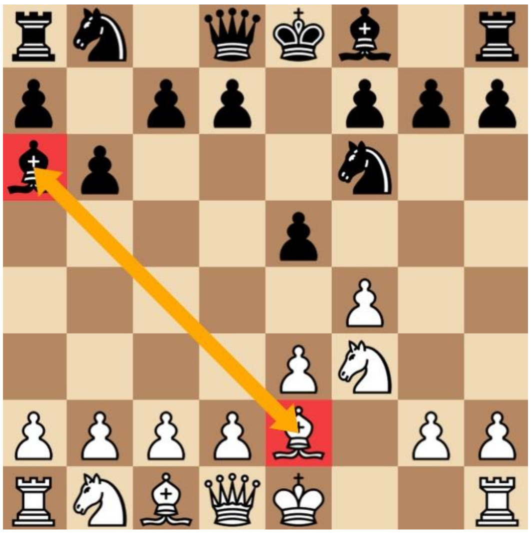


Fig.62. ВИЗС (3.5.1.)

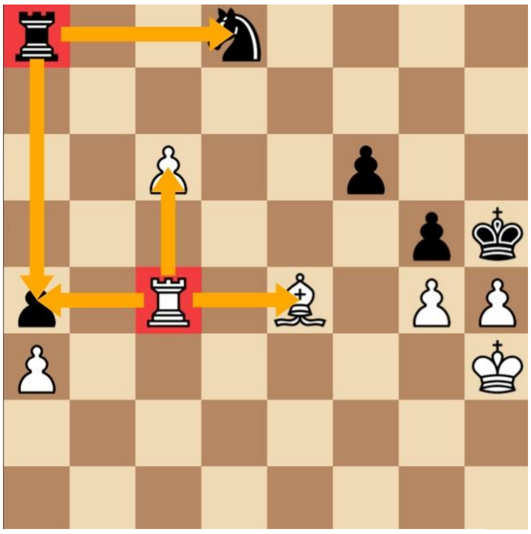


Fig.63. ВИЗЛ (3.5.2.)

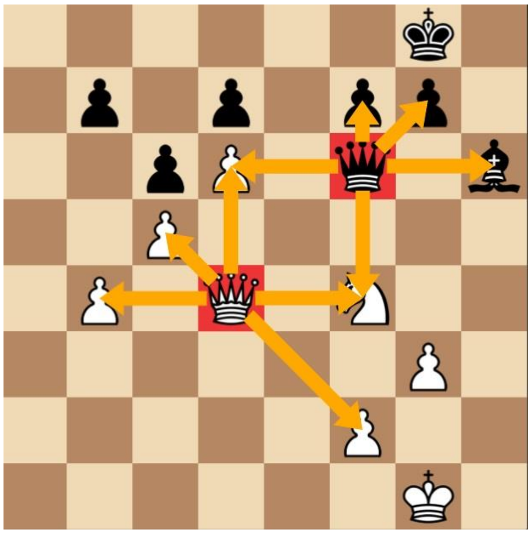


Fig.64. ВНЗФ (3.5.3.)

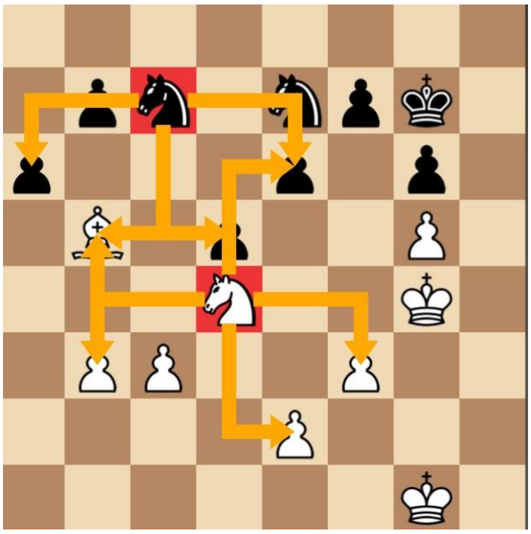


Fig.65. ЗИЗК (3.5.2.)

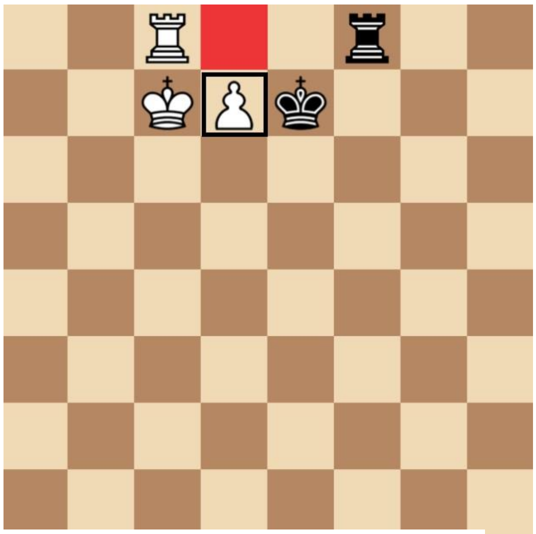


Fig.66. ОППППП (4.1.1.)

Ферзь перегружен, если выполнены хотя бы 4 из условии 1-6.

Таким образом, фигура считается перегруженной, если количество возложенных на нее функций больше определенного. (Рис. 71,72,73,74).

Соответствующие Ф - программы ПЕРС(А,В), ПЕРК(А,В), ПЕРЛ(А,В), ПЕРФ(А,В) предназначены дан проверки, перегружена ли рассматриваемая фигура. Координаты рассматриваемой фигуры задаются в качестве параметров программ.

Работа программ заключается в проверке указанных выше условий. В зависимости от того, выполнено или нет необходимое количество условий определенной переменной присваивается то или иное значение, и программа завершает работу.

Условие 1 проверяется обращением по координатам рассматриваемой фигуры к соответствующей прогремме из блока 2.1. – «Открытые линии».

Для проверки условия 2 вызывается программа ПОЯК по координатам короля и производится покоординатное сравнение всех клеток полученного множества с клеткой (А,В) .

Условие 3 проверяется обращением к программе ОТЛИН по координатам рассматриваемой фигуры и короля, с последующей проверкой клеток отрезка на занятость.

Условие 4 проверяется обращением в соответствующей программе блока 3.5. - "Взятие или защита".

Для проверки условий 5,6 выписываются координаты всех пешек, среди них выделяются (обращением к программам ПРОП, ИЗОП, ОТП, СЛАП, СДВП) проходные и слабые. Затем обращением к соответствующий программе блока 3.5. проверяется защищенность данной пешки рассматриваемой фигурой.

Результатом работы программ является переменная ПЕР, определенная следующим образом:

0, если белая фигура (А,В) перегружена,
ПЕР = 1, если черная фигура (А,В) перегружена,
2, в противном случае.

Библиотека внутренних программ:

1. ПОЯК	6. ОТЛИН	11. ПДУЗК	16. ПРОП
2. ОВЕРТ	7. ОТВЕР	12. ПДУЗС	17. ИЗОП
3. ОГОРЗ	8. ОТГОР	15. ПДУЗЛ	18. ОТП
4.ОЛЕДИ	9. ОТЛЕД	14. ПДУЗФ	19. СДВП
5.ОПРДИ	10. ОТПРД	15. F	20. СЛАП
21.СПИБФ	22.СПИЧФ		

Память:

ПЕРС - 157 операторов
86 ячеек памяти

ПЕРК - 91 операторов
369 ячеекк памяти

ПЕРЛ - 91 операторов
361 ячеек памяти

ПЕРФ - 91 операторов
358 ячеек памяти

3.9. Двойной удар

Дескрипторы данного блока сгруппированы по идентичности выполняемых ими функций. В них проверяется факт, что у определенной фигуры одной из сторон есть двойней удар.

Понятие "двойного удара" является у нас попыткой формализации одноименного понятия из шахматной теории. Понятие это в шахматах используется довольно часто, иногда под названием "вилка" однако, в конечном счете, точного его определения в теории шахмат не дается.

В основе понятия "двойной удар фигурой" лежит одна из основных идей шахматной теории: одновременная атака конкретной фигурой на две или более неудобно расположенные, более ценные или просто незащищённые фигуры противника. Понятие важно тем, что в ходе игры обычно приводит к улучшению позиционной или материальной оценки стороны наносящей двойной удар.

3.9.1. Двойной удар пешкой.
3.9.2. Двойной удар слоном.
3.9.3. Двойной удар конем.
3.9.4. Двойной удар ладьей.

Будем говорить, что рассматриваемая фигура имеет двойной удар, если выполнены следующие условия:
1) Рассматриваемая фигура достаточно защищена в том смысле, что при взятии этой фигуры противник понесет материальные потери в результате размена на данном пункте.
2) Рассматриваемая фигура одновременно атакует две более ценные, более ценную и незащищенную, или же две незащищенные фигуры противника. При этом сама фигура не должна попасть под удар менее ценной или равной фигуры.
3) Рассматриваемая фигура атакует более ценную или незащищенную фигуру противника и одновременно шахует короля противника. (Рис.79,80,81,82,83).

Соответствующие Ф – программы ДУПЕ(А,В), ДУСЛ(А,В), ДУКО(А,В), ДУЛА(А,В) предназначены для выявления факта, что рассматриваемая фигура имеет двойной удар. Координаты рассматриваемой фигуры задаются в качестве формальных параметров.

Работа программы заключается в проверке указанных выше условий и присвоении определенной переменной того или иного значения в зависимости от результата проверки. Условие 1) проверяется обращением к программе КИПИН. Условие 2) для пешки проверяется непосредственно, а для фигур обращением к программам ПДУЗК, ПДУЗС или ПДУЗЛ по координатам рассматриваемой фигуры с последующей проверкой ценности фигур, находящихся под ударом. При проверке условия король рассматривается как наиболее ценная фигура и производятся действия, аналогичные проверке условия 2.

Результатом работы программ является переменная НИД, определенная следующим образом:

0, если у белой фигуры есть двойной удар,
НИД = 1, если у черной фигуры ость двойной удар,
2, в противном случае.

Библиотека внутренних программ:

1.ПДУЗС	4.F	6.КИПИН	8. СПИБФ
2.ПДУЗК	5.H		9. СПИЧФ
3.ПДУЗЛ			

Память:
Д У П Е - 46 операторов
345 ячеек памяти
Д У С Л - 84 операторов
530 ячеек памяти
Д У К 0 - 84 операторов
530 ячеек памяти
ДУЛА - 82операторов
512ячеек памяти

5.9.5. Двойной удар ферзем.

Скажем, что ферзь имеет двойной удар, если выполняются следующие условия:
1)Ферзь не атакован фигурами противника.

2)Ферзь одновременно атакует хотя бы две незащищенные фигуры противника.
3)Ферзь атакует хотя бы одну незащищенную фигуру противника и одновременно шахует короля противника.

Соответствующая Ф - программа ДУФЕ (А,В) предназначена для проверки, имеет ли ферзь двойной удар. Координаты ферзя задаются в виде параметров программы.

Работа программы заключается в проверке указанных выше условий, По результатам проверки определенной переменной присваиваемся то или иное значение, и программа завершает работу. Условие 1) проверяется обращением к программе КИПИН по координатам ферзя. Условие 2) проверяется обращением к программе ПДУЗФ с последующей проверкой фигур, находящихся под ударом, на защищенность. При проверке условия 3) король рассматривается как наиболее ценная фигура, и производятся действия, аналогичные проверке условия 2.

Результатом работы программы являемся переменная НИД, определенная следующим образом:

0, если белый ферзь имеет двойной удар,
НИД = 1, если черный ферзь имеет двойной удар,
2 - в противной случае.

Библиотека внутренних программ:
1.ПДУЗФ 5. СПИБФ
2.КИПИН 4.F 6. СПИЧФ

Программа ДУФЕ содержит 79 операторов и занимает 480 ячеек памяти.

3.10. Блок «Фигура связана»

Включенные в этот блок дескрипторы объединены по идентичности определяемых ими функций. В них проверяется факт, что определенная фигура одной из сторон связана.

Понятие «связки» у нас является попыткой формализации одноименного понятия шахматной теории. Это понятие довольно часто используется в шахматах и имеет точное определение в шахматной теории.

Основой понятия "связка фигуры" является атака на фигуры противника, которые прикрывают по какой-либо линии на доске незащищенные фигуры своего цвета или своего короля.

Понятие важно тем, что в ходе игры приводит, как правило, к ухудшению позиции или материальному проигрышу стороны, имеющей связанные фигуры.

3.10.1. Слон (А,В) связан.
3.10.2. Конь (А,В) связан.
3.10.3. Ладья (А,В) связана.
3.10.4. Ферзь (А,В) связан.
3.10.5. Пешка (А,В) связана.

Фигуру назовем **связанной**, если имеют место следующие условия:
1) фигура стоит на одной линии Н со своим королем или своей незащищенной фигурой и клетки между ними свободные;

2) по этой же линии Н фигура атакована противником, причем атакующая фигура либо сама атакована более ценными фигурами, но достаточно защищена, либо вообще не атакована. (Рис. 75,76,77,78).

Соответствующие Ф - программы СЛОС(А,В), КОНС(А,В), ЛАДС(А,В), ФЕРС(А,В) предназначены для проверки, находится ли данная фигура под связкой. Координаты рассматриваемой фигуры задаются в виде параметров программ.

Работа программ заключается в следующем: проверяется выполнение указанных выше условий, и, в зависимости от результатов проверки, определенной переменной присваивается то или иное значение. После этого программы завершают работу.

Для проверки условия 1) (обращением к соответствующим программам блока 1.1. "Линии" по координатам рассматривавшей фигуры) порождаются нужные линии, проходящие через клетку, занятую рассматриваемой фигурой. Затем проверяется, стоит ли на той же линии незащищенная фигура или король своего цвета, и свободны ли клетки между ними и рассматриваемой фигурой.

Проверка условия 2) также начинается просмотром порожденных линий. Выясняется, стоит ли на них соответствующая фигура противника, т.е. атакована ли рассматриваемая фигура. Затем обращением к программе КИПИН по координатам атакующей фигуры проверяется, достаточно ли она защищена и атакована ли только более ценными фигурами.

Результатом работы программ является переменная СВ, определенная следующим образом:

0, если белая фигура связана,
СВ = 1,если черная фигура связана,
2, в противном случае.

Библиотека внутренних программ:
1. ВЕРТ 4. ПРАДИ
2.ГОРИЗ 5. КИПИН
3.ЛЕДИ 6. F

Память:
СЛОС - 201 операторов 970 ячеек памяти.
КОНС - 11 операторов 87 ячеек памяти.
Л А Д С - 347 операторов 2070 ячеек памяти.
Ф Е Р С - 11 операторов 87 ячеек памяти.

3.11. Блок «Разное».

3.11.1. Изолированная пешка (А,В) – на открытой линии.

Соответствующая Ф - программа ИЗПОЛ(А,В) предназначена для выявления факта, что изолированная пешка (А,В) находится на «открытой линии», что эквивалентно выполнению следующих условий :
1) на клетках (А,К), К = В+1,...,8, нет пешек противника.
2) пешка (А,В) - не проходная. (Рис. 54)

Работа программы начинается проверкой пешки (А,В) на изолированность, которая производится обращением к программе ИЗОП(А,В) по координатам рассматриваемой пешки. Если пешка (А,В) не изолированная, программа завершает работу. В противном случае вызывается программа ПРОП по тем же координатам для проверки, является ли пешка проходной. Если пешка не проходная и одновременно выполняется условие 1), то определенной переменной присваивается некоторое значение. В противном случае та же переменная получает иное значение и программа завершает работу.

Результатом работы программы является переменная НПО, определенная следующим образом:

0, если условия выполнены для белой пешки,
НПО = 1, если условия выполнены для черной пешки,
2, если условия не выполнимы.

Программа ИЗПОЛ использует в качестве подпрограмм Ф - программы ПРОП(А,В) и ИЗОП(А,В).

Программа ИЗПОЛ занимает 161 ячеек памяти и содержит 28 операторов.

3.11.2 Пешки заграждают путь слону (А,В).

Пешки заграждают путь слону противника, стоящему на клетке (А,В), есливыполнены следующие условия:

1) на доске хотя бы 4 пешки и слон противника.
2) пешки образуют цепи одного цвета с полем, на котором стоит слон противника, причем общая длина цепей не меньше 4.
3) ордината самой нижней (для белых) ила самой верхней (для черных) пешки цепи строго меньше или, соответственно, строго больше ординаты слона противника (Рис.53).

Координаты слона противника задаются в качестве параметров соответствующей Ф - программы ПЗАС(А,С).

Программа последовательно проверяет условия, указанные выше. Для проверки первого условия вызываются программы СПИБФ и СПИЧФ, второго - ЦБЕП или ЦЧЕП в зависимости от цвета пешек. Если все условия выполнены, то определенной переменной присваивается некоторое значение. В случае, если хотя бы одно из условий не выполнено, той же переменной присваивается другое значение, и программа завершает работу.

Результатом работы программы является переменная ДИЛ, определенная следующим образом:

0, если белые пешки заграждают путь слону черных,
ДИЛ = 1, если черные пешки заграждают путь слону белых,
2, если пешки не заграждают путь слону.

Программа ПЗАС в процессе работы использует программы СПИБФ, СПИЧФ, ЦБЕП, ЦЧЕП.

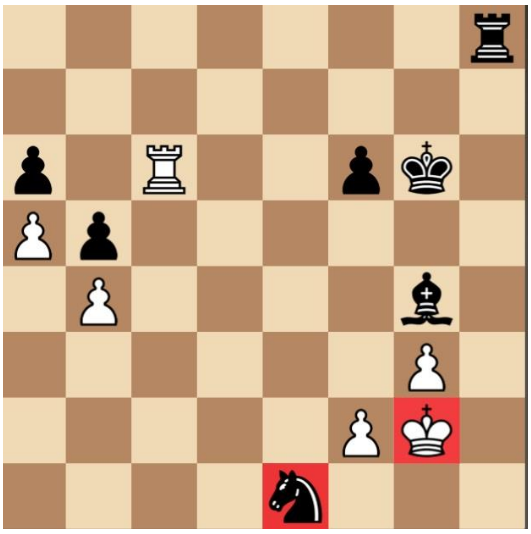


Fig.67. ШАХ (4.2.1.)

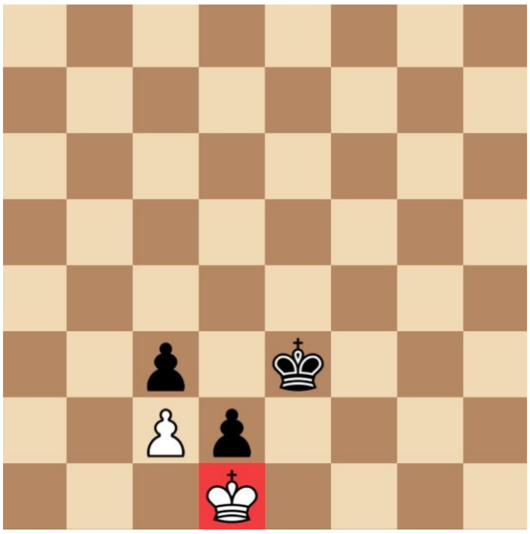


Fig.68. ПАТ (4.2.2.)

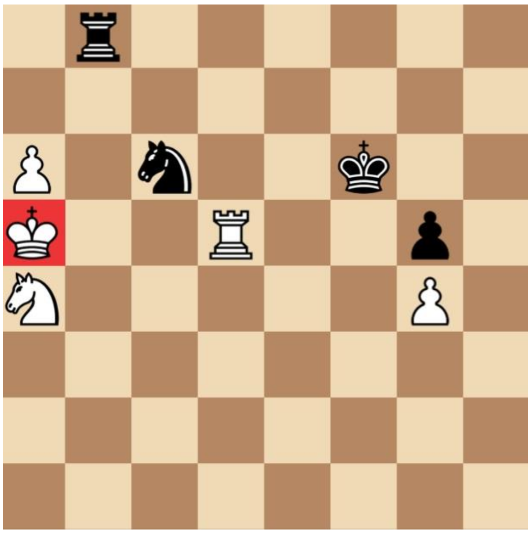


Fig.69. МАТ (4.2.1.)

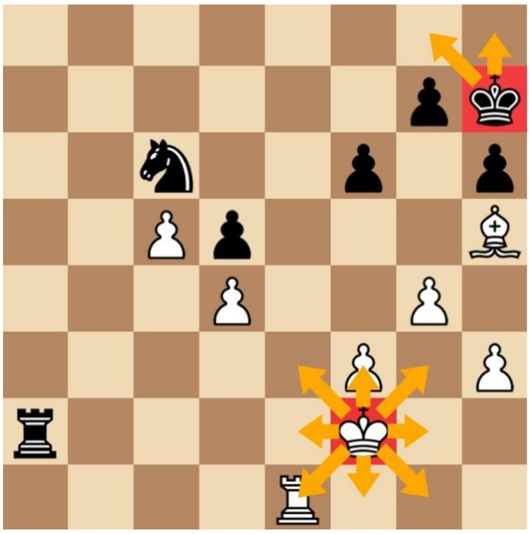


Fig.70. ПЛУКР (2.4.5.)

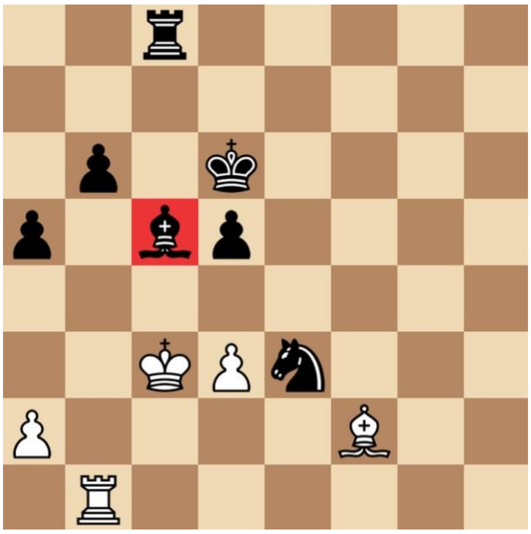


Fig.71. ПЕРС (3.8.1.)

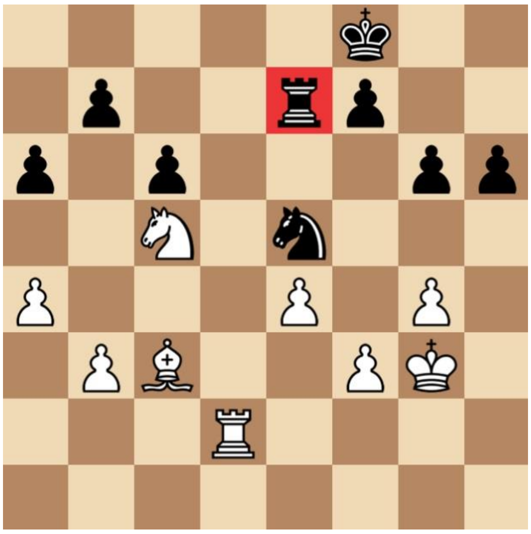


Fig.72. ПЕРЛ (3.8.3.)

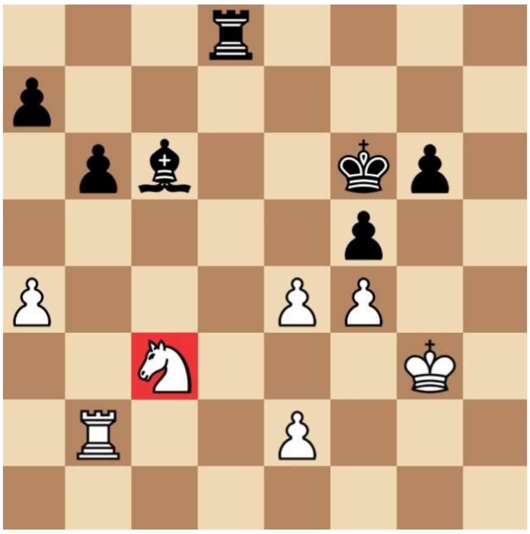


Fig.73. ПЕРК (3.8.2.)

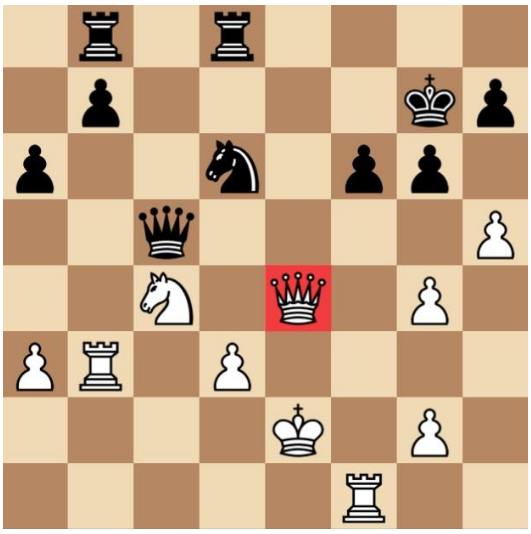


Fig.74. ПЕРФ (3.8.4.)

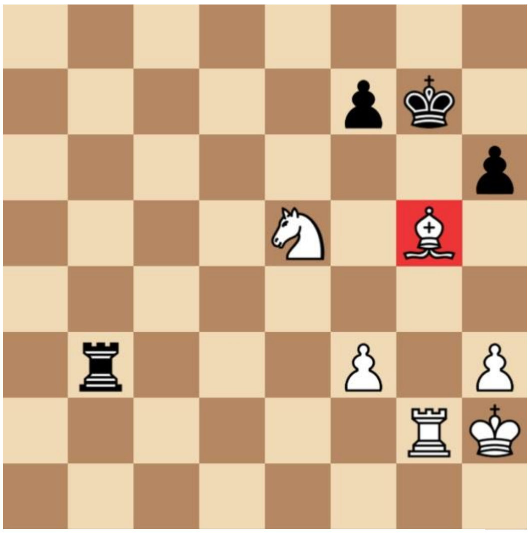


Fig.75. СЛРС (3.10.1.)

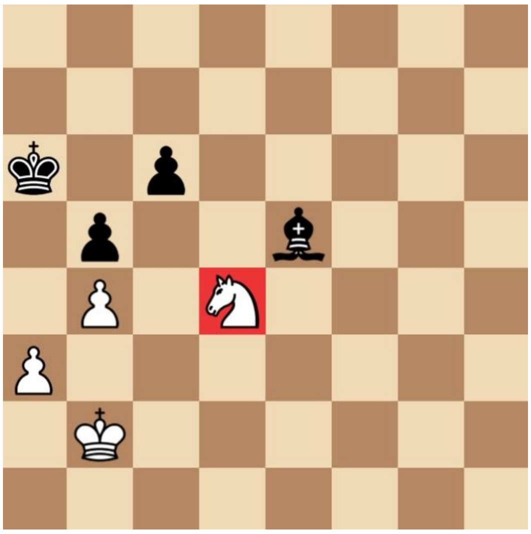


Fig.76. КОНС (3.10.2.)

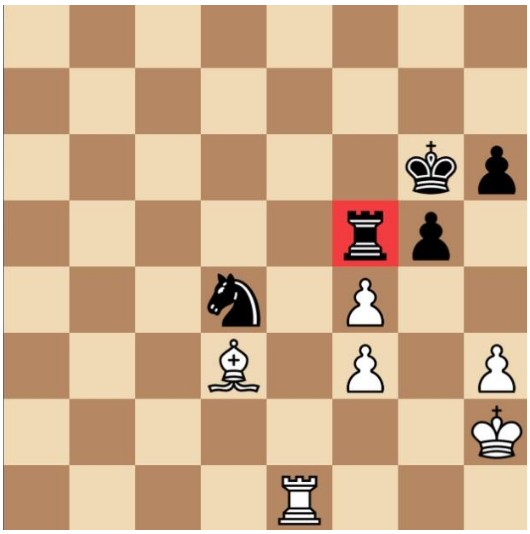


Fig.77. ЛАДС (3.10.3.)

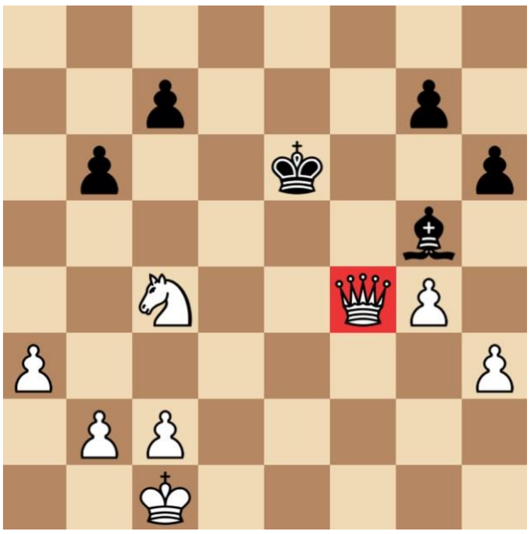


Fig.78. ФЕРС (3.10.4.)

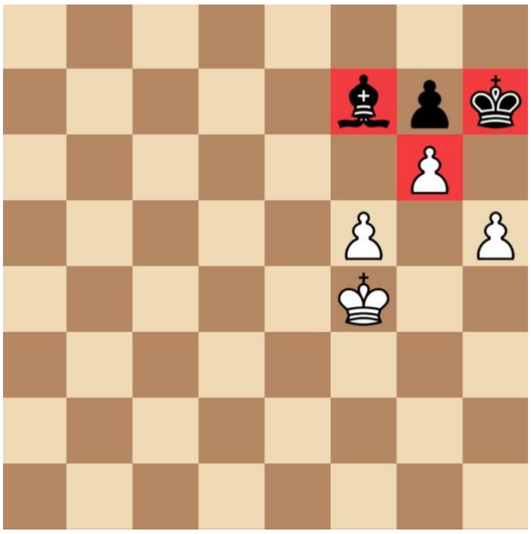


Fig.79. ДУПЕ (3.9.1.)

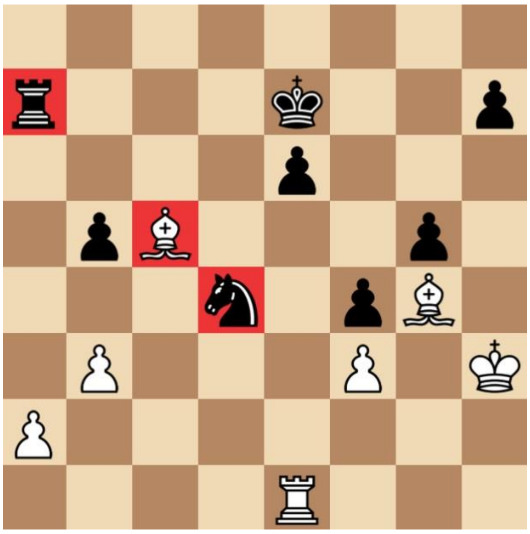


Fig.80. ДУСЛ (3.9.2.)

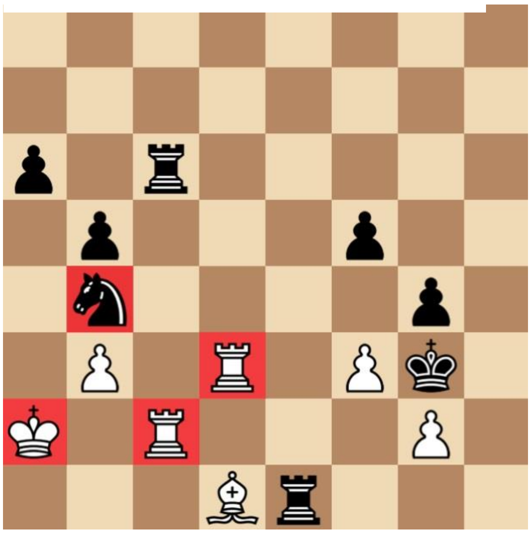


Fig.81. ДУКО (3.9.3.)

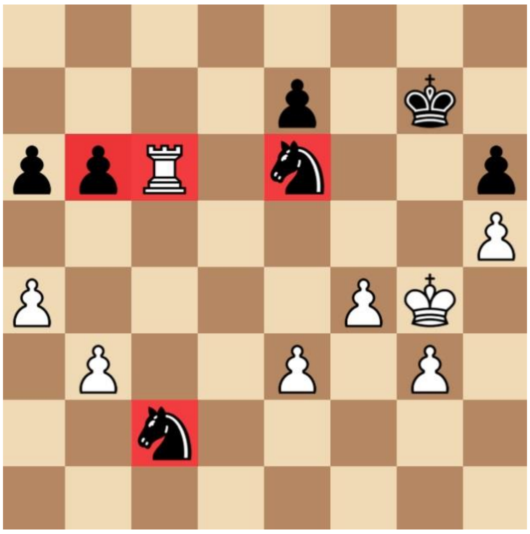


Fig.82. ДУЛА (3.9.4.)

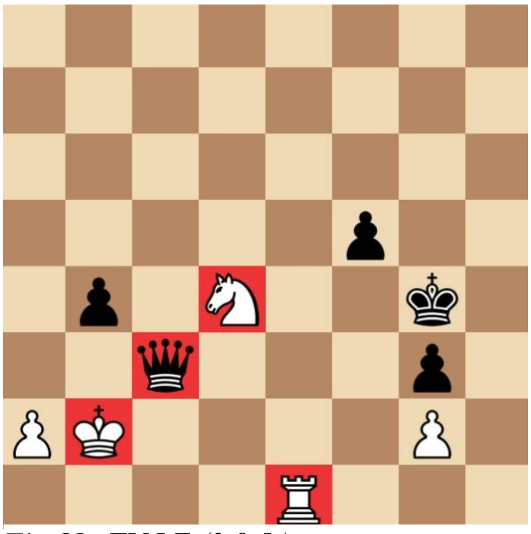


Fig.83. ДУФЕ (3.9.5.)

IV. УРОВЕНЬ

4.1. Захват важных пунктов или полей.

4.I.I. Захват доля превращения проходной пешки; ОПППП(A,B).

Будем говорить, что превращение проходной пешки реально, если выполняются следующие условия:

1. Проходная пешка стоит на 7-ой или 6-ой (для черных, соответственно, 2-ой или 3-ей) горизонтали.
2. Количество своих ударов на поле превращения пешки больше, чем количество ударов противника, а взятие ее приводит к материальному проигрышу для противника. (Рис. 66)

Соответствующая Ф - программа ОПППП(A,B) предназначена для выявления факта, что превращение проходной пешки одной иа сторон реально. Координаты пешки задаются в виде параметров программы.

Программа работает следующим образом:

Проверяются условия реальности превращения проходной пешки и производится материальная оценка фигур противника и обоих фигур, атакующих поле превращения пешки. В зависимости от результатов выходной переменной присваивается то или иное значение и программа завершает работу.

Проверка условия 1 производится с помощью обращения к программе ПРОП по координатам пешки. Затем обращением к программе ПППП по тем же координатам формируются координаты поля превращения рассматриваемой пешки.

Условие 2 проверяется обращением к программе КИПИН по координатам поля превращения. Материальная оценка также производится по результатам работы программы КИПИН.

Результатом работы программы является переменная НОДИ, определенная следующим образом:

0, если обоснован захват поля превращения для белых,
НОДИ = 1, если обоснован захват поля превращения для черных,

2, в протвном случае.

Библиотека внутренних программ:

1. ПРОП 3. ПППП
2. 4. КИПИН

Программа ОПППП(А,В) содержит 33 оператора и занимает 197 ячеек памяти.

4.2. Блок "Критические ситуации".

Дескрипторы данного блока объединены по близости описываемых ими понятий в шахматном смысле. Они проверяют наличие на доске критической ситуации, под которой понимается пат или мат.

4.2.1. Королю (А,В) объявлен мат; МАТ(А,В).

Будем говорить, что королю, стоящему на поле (А,В), дан мат, если выполнены следующие условия:

1. Король находится под шахом,
2. Множество полей доступности короля пусто.
3. Нет возможности закрыться от шаха или избавиться от него взятием фигуры, дающей шах. (Рис. 69).

Соответствующая Ф - программа МАТ(А,В) проверяет дан ли королю одной из сторон мат. Координаты короля задаются в виде параметров программы.

Работа программы заключается в проверке указанных выше условий и присваивании выходной переманной конкретных значений в зависимости от результатов проверки.

Первое условие проверяется обращением к программе ШАХ по координатам короля.

Второе условие проверяется с помощью обращения к программе ПДУКР по координатам короля.

Для проверки третьего условия выписываются поля доступности всех своих фигур списка (обращением к программам СПИБФ или СПИЧФ и к программам блока 2.4.) и затем проверяется совпадение хотя бы одного из этих полей с полем, лежащим между королем и шахующей фигурой (если она - не конь) на одной линии с ними.

После этого проверяется возможность взятия шахующей фигуры (обращением к программе КИПИН по координатам фигуры).

Результатом работы программы является переменная ОНД, определенная следующим образом:

0, если белому королю дан мат,
ОНД = 1, если черному королю дан мат,
2, в противном случае.

Библиотека внутренних программ:

1.СПИБФ	4.ПДУЗС	7.ПДУЗФ	10.ОТВЕР	13.НТЖЕ
2.СПИЧФ	5.ПДУЗК	8.ПДУКР	11.ОТГОР	14.ОТПРД
3.	6.ПДУЗЛ	9.ШАХ	12.	

Программа МАТ(А,В) занимает 533 ячейки памяти и содержит 75 операторов.

4.2.2. Королю (А,В) объявлен ПАТ; ПАТ(А,В).

Будем говорить, что у одной из сторон - п а т, если множество полей доступности и полей, на которых возможно взятие, каждой фигуры этой стороны, включая короля пусто. (Рис. 68).

Соответствующая Ф - программа ПАТ(А,В) проверяет, находится ли король одной из сторон в патовом положении. Координаты короля задаются в виде параметров программы.

Программа по спискам фигур (обращение к программам СПИБФ или СПИЧФ) проверяет наличие на доске фигур соответствующего цвета (кроме короля). В случае наличия фигур порождаются поля доступности каждой фигуры и проверяется равен ли 15 первый элемент соответствующего массива. Затем порождаются поля доступности короля, и обращением к программе КИПИН проверяется, находятся ли эти поля под ударом. В зависимости от результатов проверок выходной переменной присваивается то или иное значение и программа завершает работу.

Результатом работы программы является переменная ОДН, определенная следующим образом:

$$ОДН = \begin{cases} 0, \text{ если у белых пат,} \\ 1, \text{ если у черных пат,} \\ 2, \text{ в противном случае.} \end{cases}$$

Библиотека внутренних программ:

1.СПИБФ	4.ПДУЗС	7.ПДУЗЛ
2.СПИЧФ	5.ПДУЗФ	8.ПОЯК
3.	6.ПДУЗК	9.ШАХ

Программа ПАТ(А,В) содержит 132 оператора и занимает 730 ячеек памяти.

<u>4.3. Блок «Обращающиеся программы».</u>

В этот блок входят программы, являющиеся вспомогательными в том смысле, что они не вычисляют конкретных предикатов или функций, а служат лишь для обращения по коду фигуры к соответствующему элементу определенного блока.

<u>4.3.1. Угроза взятия фигуры (C,D) фигурой (A,B); УВФИ(A,B,C,D)</u>
<u>4.3.2. Защита фигуры (C,D) фигурой (A,B); ЗФФИ(A,B,C,D)</u>
<u>4.3.3. Клетка (C,D) – поле доступности фигуры (A,B); ПДФИ(A,B,C,D)</u>
<u>4.3.4. Клетка (C,D) – под ударом фигуры (A,B); ПУФИ(A,B,C,D)</u>
<u>4.3.5. Фигура (A,B) – перегружена; ФИПЕ(A,B)</u>
<u>4.3.6. Двойной удар фигурой (A,B); ДВОУД(A,B)</u>
<u>4.3.7. Фигура (A,B) – связана; ФИСВ(A,B)</u>

Соответствующие Ф - программы УВФИ(A,B,C,D), ЗФФИ(A,B,C,D), ВДФЙ ПДФИ(A,B,C,D), ПУФИ(A,B,C,D), ФИПЕ(A,B), ДВОУД(A,B), ФИСВ(A,B) служат для обращения по коду фигуры к определенной программе того или иного блока. Координаты фигуры задаются в виде параметров программ.

Программы выделяют код фигуры, стоящей на рассматриваемой клетке, т.е. содержимое данной клетки доски, а затем по полученному коду производился обращение к соответствующему блоку.

Библиотека внутренних программ:

УВФИ,ЗФФИ

1. ВИЗБП	5. ВИЗЛ
2. ВИЗЧП	6. ВИЗФ

3. ВИЗС 7. ВИЗКР
4. ВИЗК

ПУФИ, ПДФИ
1.ПУБП 4.ПДЧП 7.PDIUL
2.ПУЧП 5.PDIUS 8.PDIUF
3.ПДБП 6.PDIUK 9.PDUKR

ФИПЕ
1.ПЕРС 3.ПЕРЛ
2.ПЕРК 4.ПЕРК

ДВОЦД
1.ДУПЕ 4.ДУЛА
2.ДУСЛ 5.ДУФЕ
3.ДУКО

ФИСВ
1.СЛОС 3.ЛАДС
2.КОНС 4.ФЕРС

ПАМЯТЬ
УВФИ - 30 операторов
197 ячеек памяти.
ЗФФИ - 30 операторов
194 ячеек памяти.
ПУФИ - 30 операторов
194 ячеек памяти.
ПДФИ - 30 операторов
194 ячеек памяти.
ФИПЕ - 20 операторов
105 ячеек памяти.
ДВОЦД - 34 операторов
194 ячеек памяти.
ФИСВ - 20 операторов
105 ячеек памяти.

5.2.1. Поля доступности и удары королем; PDUKR(A,B,C,D)

Соответствующая - программа PDUKR(A,B,C,D) предназначена для проверки, является ли клетка (C,D) полем доступности короля (A,B), или находится ли она под его ударом. Координаты рассматриваемых клеток задаются в виде параметров программы.

Работа программы начинается с выписывания координат всех полей доступности короля и полей, находящихся под его ударом. Для этого программа обращается к программе ПДУКР. Затем производится покоординатное сравнение всех полученных клеток с клеткой (C,D), и в зависимости от этого выходной переменной присваивается соответствующее значение. Этим программа завершает работу.

Результатом работы программы является переменная ДН, определенная следующим образом:

0, если клетка (C,D) - поле доступности короля (A,B),
ДН = 1, если клетка (C,D) находится под ударом короля (A, B),
2, в противном случае.

В процессе работы программа PDUKR обращается к программе ПДУКР.

5.2.2. Взятие и зашита королем: ВИЗКР(A,B,C,D).

Соответствующая Ф - программа ВИЗКР(A,B,C,D) проверяет, имеет ли возможность король, стоящий на (A,B), взять фигуру противника, стоящую на (C,D), или защитить свою фигуру на пункте (C,D). Координаты короля и рассматриваемой клотки задаются в виде параметров программы.

Работа программы начинается выписыванием всех пунктов, на которых возможно взятие или защита королем. Для этого производится обращение к программе ПДУКР. Затем производится покоординатное сравнение полученных клеток с клеткой (C,D) и в зависимости от этого выходной переменной присваивается соответствующее значение. Этим программа завершает работу.

Результатом работы программы является переменная ДН, определенная следующим образом:

0, если возможно взятие королем,
ДН = 1, если возможна защита королем,
2, в противном случае.

В процессе работы программа ВИЗКР обращается к программе ПДУКР.
Программа ВИЗКР занимает 189 ячеек памяти и содеркит 29 операторов.

A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations **National Academy of Sciences of Armenia, IPIA, 1974-1980 research reports, Part 2.**

Chess Conceptual and Procedural knowledge
For expert strategy knowledge we distinct *conceptual* and *procedural* types that are represented, particularly, by *attributes, classifiers and rules*.
Attributes are kind of operators defined on the considered types of realities and having values in a given set of other realities. Particularly, they may be numbers of some range.
We distinct *basic attributes* and *composed* ones that use basic attributes to find their values.
Attributes with 0/1 or “false” / ”true” values are named *classifiers* and correspond to *concepts* and *goals.* Concepts are kind of human knowledge to identify and recognize realities.
Motives are attributes used to argue the preferences of some goals to pursue in the analyzed situation.
Strategies, plans and goals are expert knowledge to specify compositions of his actions in time.
Rules are kind of *if* $\boldsymbol{x}$ *– then* $\boldsymbol{y}$ operators to specify a procedure $\boldsymbol{y}$ for the realities that fit to the $\boldsymbol{x}$ requirements.

Оглавление

4. Конфигурации групп фигур.
5. Статические характеристики позиции.
6. Тактические действия групп фигур.
7. Возможности фигур.
8. Характеристики позиции короля.
9. Типы шахов.
1О. Типы преимуществ (перевесов).
11. Мотивы (предпосылки) каких-либо действий.
12. Типы позиций.
13. О дебютных стратегиях.
14. О темпе и инициативе в игре.
15. Типы действий (игры) сторон в каком-либо промежутке партии.
16. Типы фигурных центров.
17. Типы ходов.

1. АТАКА - КОНТРАТАКА

1. Атака полей фигурами.
2. Атака слабых полей фигурами.
3. Атака слабой (изолированной, отсталой,сдвоенной и т.д.) пешки.
4. Атака на важный пункт.
5. Фланговая атака.
6. Концентрическая атака.
7. Взрывающая атака.
8. Атака на слабые фигуры.
9. Атака на “перегруженные” фигуры.
10. Атака короля в центре, если свой - в центре.
11. Атака короля в центре, если свой (не) рокировавший - на левом (правом) фланге.
12. Атака с фланга на короля при односторонней (разносторонней) рокировке.
13.Атака с фланга на позиции рокировавшего (нерокировавшего) противника.
14. Атака на позицию короля, если свой король - в центре.
15.Атака с фланга на позицию короля, если свой король рокировал на тот (противоположный) фланг.
16. Атака короля тяжелыми (легкими) фигурами.
17. Взлом.
18. Взлом позиции короля посредством размена пешек (легких фигур),
19. Взлом позиции короля посредством жертвы пешек (легких или тяжелых фигур).

АТАКА

Под атакой (контратакой) подразумеваются коллективные действия одной из сторон, направленные на достижение каких-либо выгод (позиционных или материальных) на доске.
Элементарными случаями атаки являются угроза взятия фигуры противникà и удар фигурой на свободную клетку (поле).
Под это определение попадают также к о м б и н а ц и и.
Атаки на доске ведутся в двух направлениях:
1) атака на слабые пункты или поля в позиции противника;
2) атака на удачно расположенные и важные по назначению пункты противника.
В общем же случае атаки ведутся с целью достижения самого главного на доске - мата неприятельскому королю.

АТАКА ПОЛЕЙ ФИГУРАМИ

Имеются в виду действия, направленные на овладение важных на данном этапе игры полей на доске (в большинстве случаев - в позиции противника

1) атака соседних с неприятельским королем полей;
2) атака некоторых полей в лагере противника с целью ослабления его тыла;
3) атака подай, находящихся рядом и впереди изолированной пешки противника;
4) атака полей, уже не находящихся под воздействием пешек противника (эти поля слабы потому, что представляют возможность длительного и нередко опасного вторжения в лагерь противника).

АТАКА СЛАБЫХ ПОЛЕЙ ФИГУРАМИ

Атакуются поля в лагере противника, которые становятся слабыми в связи ñ маневрами на доске (продвижение пешек противника, невозможность непосредственной защиты противником определенных полей и т.п.).

Но даже среди слабых полей нас интересуют только те из них, которые можно использовать непосредственно. Другими словами, для каждого такого поля имеется хотя бы одна наша фигура, передвижение которой на это поле чревато для противника неприятными последствиями (хотя бы - в позиционном смысле и, разумеется, на некоторую глубину).

Возможный подход к формализации

Для каждого слабого поля в позиции противника проверяется возможность хода нашими фигурами на это поле и наличие ударов наших пешек туда же.

АТАКА СЛАБОЙ ПЕШКИ

Атака слабых пешек противника имеет смысл, проще всего, с точки зрения материального выигрыша.

А в общем случае:

1) на защиту слабых пешек противник обычно израсходует немало сил и времени (темпа);
2) слабые пешки "слабы", так как, в большинстве случаев, "нетруднодоступны";
3) при организованной атаке слабых пешек естественным образом достигается дальнейшее развитие наших фигур;
4)часто на этом пути достигаются "промежуточные" выгоды, не уступающие по своему значению выигрышу атакуемой пешки.

О с д в о е н н ы х п е ш к а х заметим следующее.

Сдвоенные пешки слабы, так как:

а) ограничена их мобильность;

если они и з о л и р о в а н н ы е, то, как правило в эндшпиле легко берутся;

б) из них трудно создать проходную;

в) из них трудно создать пешечную цепь,так как в результате приложенных по этому направлению усилий могут возникнуть и з о л и ро в а н н ы е пешки (динамическая слабость) либо понижается атакующая мощь уже построенной пешечной цепи.

Отсюда следует, что сдвоенные пешки нужно "принудить" к продвижению, так как при зтом только проявляется их д и н а м и ч е с к а я с л а б о с т ь;

г) с т а т и ч е с к а я с л а б о с т ь сдвоенных пешек проявляется не в процессе их продвижения, а когда на них обрушиваются наши пешки. В этом случае попытка противника защитить эти пешки приводит к позиционному или, что хуже, к материальному проигрышу.

Отсюда следует, что статическая слабость сдвоенных пешек используется путем прямой атаки на них, не боясь при этом, что "сдвоенность" их исчезнет, так как одна из них, как правило, останется слабой.

Таким образом, сдвоенные изолированные пешки, а такке сдвоенные пешки, которые более или менее мобильны, следует атаковать нашими пешками, "принуждая" при этом к продвижению еще не продвигавшихся сдвоенных пешек.

Возможный подход к формализации.

Для каждой слабой пешки противника вычисляется количество нападений на нее нашими фигурами и количество защит фигурами противника.

Другими словами, проверяется возможностъ взятия этой пешки нами (учитывая при этом ценности атакующих и защищающих фигур и порядок разменов, возникающих при взятии). Главный смысл атаки на важный пункт противника заключается в том, чтобы обезопасить (хотя бы временно) фигуру, занимающую этот пункт или чтобы открыть доступ к тому более ценному или незащищенному, что скрывается за этой фигурой.

Возможный подход к формализации.

Вычисляется количество всех угроз взятия и ударов на какой-либо важный (для противника) с некоторой точки зрения пункт противника, а конкретно:

1) на фигуры противника из пояса своего короля;
2) на проходные пешки противника;
3) на фигуры противника, защищающие поля превращения своих проходных пешек;
4) на центральные пешки противника;
5) на далеко продвинутую, опасную пешку противника;
6) на активную фигуру противника с целью ее связывания или отвлечения от важных событий и. т.п.

ФЛАНГОВАЯ АТАКА

Под фланговой атакой подразумеваются действия наших фигур на каком-либо фланге, направленные на достижение определенной цели (не обязательно на том же фланге).

О дебютных фланговых атаках.

1) Прочность собственных позиций в центре определяет успех предпринимаемых атак на фланге;
2) дебютные фланговые атаки - это важное дополнительное средство борьбы за центр, состоящие в отказе от лобовой атаки в пользу фланговых ударов и обходов. Таким путем осуществляется стремление уменьшить и ослабить воздействие фигур противника на центр, чтобы добиться тем превосходства в силах (применение - преимущественно в закрытых и полузакрытых дебютных позициях).

Фланговые атаки очень часто применяются также в середине игры. В этом случае на определенном фланге доски, в силу сложившейся там обстановки (слабости противника или превосходство наших сил) и при наличии определенной цели, часто бывает возможным проведение успешных действий (атаки) нашими фигурами.

Возможный подход к формализации

Проверяется возможность нанесения целенаправленного удара на определенном фланге нашими фигурами, часть которых уже находится на этом фланге, а другую часть легко можно перебросить туда с целью достижения нужного превосходства сил для успешного проведения атаки.

Концентрическая атака

Под этим подразумевается накопление (концентрация) угроз для завоевания пешки, во-первых, ради материального выигрыша и, во-вторых, для открытия линии. Противник пытается подвести столько защит, сколько мы накопили угроз. Тогда:

1) прогоняем защищающие фигуры;
2) обмениваем их;

3) отрезаем их от объекта защиты;
4) отвлекаем их от объекта защиты.

Взрывающая атака

Представляет из себя:
1) внезапную атаку с жертвой (комбинацию);
2) атаку на загребающую пешку или ее защищающую;
3) вторжение на 7-ую или 8-ую горизонталь;
4)обьявление мата королю противника или достижение жертвующей стороной, в крайнем случае, материального равновесия либо такого позиционного перевеса, который вознаграждает все жертвы.

Атака на слабые фигуры

Смысл атаки на слабую фигуру противника состоит в том, чтобы при возможности взять эту фигуру, с целью реализации этого преимущества на последующих этапах партии (в эндшпиле). При этом взятие должно быть подготовлено нужным образом, путь на него - своевременно начат и закончен, так как в противном случае это грозит хотя бы п о т е р е й т е м п а.

Атака на "перегруженные"фигуры

Смысл атаки на "перегруженную" фигуру противника заключается в том, что эта фигура, по определению, выполняет слишком много обязанностей в отношении других фигур того же цвета и некоторых важных полей в своем лагере.

Хорошо подготовленная атака на такую фигуру не сулит ничего хорошего противнику; в результате атаки может быть потеряна или эта фигура или другая из тех, которых она защищала или закрывала. В лучшем случае атака на "перегруженную" фигуру грозит противнику чувствительным п о з и ц и о н н ы м п р о и г р ы ш е м или же потерей темпа.

Атака КОРОЛЯ тяжелыми (легкими) фигурами

Под атакой короля противника тяжелыми (легкими) фигурами подразумеваются такие грозные действия нашими фигурами соответствующего типа или же их совместные усилия, когда события развиваются уже в глубоком тылу противника и иногда подворачиваются для противника роковыми последствиями (мат королю или крупный материальный проигрыш). Причем, как правило, события развиваются в два этапа:

1) делается попытка нашими тяжелыми (легкими) фигурами достигнуть превосходства сил (в первую очередь - по количеству ударов и угроз) на 7-ой или (и) 8-ой горизонталях или же в непосредственной близости от короля противника (на клетках пояса короля);
2) с помощью тяжелых или (и) легких фигур осуществляется нажим не позицию короля с тем, чтобы королю негде было "спрятаться" от их многочисленных ударов и угроз; в такой ситуации резко увеличивается возможность достижения нами если не главной цели шахматной игры - мата неприятельскому королю, то, во всяком случао, немалого - подавляющего позиционного и материального преимущества, часто – ñ ясными перспективами на решающую атаку позиции короля противника.

В О З М О Ж Н Ы Й __П О_Д_Х_О_Д__В Ф О Р М А Л И З А Ц И И

I) Для определенных тяжелых фигур проверяется факт, что они находятся г тылу противника и а) какая-либо из этих фигур имеет возможность безнаказанно взять пешку (или фигуру) противника из пояса короля или же просто стоящую на 7-ой или 8-ой горизонтали, либо:

á) эта фигура отнимает (отрезает) некоторые поля доступности короля противника;

в) другая тяжелая фигура шахует короля противника либо отнимает у него некоторые, отличные от вышеупомянутых, поля доступности.

2) Для легких фигур проверяется факт, что хотя бы две легкие Фигуры:

а) одновременно шахуют короля противника и отнимают у него некоторые поля доступности, либо

б) отнимают некоторые поля доступности у короля противника. Можно добавить также случай, когда слон связывает фигуру противника, стоящую в поясе короля или же когда конь (слон) имеет двойной удар (или же возможность такового) в отношении короля и другой неприятельской фигуры (более ценной или незащищенной, находящейся опять же из подступах позиции короля).

Следует учесть, что в реальных партиях атака позиции короля противника нашими фигураки осуществляется (в большинстве случаев) на фоне взаимодействия разных типов фигур.

Взлом

Под взломом подразумевается та разновидность атаки, когда на защиту какой-либо важной для своих целей клетки доски (множества клеток) противником переброшено немало сил и построено немало “защитных сооружени”, но в результате целенаправленных действий группы наших фигур:

1) защищающие или закрывающие эту клетку (множество клеток) фигуры противника:

а) прогоняются;

б) отвлекаются от объекта защиты;

в) отрезаем от объекта защиты;

г) обмениваются;

д) при возможности - уничтожаются;

2) таким же путем (а может и вследствие уже приложенных усилий) разрушаются “защитные сооружения” вокруг этой клетки (множества клеток) и, таким образом, цель становится достижимой.

Взлом позиции короля разменом пешки
(легких фигур)

Имеемся в виду ситуация, когда наша атака на неприятельского короля уже под готовлена (наши фигуры уже переброшены к месту действия) и наступает решающий момент. Если на подступах позиции короля стоит наша пешка, то мы предлагаем противнику размен пешек (на пешку из пояса короля) или же легкой фигуры (если таковые имеются с обеих сторон).

Ситуация должна быть такой, что:

1) противник должен (форсированно) пойти на предлагаемый нами размен (например, если предлагается размен пешек, то при отклонении его, следующим ходом берется пешка противника и, при соответствующей поддержка других наших фигур, нашей пешке остается один ход для успешного превращения или же дойтижения какого-либо выигрыша);

2) размен тоже не сулит ничего утешительного противнику, в том смысле, что по уже свободному полю пояса короля (или же по линии, кончавшейся на этом поле) открывается прямой, нередко роковой, доступ другим нашим фигурам к позиций короля.

Вçлом позиции короля жертвой пешек (легких Фигур)

Имеется в виду аналогичная описанной в предыдущем пункте ситуация, когда становится возможным проведение атаки (форсированного варианта), когда ценой одной или несволько пешек (легких фигур) в конце концов достигается нечто несравнимо большее, чем стоимость пожертвованного материала.

Часто это бывает мат неприятельскому королю, или же достижение подавляющего и решающего материального перевеса.

2. ЗАЩИТА

I. Защита позиции, если свои король в центре (рокировал на противоположный фланг; не рокировал).
2. Защита наступающих пешек с тыла тяжелыми фигурами.
3. Защита проходной пешки.
4. Защита активной пешки.
5. Защита слабой пешки.

Защита

Под защитой подразумеваются действия группы наших фигур, направленные на то, чтобы не допускать взятия (мат) противником определенной нашей пешки или фигуры (короля).
Частным случаем защиты является защита какой-либо фигурой другой фигуры того же цвета, что и первая.

Защита наступающих пешек с тыла тяжелыми фигурами

Фактор, имеющий решающее значение при атаках чего-либо важного
или ценного в лагере противника. Пешка, участвующая в таких действиях и надежно защищенная (в частности - с тыла тяжелыми фигурами, что наиболее удобная из защит пешки, имея ввиду факт, что пешка продвигается вперед только по вертикали) - грозная сила. Если же она не защищена или же недостаточно поддерживается нашими фигурами, то может стать легкой добычей фигур протвника, что одновременно грозит нам потерей темпа или же инициативы.

Защита проходной пешки

Если в позиции не ведутся какие-либо активные действия, направленные на достижение чего-либо важного или ценного и на доске присутствует наша проходная пешка, то, обычно, с её превращением связываются всякие надежды на успех. И, взависимоñти от того, защищена ли проходная пешка надежно или нет, контролируется ли её поле превращения (если до него уже рукой подать) нашими фигурами -- нам будет сопутствовать либо удача, либо какой-либо вид проигрыша.

Защита активной пешки

Когда пешка становится проходной или, в силу определенных обстоятельств, активной ударной силой, опасной, то первостепенной задачей бывает её защита и поддержка.

Защита слабой пешки

Имея в виду факт, что мишенями наших атак становятся главным образом и прежде всего слабости позиции противника, нужно своевременно (когда только-только начинается переброска наших сил в определенную область и намечаются какие-либо атакующие действия против неприятеля) защищать аналогичные слабости нашей позиции, возникших в силу тех или иных обстоятельств, в том числе и наши слабые пешки. Это делается прежде всего для того, чтобы не потерять темп и не дать инициативу в руки противнику.

3. Переоценка ценностей

1. Увеличение ценности коня.
2. Увеличение ценности пешки.
3. Конь находится на краю доски.

Увеличение ценности коня

В позициях, где фигурируют пешечные цепи противника одного цвета с нашим слоном, возрастает ценность нашего коня. Другими словами, пешечные цепи противника могут так заградить путь нашему слону, что из его автивности (подвижности, дальнобойности) не останется и следа и в такой ситуации наш конь может вполне заменить слона (взять на себя его функции).

Возможный подход к формализации

Будем считать, что ценность нашего коня увеличена, если:

I) Наш конь активен, т.е.:

а) на него нет опасных атак (скажем, комбинаций) противника;

б) наш конь подвижен (число заграждений и отрезанных полей не больше заранее задашюго числа N);

в) наш конь стоит на передних позициях (скажем, выше 5-ой горизонтали, в лагере противника либо в центре доски занимает активную, важную позицию);

2. На доске фигурирует наш слон;

3. Некоторое количество пешек противника (число которых не меньше заранее заданного числа *M*) стоят на клетках одного цвета с нашим слоном, заграждая ему путь или отрезая поля его доступности.

Переоценка ценностей

I. Увеличение ценности коня.
2. Увеличение ценности пешки.
3. Конь находится на краю доски.

Увеличение ценности коня

В позициях, где фигурируют пешечные цепи противника одного цвета с нашим слоном, возрастает ценность нашего коня. Другими словами, пешечные цепи противника могут так заградить путь нашему слону, что из его автивности (подвижности, дальнобойности) не останется и следа и в такой ситуации наш конь может вполне заменить слона (взять на себя его функции).

Возможный похход к формализации

Будем считать, что ценность нашего коня увеличена, если:

I) Наш конь активен, т.е.:

а) на него нет опасных атак (скажем, комбинаций) противника;

б) наш конь подвижен (число заграадении и отрезанных полей не больше заранее заданного числа ***N***);

в) наш конь стоит на передних позициях (скажем, выше 5-ой горизонтали, в лагере противника либо в центре доски занимает активную, важную позицию);

2. На доске фигурирует наш слон;

3. Некоторое количество пешек противника (число которых не меньше заранее заданного числа *M*) стоят на клетках одного цвета с нашим слоном, заграждая ему путь или отрезая поля его доступности.

Увеличение ценности пешки

Подразумеваются такие ситуации, когда:

1) пешка стоит на 7-ой горизонтали, а известно, что о её превращением в большинстве случаев положение противника отнюдь НÅ становится лучше;

2) пешка непосредственно участвует в главных событиях на доске (в решающей атаке на короля противника);

3) пешка непосредственно участвует в атаке на ключевые пункты (фигуры) в позиции противника;

4) пешка непосредственно участвует в атаке за овладевание какими-либо важными клетками (линиями) в позиции.

Конь находится на краю доски

Имеются ввиду часто встречающиеся в партиях ситуации, когда, в силу неправильных действий, конь оказывается вне всяких активных (атакующих или защитных) маневров, что влечет за собой неизбежное ухудшение позиции и потерю темпа.

4. КОНФИГУРАЦИЯ ГРУПП ФИГУР

1. Пешечная структура.
2. Конфигурации из фигур и пешек.
3. Конфигурации из фигур.

Пешечная структура

Хорошее, удобное расположение пешек в позиции на доске - один из важнейших факторов в большинстве случаев предопределяющий благоприятный исход партии.

С первых же дебютных ходов в партии нужно стремиться к целенаправленному и своевременному развитию своих пешек. Одако, в то же время, нужно всесторонне учитывать как можно больше, возникающих при продвижении пешек, сильных и слабых сторон отдельных пешек и пешечных конфигураций, таких как:

I) наличие пешечных фаланг, цепей и проход
н ы х п е ш е к, их расположение в позиции на доске (активность и пассивность, расстояние от краев доски и. т.д);

2)неудачное расположение пешек, при этом возможны:

а) временные слабости (возникающие, как правило, вследствие продвижения пешек или неудачных маневров);

б) постоянные или длительные слабости (изолированные, сдвоенные, блокированные, отсталые либо вообще расположенные неудачно пешки);

3) слабости геометрического характера.

Конфигурации из фигур и пешек

Часто в партиях случается, что большую роль в групповых дей
ствиях играют взаимосвязанные каким-либо образом фигуры и пешки одного цвета. Примером такой взаимосвязи могут служить конфигурации из коня и пешки:

а) фаланга из пешки и коня;

б) конь сзади от пешки;

в) конь сбоку от пешки.

Конфигурации из фигур

Конфигурации из фигур целесообразно рассматривать с точки зрения взаимодействия этих фигур (в силу большей, чем у пешек, подвижности фигур) на пути достижения какой-либо цели на доске. Примером такого взаимодействия может служить двойной (тройной) удар по определенной линии доски.

Возможный подход к формализации

1) наличие сдвоенных ладей на вертикали, при условии, что на вертикали нет пешек и других фигур либо есть некоторые другие фигуры;

2) наличие сдвоенных ладей на 7-ой или 8-ой горизонталях;

3) наличие на одной диагонали ферзя и слона, при условии, что на этой диагонали нет пешек (кроме, может быть, тех случаев, когда объектом нашей атаки служит пешка противника на этой линии) и у противника нет возможности поставить на эту диагональ свои пешки;

4) наличие в лагере противника наших коней, защищающих друг друга, которых противник не имеет возможность выиграть или удачно разменять;

5) наличие ф о р п о с т а (конь на подступах к лагерю противника и пешка надежно защищает его);

6) наличие сильных пешечных цепей;

7) наличие удачно расположенных, развитых фаланг пешек на доске.

5. СТАТИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПОЗИЦИИ

1. Мобилизованность фигур.
2. Подвижность фигур.
3. Скученность фигур.
4. Фигуры развиты.
5. Фигуры централизованы.
6. Фигуры сконцентрированы.
7. Взаимодействие фигур в позиц ии.
8. Фигуры проникли в тыл противника.
9. Активность фигур.
10. Атака слабых полей фигурами в позиции
11. Атака слабых пешек фигурами в позиции.
12.Заграждения.
13.Отрезаны полая.
14. Возможность удачных разменов.
15. Наличие форпостов.

Мобилизованность фигур

Под мобилизованностью фигур подразумевается степень готовности фигур (отдельных или некоторых их групп) к каким-либо конкретным действиям в некоторой области на доске.
Частным случаем сказанного является мобилизованность фигур (фигуры) в отношении какой-либо клетки доски.

Возможный подход к формализации

Фигуры мобилизованы в некоторой области доски, если:

I) каждая из этих фигур не может быть выиграна противником или удачно разменена на слабую (неразвитую, неудачно расположенную) фигуру противника;

2) да каждую ключевую (с точки зрения поставленной цели) клетку из данной области есть больше ударов нашими фигурами, чем фигурами (не менее ценных) противника.

Подвижность фигур

Для каждой фигуры множество доступных (реально) ей полей сравнивается с максимально возможным множеством полей доступности этой фигуры.

При этом нужно иметь ввиду, что;

1) может быть, следует отмечать ограничение подвижности данной фигуры фигурами того же цвета от ограничения подвижности фигурами противника;

2) в зависимости от типов позиций, а также от самой ситуации в позиции, должны быть внесены коррективы при определении подвижности той или иной фигуры (скажем, “подвижная” фигура у края доски или в отдалении от главных событий и “малоподвижная” фигура того же типа, немногочисленные атаки которой удачно вписываются в грушовые, целеустремленные действия фигур того же цвета).

ВОЗМОЖНЫЙ подход к формализации

Фигуру будем считать подвижной (в области), если она:

1) не связана в отношении более ценной или незащищенной фигуры того же цвета;

2) не находится под двойным ударом с другой фигурой того же цвета;

3) не блокирована противником;

4) не блокирует путь какой-либо фигуре противника;

5) не защищает что-либо ценное или малозащищенное;

6) не стоит на своем исходном месте, когда на доске есть еще много фигур;

7) не стоит у краев доски;

8)число заграждений и отрезанных полей на пути её продвижения меньше некоторого заранее заданного числа (которое, в общем случае, зависит от типа позиции, этапа развития партии, а также от самой ситуации на доске).

Скученность фигур

Имеется ввиду возникающие в результате неправильных действий одной из сторон или принуждающих действий противника ситуации, когда на небольшом сравнительно пространстве доски несколько фигур этой стороны резко ограничивают друг другу подвижностьсть (заграждают путь друг другу). При этом ВОÇÌÎÆÍÛ следующие случаи:

1) группа фигур этой стороны на данном множестве клеток (часто - на нескольких соседних клетках) не объединена какой-либо общей целью;

2) цель есть, но в группе есть фигуры, пребывание которых в области совсем не обязательно (может быть, даже мешает выполнению плана, направленного на достижение этой цели), что часто приводит к серьезным последствиям, вплоть до проигрыша партии.

Опасность скученности фигур на небольшом пространстве часто выражается в том, что становятся возможными связки и двойные удары в отношении некоторых фигур из этой группы, что в свою очередь приводит к материальным потерям.

ФИГУРЫ развиты

I)В связи с поставленной первоначальной д е б ю т н о й ц е л ь ю и в зависимости от конкретного типа дебюта завершена м о б и л и з а ц и я своих фигур в предполагаемые области доски.

Понятие тесно связано с д е б ю т н о й ц е н т р а л и з а ц и е й фигур и пешек.

В результате удачно сыгранного дебюта фигуры одной из сторон можно считать р а з в и т ы м и, если:

а) созданы передние (центральные, ударные) пешечные конфигурации (фаланги, цепи);

б) ударная сила наготове (слоны и кони занимают клетки, предлагаемые теорией, тяжелые фигуры не занимают исходные места (не пассивны) и готовы к переброске к местам будущих “сражений”);

в) произведена рокировка (в большинстве случаев) и король находятся под надежным прикрытием своих пешек и фигур.

2) В общем случае:

а) фигура не стоит на своем первоначальном месте;

б) централизована;

в) находится в лагере противника и не может быть им взята или удачно разменåна;

г) участвует в групповых действиях своих фигур, направлен-ных на достижение важном с какой-либо точки зрения цели (ата-ка на позицию противника или защиту собственной);

д) п е ш к а -не отсталая; ш

е) держит под ударом хотя бы одну важную (с какой-либо точки зрения) клетку доски;

ж) защищает фигуру того же цвета;

з) атакует хотя бы одну фигуру противника ценнее её самой;

и) если она - проходная, то находится вблизи от поля своего превращения.

Фигуры централизованы

I.В соответствии с конкретным типом д е б ю т а и поставленной целью одной из сторон им завершено создание фигурного,фигурно-пешечного или же пешечного центра (могут быть различные ва-
рианты расположения фигур и пешек в центре доски и на его подступах).

Успешное построение крепкого центра в дебюте обуславливает в большинстве случаев возможность дальнейших удачных действий против позиции неприятеля.

<u>Возможный подход к формализации</u>

а) Конь (кони) атакует центр доски или же сам расположен там;

б) пешка (пешки) атакует центр доски или же сама расположена там;

в) слон (слоны) атакует центр доски или же находится на его подступах;

г) ферзь не стоит на исходном месте, не атакован противником и защищает передние свои силы.

Причем, в отношении всех указанных фигур у противника нет возможности выиграть их или удачно разменять на свои слабые, пассивные фигуры.

2) В с е р е д и н е и г р ы, а также в э н д ш п и л е также большую роль играет централизованность фигур и пешек. Из центра доски становится возможным быстрейшая переброска имеющихся там фигур в области, где должны быть проведены какие-либо атакующие действия.

а) в э н д ш п и л е иногда король, достигший центра ДОСКЀ или находящийся на его подступах - грозная сила, нередко решающий исход партии в свою пользу;

б) для легких фигур предпочтительнее, если они находятся в середине доски, так как в таких ситуациях увеличивается их атакующая мощь и, во-вторых, в э н д ш п и л ь н ы х позициях они служат надежным прикрытием для своего короля, расположенного в центре доски;

в)в э н д ш п и л е наиболее удачным считается расположение ферзя в центре (кроме, разумеется, тех случаев, когда решающая атака ведется у края доски) с тем, чтобы в нужный момент он мог прийти на подмогу фигурам своего цвета.

ФИГУРЫ сконцетрированы

В связи с поставленной задачей (целью) завершена подготовка к активным действиям, направленных на конкретную реализацию этой задачи на доске. Другими словами, нужное количество своих фигур(соответствующих типов), которые будут участвовать в достижении данной цели, уже переброшено к "месту действий" (в нужную область доски). Предполагается, что последующие ходы этими фигурами уже будут активными, т.е. начнется сама атака на достижение поставленной цели.

Возможный подход к формализации

В некоторой области доски:

1) у противника имеются какие-либо слабости (слабые фигуры или клетки) либо определенная группа его фигур расположена неудачно (воçможны связки, двойной удар и т.д.);

2) у нас есть материальный перевес (пешечный или в фигуру);

3) у группы наших фигур есть позиционный перевес (т.е. они расположены в этой области удачнее, чем соответствующие фигуры противника);

4) наши фигуры в этой области но стоят скученно;

5) в отношении наших фигур в этой области у противника нет возможности выиграть их или удачно разменять на свои слабые и пассивные фигуры.

Взаимодействие фигур в позиции

На позиции выявляются группы фигур, участвующих в каких-либо действиях, направленных на достижение определенной цели на доске:

I) в маневрах или переброске из одной области доски в другую;

2) в атаке на определенную фигуру противника или на некоторые клетки в его лагере;

3) в защите чего-либо ценного или малозащищенного в собственном лагере.

Возможный подход к формализации

Фигуры взаимодествуют, если:

1) они мобилизованы с определенной целью в некоторой области;

2) их удары сконцентрированы на слабой фигуре противника или на важную с какой-либо точки зрения клетку в позиции противника;

3) они мобилизованы в центре (централизованы) с целью дальнейшего развития своих фигур и усиления давления на позицию противника;

4) они участвуют в каких-либо проводящихся на доске атакующих действиях (атака слабых пешек и фигур, атака позиции короля и т.п.);

5) они участвуют в каких-бо защитных операциях, направленных на пресечение попыток противника атаковать слабости нашей позиции и занятия там ключевых клеток.

Фигуры проникли в тыл противника

Когда тяжелые фигуры, поддерживаемые своими легкими фигурами (иногда -пешками) безнаказанно вторгаются на 7-ую таи 8-ую горизонтали, причиняя немало неприятностей противнику (часто-непоправимых), то говорят, что эти фигуры проникли в тыл противнику.

Возможный подход к формализации

Фигуры (из тяжейых)находятся на 7-ой или 8-ой горизонталях, где находится и король противника.Противник их не атакует либо не имеет возможности выиграть их или разменять на свои равноценные фигуры,и с их стороны возможны:

1) разрушение с тыла противника его пешечных конфигураций с целью открыть доступ в тыл противника, в первую очередь, своим пешкам, а также другим фигурам, тем самым сжимая кольцо вокруг короля, противника;

2) прямое нападение на короля противника с последующим усилением своих позиций;

3) материальный выигрыш какой-либо фигуры противника (может быть - комбинация с последующим матом королю).

Активность фигур

Фигуру назовем а к т и в н о й, если она:
1) развита;
2) участвует в атаке на короля противника или содействует в ней своим фигурам;
3) связывает какую-либо фигуру (пешку) противника, укрывающую что-либо ценное, важное или малозащищенное;
4) у фигуры есть двойной удар в отношении некоторой пары фигур противника либо возможность такого удара;
5) является одной из проникших в тыл противника либо содействующих такому проникновению;
6) владеет свободной линией.

Атака слабых полей фигурами в позиции
и
Атака слабых пешек фигурами в позиции

Слабости в лагере противника (слабые поля, пешки и фигуры) только тогда влияют на течение партии и её исход, когда они атакуются в соответствующей ситуации группой фигур. Имея ввиду сказанное, становится необходимым выяснить: какие “потенциальные” слабости фигурируют в позиции противника и в какой степени найденные клетки в позици противника контролируются своими фигурами (с целью создания в дальнейшем на этих полях б а з и ф о р п о с т о в для атаки либо выигрыша неудачно расположенных на этих клетках пешек противника).

Заграждения

З а г р а ж д е н и е м для некоторой фигуры противника служит наша менее ценная фигура (или пешка), которая загораживает хотя бы одну клетку доступности этой фигуры противника, и которую противнику нецелесообразно брать (грозит потеря темпа, позиционный или материальный проигрыш). Таким образом в худщем случае такая наша фигура загораживает одну клетку (на которой находится она сама) доступности фигуры противника, а если последняя ходит по линиям, то, кроме того, - весь отрезок линии, находящийся за этой клеткой.

Целями построения заграждений являются:

1) помешать фигуре противника перемещаться в какую-либо область доски (дая защиты некоторой фигуры своего цвета, для нанесения удара по нашей позиции и т.п.);

2) поймать эту фигуру в ловушку с целью последующего её выигрыша;

3) вынудить эту фигуру ходить в неудобное дая противника место

с целью отрезать её от главных событий на доске.

Следует отметить, что заграждением на пути некоторой фигуры может служить также неудачно расположенная фигура того же цвета.

Отрезаны поля

В отношении какой-либо фигуры противника проверяется наличие таких полей её доступности, на которые она реально не может пойти в силу того,что:

1) если на эти поля нет ударов противника,то каждое из них хотя бы один раз атаковано нашими фигурами; либо

2) на эти поля есть хотя бы один удар менее ценной (в отношении исследуемой фигуры противника) нашей фигурой.

При исследовании отрезанных полей преследуется аналогичные описанным в предыдущем параграфе(см"Заграждения") цели.

Возможность удачных разменов

Под возможностью удачного размена подразумевается ситуация, когда некоторая наша фигура атакует однотипную фигуру противника, причем:

1) наша фигура слаба (отсталая, неактивная, малоподвижная и т.п.)
2) фигура противника, напротив, более активна, развита, непассивна, опасна.

Выяснение вопроса - сколько наших фигур можно разменять на равноценные им фигуры противника, имеет большое значение, так как осуществление таких разменов, как правило, заметно усиливает нашу позицию.

Наличие форпостов

Форпостом называется собственная фигура в неприятельском лагере, находящаяся на открытой линии и защищенная пешкой. Форпост служит базой для проведения новых атак и провоцирует ослабление сопротивляемости противника на открытой линии.

Отсюда видно, сколь важна роль форпостов в партии. Наличие в позиции форпостов одной из сторон означает, что у нее есть неплохие шансы провести какие-либо атакующие действия, направленные на подрыв позиции противника (проникновение в его тыл с вытекающими из этого самыми серьезными последствиями).

6. Тактические действия групп фигур

1. Борьба за центр.
2, Концентрация сил в области.
3. Концентрация сил против важного пункта.
4. Мобилизация фигур в область,
5. Проникновение в тыл противника.
6. Провоцирование на слабый ход.
7. Осложнение позиции.
8. Отвлечение фигуры из области.
9. Завлечение фигуры в область.
10. Построение заграждений.
II. Отрезание полей.
12. Применение разносторонних угроз.
13. Запирание линии.

14. Захват линии.
15. Овладение линиями.
16. Овладение важными клетками (полями).
17. Овладение полем превращения проходной пешки.
18. Овладение критическими полями сильной (слабой) пошкѐ.
19. Использование полей соответствия.
20. Создание “форточки”.
21. Вскрытие.
22. Вскрытие линии посредством размена (жертвы).
23. Размен.
24. Использование удачных разменов.
25. Блокада.
26. Взаимодействие фигур.
27. Комбинация.
28. Достижение вечного шаха.
29. Создание форпостов.

Борьба за цели

I) Является одной из главных д е б ю т н ы х целей.

В соответствии с поставленной задачей - построения прочного центра како го-либо типа и создания мощной базы для последующих атакующих действий, а также в силу сложившихся на доске обстоятельств (контрвыпады, варианты противника),свои фигуры перебрасываются в центр доски или к его подступам с максимально возможной пользой для себя и, одновременно, подготавливая пространство и ресурсы для дальнейших решительных действий, направленных на штурм позиции неприятеля.

2) В середине партии, а иногда и в эндшпиле, успешные действия, проводимые с целью достижения каких-либо выгод на доске, просто не мыслимы, не отделимы от одновременной борьбы за центр и усиления там уже завоеванных позиций.

Успешно строить центр, гармонично развивая свои фигуры и пешки, увеличивая с каждым ходом давление на позицию противника, одновременно накапливая позиционные преимущества - это уже твердая основа для проведения успешной атаки неприятельского лагеря (часто - решающей исход партии).

Концентрация сил в области

В связи с поставленной целью (коллективные, групповые действия фигур, направленные на атаку или защиту чего-либо) осуществляется переброска соответствующего контингента сил в область доски, где эти действия предполагается произвести.

Концентрация сил против вазшого пункта

То же самое, что и в предыдущем пункте, но в этом случае усилия направлены конкретно на о в л а д е н и е некоторым в а ж н ы м п у н к т о м в позиции противника.

Следует отметить следующее:

1) нелепо сосредоточивать силы против п о д в и ж н о й фигуры, которая Â любой момент может уйти;

2) можно концентрировать силы против с в я з а н н о й фигуры, совершенно неподвижной, либо против защищающей фигуры, подвижность которой ограничена;

3) концентрация сил иногда производится не для овладения пунктом, а для о в л а д е н и я л и н и е й.

Мобилизация фигур в область

После того, как просматриваются предпосылки (мотивы) для проведения каких-либо атакующих действий или же возникает необходимость защиты чего-либо ценного или незащищенного и становится ясным план (хотя бы приблизительный) этих действий, нужное количество фигур соответствующих типов перебрасывается (максимально удобно и быстро) к месту предполагаемых событий.

Понятие по сущности очень близко к понятию “Концентрация сил в области”.

Проникновение в тыл противника

Групповые атакующие действия (вторжение за передние оборонительные линии, заñлоны и пункты противника) с обязательным участием в них своих тяжелых Фигур, направленные на подрыв изнутри лагеря противника.

Осуществляются приблизительно в таком порядке:

1) о с л а б л е н и е какой-либо части (на определенном фланге или в центре) передних защитных сооружений противника посредством концентрического удара своих фигур, их принуждающих действий;

2) ф о р м и р о в а н и е в возникших “провалах” (имеются ввиду плохо защищенные либо незащищенные клетки) переднего края позиции противника о т к р ы т ы х л и н и й и ф о р п о с т о в;

3) в т о р ж е н и е на 7-ую (8-ую) горизонталь при поддержке своих легких фигур и, может быть, пешек (затрудняющих защитные действия фигур противника и держащих в напряжении “окружение” его короля).

При этом преследуются такие цели, как:

а) с т е с н е н и е п о з и ц и й к о р о л я п р о т и в н и к а;

б) з а с т а в и т ь к о р о л я з а н я т ь н е у д о б н ы е д л я с е б я к л е т к и (например, принуждение его к преждевременному, “самоубийственному” выходу из-за пешечного прикрытия и т.п.);

в) в з я т и е ценных фигур противника (нанося двойные удары, связывая их и т.п.)

г) в з я т и е плохо защищенных иди незащищенных фигур или пешек в лàгере противника;

д) в ы г о д н ы й р а з м е н(с получением после него более перспективной позиции, не отдавая при этом инициативу противнику).

Провоцирование на слабый ход

1) После нашего хода противник вынужден ф о р с и р о в а н н о делать неудобный для себя, неудачный ответный ход;

2) то же самое он делает после некоторых форсированных ходов, лосле того, как мы ограничиваем его действия Â рамках какого-либо варианта.

При этой с нашей стороны могут быть применены:

а) поля соответствия;

б) достижение положения цугцванга с правом хода в нем противника;

в) увеличение потенциальных угроз в различных областях доски (применение разносторонних и многочисленных угроз);

г) з а в л е ч е н и е фигур противника в ловушку или неудобное для них место;

д) о т в л е ч е н и е фигур противника из области главных событий (от объекта защиты, от ключевых постов и т.п.).

Осложнение позиции

В позициях, где не видны пути дальнейшего развития действий, вñе хорошо и надежно защищено или, другими словами:

1) отсутствуют полезные рекомендации;
2) затруднена (неочевидна) оценка позиции;
3)позиция носит закрытый характер, делаются выжидательные, подготовительные ходы, не ухудшающие нашу позицию, одновременно нагнетая обстановку по всей доске.

Делается это посредством усиления уже имеющихся позиций на доске и создания новых угроз и ударов в различных областях доски (применение разносторонних угроç, концентрация сил, централизация и т.п.). В результате вышеуказанных действий анализ возникающих вновъ позиций от хода к ходу все осложняется, а при удачном их проведении (не далая беñполеçных ходов и накапливая позиционные преимущества), начиная с некоторого момента, у противника могут проявляться кое-какие слабости, которые в дальнейшем будут стремиться использовать (уже целеустремленно) наши фигуры.

Отвлечение фигуры из области

Действия (манåвры) нашими фигурами, иногда ииеющие некоторые оттенки к о м б и н а ц и и, в результате которых в конце концов противник в ы н у ж д е н н о (т. е. форсированно) делает неуход некоторой своей фигурой (что и было задумано нами), вследствие чего эта фигура отрезается от области главных событий. Тем самым иногда открывается путь:

1)к улучшению нашей позиции (с дальнейшими, лучшими, чем у противника, перспективами);
2) к в ы и г р ы ш у н а м и т е м п а и у с и л е н и ю нашей и н и ц и а т и в ы ;
3) к проникновению в тыл противника;
4) к выигрышу ставшей уже незащищенной фигуры противника или же плохо защищенной фигуры противника;
5) к проведению комбинации (нередко-решающей);
6) к успешному превращению нашей проходной пешки;
7) к овладению о т к р ы т о й л и н и е й;
8) к запиранию л и н и и.

Завлечение фигуры в область

То же самое, что и в предыдущем пункте ("Отвлечение фигуры из областй"), с той лишь разницей, что в результате неудачного хода некоторой фигурой противника, эта фигура перемещается в невыгодное для себя место, при этом нередко становится возможным:

1) у л у ч ш и т ь н а ш у п о з и ц и ю (с дальнейшими, лучшими, чем у противника,перспективами);
2) в ы и г р а т ь т е м п и р а з в и в а т ь и н и ц и а т и в у;
3) выиграть завлеченную фигуру;
4) выиграть некоторую другую фигуру противника, связи между которой и завлеченной уже не существует;
5) объявить мат завлеченному королю;
6) на основе возникших в позиции слабостей противника начать к о м б и н а ц и ю на приобретение задуманного;
7) нанести д в о й н о й у д а р;
8) о т к р ы т ь или з а п е р е т ь л и н и ю;

9) ограничить подвижность неприятельских фигур.

Построение заграждений

Сущность построения заграждений заключается в том, чтобы ограничить подвижность мобильных, активных фигур противника, а также его короля путем занятия клеток их доступности, а также, тем самым, закрывая линии их следования какими-либо нашими фигурами, которые:

1) менее ценны в сравнении с заграждаемой фигурой противника;

2) не “съедобны” (т.е. у противника нет возможности брать их или же удачно разменять ав свои однотипные фигуры).

Целями, преследуемыми при построении заграадений в общем случае являются:

а) в ы к л ю ч е н и е из игры а к т и в н о й, р а з в и т о й ф и г у р ы противника (например, активного слона противника - нашими пешками);

б) о г р а н и ч е н и е боеспособности а к т и в н о й фигуры противника;

в) еще больше ограничить подвижность фигуры противника, которая малоподвижна или неудачно расположена, с целью “захлопнуть” окончательно ловушку и выиграть ее.

Отрезание полей

Понятие, близкое по содержанию “Построению заграждений” (см. предыдущий пункт).

Имеется ввиду ограничение подвижности фигур противника, а также его короля, путем нанесения ударов на поля доступности этих фигур (в результате чего атакованные поля “отрезаются” для этих

фигур противника) некоторыми нашими фигурами, которые:

1) менее ценны в сравнении с фигурой противника, поля доступности которой “отрезаются”;

2) просто не допускают хода фигурой противника на исследуемое поле ее доступности (грозя ее взятием);

3) не могут быть взяты противником или удачно разменены им на свои неудачно расположенные фигуры той же стоимости.

Целями отрезания полей доступности фигур противника являются те же, что и при построении заграждений на пути этих фигур.

Применение разносторонних ÓÃĐÎÇ

Имеется ввиду факт, что план какого-нибудь атакующего маневра иногда строится на сочетании различных угроз с тем, чтобы, если противник и спасается от одной угрозы, он станет жертвой другой.

Запирание линий

Важную роль в ходе партии, в первую очередь при защите ÑÂÎÈÕ позиций, играют тактические действия фигур одного цвета, направленные на запирание линий с целью ограничения подвижности тяжелых фигур (редко - слона) противника по этим линиям.

Линия на доске (в большинстве случаев под этим понимается в е р т и к а л ь) считается закрытой (запертой) для определенной фигуры (фигур) противника, если на ней стоит:

1) пешка (наша, которую противник не может взять, или противника) -длительно запертая линия;

2) фигура противника, подвижность которой в той или иной степени ограничена нами (скажем, она связана нашими фигурами в отношении какой-либо другой фигуры противника);
3) наша фигура (менее ценная в сравнении с исследуемой фигурой противника), которую противник не может взять или же удачно разменять на свои равноценные фигуры.

В общем случае запирание линий может быть применено как в атакующих, так и в защитных действиях фигур. В первом случае преследуется цель обезопасить свой тыл от возможных контрдействий (контратак) вражеских фигур в ответ на наши атакующие маневры. Во втором же случае линии запираются за тем, чтобы ослабить нажим фигур противника на наши позиции и вместе с тем подготовить, сконцентрировать свои силы для дальнейших активных действий - перехода от защиты к атакующим действиям (контратаке).

Захват линии

В соответствии с поставленными целями (в большинстве случаев-при атаке позиций противника) предпринимаются целенаправленные действия, чтобы, во-первых, открыть соответствующую линию не доске (если она з а к р ы т а для некоторой нашей фигуры) и, во-вторых, захватить эту линию некоторой нашей фигурой (фигурами). При этом возможны следующие ситуации:

1) противник не владеет линией, но и у нас нет пока возможности поставить на нее ту фигуру, которую хотелось бы, с одновременным захватом этой линии; таким образом, в этом случае линия сначала должна быть освобождена от мешающих ее захвату наших фигур и фигур противника, после чего, когда все уже подготовлено - захвачена нами;

2) линяя стала о т к р ы т о й в силу определенных разменов (может быть, спровоцированными нами) и ведется борьба за ее захват не только нами, но и противником. В этом случае линию захватит тот, у которого все было задумано основательно и за кем была инициатива;

3) противник уже владеет линией, но захват ее нам крайне необходим для наших целей (атакующих): борьба производится с таким расчетом, чтобы, во-первых, удалить с линии и ее подступов захватившие линию фигуры противника, ограничить их подвижность и, во-вторых, перебросить к линии столько наших фигур, сколько хватило бы для з а п и р а н и я ее для фигур противника, а потом и ее захвата нашими фигурами.

Иç оказанного следует, что действия наших фигур, направленные на захват линии, должны носить ф о р с и р у ю щ и й характер, т.е. представлять из себя какие-либо у г р о з ы или являться комбинациями. Ясно также, что приемами при этом, в частности, могут быть: о т в л е ч е н и е фигур, завлечение фигур, д в о й н ы е у д а р ы, с в я з к и и т.д.

Овладение линиями

Аналогичное описанному в предыдущем пункте (“Захват линии”) понятие.

Овладение важными клетками (полями)

Ставится задача взять под свой контроль либо занять какую-либо важную с некоторой точки зрения клетку (поле) на доске (например, поле превращения нашей проходной пешки, близкой к превращению - с целью обеспечить удачное завершение ее опасного для противника и нередко победного “марша”).

При этом могут быть применены следующие приемы:

1) м о б и л и з а ц и я наших фигур в нужную область доски;
2) к о н ц е н т р а ц и я наших сил вокруг (на подступах) этой клетки (поля);
3) о т в л е ч е н и е некоторых фигур противника в сторону от главных событий;
4) з а в л е ч е н и е некоторых фигур противника в неудобные для себя места.

Овладение критическими полями сильной
(слабой) пешки

Известно, что каждый тип пешек (проходной, изолированной и т.д.) имеет на диске свои “достоинства”” и слабости. Одной из слабостей расположения конкретной пешки является характеризующее ее г е о м е т р и ч е с к и на доске понятие –к р и т и ч е с к и е п о л я этой пешки. Если вспомнить, что овладение этими полями нередко приводит к ее уничтожению (если пешка противника) или к неприступности ее позиции (если пешка-наша), то становится ясным смысл предпринимаемых в этом направлении действий.

Использование полей соответствия

Нами преследуется цель повести партию по такому руслу, чтобы получилась позиция, где ход - противника и у него исчерпаны все осмысленные, удачные ходы (положение так называемого ц у г ц в а н г а). При этом в начале этих действий исходной парой полей соответствия являются клетки местонахождения королей. В начальной позиции (где мы начинаем наши действия, направленные на использование полей соответствия), как правило, на два или три наших хода у противника существуют форсированные ответы. Но после некоторого нашего хода (обычно - третьего или четвертого) у противника не находится вразумительного ответа и он нарушает п р и н ц и п с о о т в е т с т в и я (т.е. отвечает на наш, отличный от прежних ход повторением одним из своих прежних ходов), что в таких позициях приводит к невосстанавливаемым потерям (часто - к форсированному мату).

Создание “форточки”

В зависимости от хода партии и наших текущих задач в ней, часто бывает необходимо учитывать факт, что определенная тяжелая фигур(фигуры) противника может ири благоприятствующих этому обстоятельствах, проникнуть в наш тыл и зоматовать нашего короля. В силу этого, иногда бывает целесообразным продвинуть на одну клетку одну из пешек (часто - крайнюю правую либо крайнюю левую, смотря по обстоятельствам) перед нашим рокировавшим направо или налево королем. При этом стараемся, чтобы открытая таким образом клетка (“форточка” для отступления королем) не находилась по диагоналям под ударом каких-либо фигур противника,

С нашей же стороны в отношении подобных действий противника проводятся маневры с целью не допускать создания “форточки” для его короля, а если же она создана, то держать ее под ударом наших фигур (при условии, что наши другие фигуры атакуют с тыла позиции короля противника).

В случае же если первая (вторая) горизонталь нами хорошо защищена (наши ладьи надежно прикрывают короля), необходимость создания “форточки” для ухода короля не возникает.

Вскрытие

Имеются ввиду действия наших Фигур, когда на пути реализации какой-либо цели на доске открываются определенные линии либо открывается доступ в определенную область доски. В эти действия входят также, разумеется, такие маневры нашими фигурами, которые принуждают противника “содействовать” задуманному нами. Таким образом, эти действия в большинстве случаев должны носить ф о р с и р о в а н н ы й характер.

Вскрытие линии посредством размена
(жертвы)

В случаях, когда для осуществления каких-либо целей нам необходимо вскрыть определенную линию (вертикаль или диагональ), предпринимаются действия, направленные на то, чтобы;

1) отвесги в сторону наши фигуры, стоящие на этой линии либо неудобно расположенные (с точки зрения вскрытия линии);

2) вынудить уйти со своих мест те фигуры противника, которые или стоят на линии и на ее подступах, или же просто мешают вскрытию линии нами.

Отсюда вытекает, что вследствие маневров нашими фигурами противник должен ф о р с и р о в а н н о покидать удобные для своих фигур места. *А* это, как правило, достигается нами применением удачных разменов и жертвами, после которых инициатива полностью оказывается у нас.

В результате же удачно проведенных действий на вскрытие линии возникает благоприятствующая нашей основной цели ситуация, когда:

а) на линии нет собственных: фигур и пешек, кроме тех случаев, когда таковые имеются, но только-сзади той нашей фигуры, для прорыва которой и линия вскрыта;

б) на линии нет защищенных пешек или фигур противника (т.е. пешек или фигуры, брать или разменять которые нельзя).

Размен

Р а з м е н — обменивание равноценных фигур соперников, - одно
из общеизвестных шахматных понятий.

Каким образом и где происходит размен:

а) угроза размена может быть использована для того, чтобы заставить противника отказаться от той или иной сильной позиции фигуры (он предпочтет отступить ею, нежели разменяться, что только ухудшило бы его позицию);

б) если двое стремятся к одному и тому же, то неизбежен конфликт; в шахматах такой конфликт (скажем, в борьбе за поле) выражается в форме размена;

в) слабые пункты (т.е. слабые фигуры), равно как и слабые пешки, имеют тенденцию взаимно разманиваться.

Возможность удачного размена

Имеется ввиду ситуация, когда некоторая ваша фигура (пешка) атакует равноценную фигуру противника и:

1) наша фигура с л а б а (отсталая, пассивная и т.п.);

2) фигура противника а к т и в н а или о п а с н а;

3) разменом открывается доступ в нужную нам область доски.

Предлагая противнику такого рода размены (что нередко делается в таких ситуациях, когда у противника есть ограниченное число ответов, т.е. размен форсируется нами), мы преследуем цели:

а) УНИЧТОЖИТЬ активную, опасную фигуру или пешку противника;

б) если же противник не разменивается, то, обычно, его фигура отступает в неудобное для себя место, а наша фигура, кроме того, что включилась в активную борьбу (ведь она занимала далеко не лучшее для себя место), может еще и стать одной из наших активных фигур в дальнейших наших действиях.

Блокада

Б л о к а д о й называются групповые действия нескольких фигур одного цвета, имеющие цель заградить, выключить из игры, ограничить мобильность пешек или фигур противника либо не допустить их проникновения в какую-либо область доски, а потом, при возможноñти, захватить их.

Блокада против подвижной, мобильной фигуры нецелесообразна.

Взаимодействие фигур

Так назовем вытекающие из какого-либо плана действия группы одноцветных фигур, направленные на достижение определенной цели на доске.

На основе этого понятия лежит общеизвестная шахматная истина: насколько важнее, труднее поставленная цель, настолько и сильнее должно быть взаимодействие фигур, т.е. согласование действий фигур на пути достижения этой цели.

Комбинация

Это форсированный маневр, связанный с жертвой фигуры.

В ситуации на доске, когда для проведения комбинации есть веские основы, подразумевается выполнение таких действий (своими фигурами), что:

1) вначале жертвуется фигура (или пешка);

2) в дальнейшем возможности противника ответить нам активными ходами сильно убывают, и, начиная с некоторого момента, сделанные им ходы никакой пользы ему не приносят (может быть, напротив, ухудшают его положение);

3) в конце концов мы не только восстанавливаем материальный баланс, но и достигаем такого позиционного или материального (часто-подавлящего)преимущества, которого не было до комбинации и к достижению которого мы стремились.

7. Возможности фигур

1. Пространство для маневра.
2. Разносторонние угрозы
3. Захваченность множества полей Х фигурами У.
4. Контролирование множества полей Х фигурами У.
5. Ресурсы обороны.

Пространство для маневра

Понятие тесно связано с п о д в и ж н о с т ь ю фигур. Для каждой фигуры выявляются те клетки (поля) доски, по которым эта фигура может свободно (т.е. без какой-либо опасности для себя) маневрировать.

К понятию можно подойти еще и с другой стороны: рассматриваем пространство для маневров некоторой группой наших фигур в пределах определенной области доски (может быть, атакуемой этой группой наших фигур).

Разноосторонние УГРОЗЫ

Очень важно для всех фигур каждой стороны по отдельности выяснить, в каких событиях на доске они участвуют и возможно ли их подключение к новым, более целеустремленным и многообещающим групповым действиям этой стороны (скажем, в комбинации).

Захваченность множества полей Х
фигурами У

Выясняется факт, насколько удачно расположена группа некоторых наших фигур (фигура У) в определенной области (областьХ) доски и в ее окрестностях. Лучшим случаем считается то,

когда эта область (множество полей) захвачено нашими фигурами. Под этим мы понимаем ситуацию, когда каждая из фигур У стоит на какой-либо клетке облвсти Х или на ее подступах и, кроме того, что она выполняет свою роль в важных для нас с некоторой точки зрения действиях
(в рàмках которых и используется область Х), эту фигуру:

1) противник не атакует, а иногда и не может атаковать в течение некоторого времени (т.е. возможно в той иной степени длительное пребывание фигуры на своем месте); либо

2) противник не может ваять (т.е. взятие ее привело бы к материальному либо позиционному проигрышу).

В отношении же других клеток области (на которых нет наших фигур) возможен следующий подход.

Если клетка области Х является п о л е м (на ней нет фигур), то:

3) фигуры противника (может быть, определенного типа) не могут ходить туда (поле о т р е з а н о, з а г р а ж д е н о либо ход туда грозит противнику м а т е р и а л ь н ы м п р о и г р ы ш е м);

4) может быть, на это поле может переместиться некоторая наша фигура (без какой-либо опасности для себя), пребывание которой там желательно.

Если же клетка области Х является пунктом противника, то:

5) мы можем удачно разменять на фигуру противника, находящуюся на этой клетке, нашу слабую, пассивную фигуру той же ценности;

6) у нас есть возможность прогнать эту фигуру противника (о т в л е ч е н и е, завлечение или просто у г р о з а в з я т и я), после чего она не будет мешать осуществлению нашего замысла.

<u>Контролирование множества полей Х
фигурами У</u>

Понятие тесно связано с описанным в предыдущем пункте (см." Захваченность множества полей Х фигурами У") и может быть рассмотрено как слабый вариант его.

<u>Ресурсы обороны</u>

Успешность предпринимаемых нами атак во многом зависит от того, правильно ли организована у нас защита, во-первых, нашего короля, во-вторых, слабостей в нашем лагере и, в третьих, наших, участвующих в каких-либо атакующих действиях, фигур. Другими словами (и более подробно), должно быть проверено, д о с т а т о ч н о ли защищено то, что в а ж н о, с л а б о или же просто должно быть н а д е ж н о з а щ и щ е н о, а именно:

1) наш король с прикрывающими к нему доступ нашими фигурами и пешками;

2) слабые или неудачно расположенные пешки (изолированные, отсталые сдвоенные, блокированные);

3) активные развитые пешки (прежде всего - проходные);

4) слабые, отсталые или неудачно расположенные наши фигуры, которые могут стать мишенью атак противника;

5) участвующие в каких-либо атакующих действиях либо уже проникшее в тыл противника наши фигуры;

6) слабые поля в нашем лагере (дырки, критические поля каких-либо наших пешек);

7) важные для нас с какой-либо точки зрения поля (скажем, поле превращения проходной пешки);

8) важные для нас линии и подступы к ним (скажем, линии, оканчивающиеся на нашем короле).

<u>8. Характеристики позиции короля</u>

1. Рокировка потеряна.
2. Защита рокировавшего короля.
3. Защита нерокировавшего короля.
4. Король находится в безопасности.
5. Позиция короля стеснена.
6. Отсутствие “форточки”.

Одним из важнейших факторов проведения успешных атак на позицию противника является надежно защищенный, недоступный противнику тыл и, в первую очередь, хорошо прикрытые своими фигурами позиции короля. Король находится в опасности - и не может быть речи об атаках, тем более - успешных.

Рокировка потеряна

Рокировке короля (длинной или короткой) в шахматах придается большое значение. Это главным образом связано с тем, что на каком-либо фланге, после рокировки, организовать надежную защиту короля оказывается более целесообразным и удобным.

Рокировка потеряна (неосуществима), если;

1) уже был ход королем или ладьей (того фланга, куда собирался рокировать король);
2) король находится под шахом;
3) после рокировки король оказывается под шахом или же ладья -под ударом;
4) какое-либо поле между королем и ладьей атаковано противником.

С первых же дебютных ходов преслдуется цель не только гармонично и оптимально развивать свои фигуры, но и подготовить и провести рокировку собственного короля в нужную сторону. При этом, разумеется, соблюдается некоторая осторожность, чтобы фигуры противника не помешали рокировке, так как защита нерокировавшего короля затрудняется и более трудоемка.

Защита рокировавшего короля

Расположение рокировавшего короля и защищающих (прикрывающих от атак противника) его фигур может быть различным в разных позициях, но защита в них позиций рокировавшего короля осуществляется некоторыми аналогичными приемами;

1) особо важна роль п е ш е ч н о г о п р и к р ы т и я перед королем;

2) в е р т и к а л ь короля должна быть закрыта пешкой либо фигурой; лучше всего, когда эта пешка (может быть одна из прикрывающих короля) или фигура стоят непосредственно впереди короля;

3) с о с е д н и е с королем в е р т и к а л и, а также д и а г о н а л и, сходящиеся на короле или на какой-либо клетке рядом с королем, должны быть надежно з а к р ы т ы в сторону короля (з а х в а ч е н ы нами либо з а г р а ж д е н н ы м и для соответствующих фигур противника);

4) если у противника имеются тяжелые фигуры, которые могут представить какую-либо опасность с тыла, то г о р и з о н т а л ь короля (может быть, и с о с е д н и е с ней горизонтали) должна к о н т р о л и р о в а т ь с я нами.

Защита нерокировавшего КОРОЛЯ

В защите нерокировавшего короля используются аналогичные описанным в предыдущем пункте приемы. Но следует учесть специфику расположения нерокировавшего короля, который находится в центре одной из крайних горизонталей, тем самым затрудняя координированные защитные действия своих фигур (в первую очередь ладьей, которые находятся по разным сторонам от короля).

Король находитñя в безопасности

Понятие тесно связано с описанными выше двумя понятиями. В общей случае позицию короля назовем б е з о п а с н о й, если:

1) перед королем расположено надежно защищенное п е ш е ч н о е п р и к р ы т и е;

2) каждая наша фигура, непосредственно участвующая в защите короля (из пояса короля или стоящая на подступах к нему) противником
не атакована либо противник не может взять ее или удачно разменять;

3) все поля пояса короля нами к о н т р о л и р у ю т с я;

4) не клетках пояса короля нет фигур противника и ни одна из них не может ходить туда (соприкасаться с королем);

5) линии, сходящиеся на клетках пояса короля, для маневра по которым у противника имеются соответствующие фигуры, з а х в а ч е н ы (к о н т р о л и р у ю т с я) нашими фигурами (или хотя бы отрезки их, проходящие через наш лагерь);

6) ни у одной из пешек противника нет возможности брать пешку из прикрытия нашего короля или разменяться на нее, тем самым открывая доступ к позиции короля;

7) у противника нет возможности дать д в о й н о й или в ñ к р ы т ы й ш а х нашему королю;

8) у противника нет возможности нанести “вилку” или д в о й н о й у д а р по королю и другой нашей фигуре, а также с в я з а т ь какую-либо нашу фигуру в отношении короля;

9) противник не может шаховать нашего короля либо посредством шахов достигнуть какого-либо материального или позиционного преимущества или же выигрыша темпа;

10) у противника нет возможности сконцентрировать удары на наши фигуры, охраняющие короля и доступы к нему, и на поля, захват которых противником ставил бы под угрозу позицию короля (его безпасность).

Позиция короля стеснена

Понятие тесно связано ñ описанным в предыдущем пункте понятием. Опять же рассматривается позиция короля, но только с противоположной точки зрения, т.е. с точки зрения с л а б о с т е й позиции самого короля, а также н е у д а ч н о г о, у я з в и м о г о в той или иной степени расположения фигур, осуществляющих защиту короля и подступов к нему.

Отсутствие “форточки”

Имеется в виду ситуация на доске, когда в результате каких-либо маневров (комбинации) тяжелая фигура (фигуры) могут проникнуть в тыл противника и, пользуясь тем, что королю негде спрятаться от шаха по какой-либо линии (в большинстве случаев - по какой-либо из крайних горизонталей), легко заматовать его. Этим иногда может обернуться отсутствие надежного убежища (“форточки”) от прямых угроз противника

9. Типы шахов

I. Вскрытый шах.
2. Двойной шах.
3. Вечный шах.

Вскрытый шах

Ситуация, когда делавшая ход фигура или пешка открывает по какой-либо линии шах королю противника другой фигурой своего цвета. Опасность же вскрытого шаха заключается в том, что ход первой фигурой, как правило, бывает ф о р с и р у ю щ и м (скажем, угроза взятия какой-либо фигуры противника), против которого противник бессилен сделать что-либо полезное для себя: ему в первую очередь нужно закрыться от шаха, после чего (после второго нашего хода) все равно он потеряет что-либо ценное для себя.

Двойной шах

Является частным случаем в с к р ы т о г о ш а х а (см. предыдущий пункт). В этом случае ходящая фигура не только открывает шах королю противника какой-либо другой своей фигурой, но и сама объявляет шах.

Двойной шах - грозное оружие. Кроме сказанного выше, тут можно добавить, что закрыться от двойного шаха без каких-либо крупных потерь (в большинстве случаев - обеспечивающих давшей шах стороне хорошие перспективы в дальнейшем, вплоть до победы) противнику, как правило,не удается.

Вечный шах

Достижение вечного шаха – “ничейного аппарата”, - цель той из сторон, которая о большем и не мечтает.

Как правило, вечный шах случается в позициях, где противник опережает нас на всей или на почти всей доске (инициативой владеет он, у него хорошее расположение фигур и т.п.), но выясняется, что:

1) у некоторой нашей фигуры есть ход с шахом королю противника;

2) как бы не ходил король противника, закрываясь от шаха или
как бы не закрывала его другая фигура противника, у нас есть возможность каким-либо ответным ходом шаховать его;

3) у нас нет возможности каким-либо образом усилить наши действия.

10. Типы перевесов (преимуществ)

I. Материальный перевес.

2. Пешечный перевес .

3.Позиционный перевес.

4. Пространственный перевес.

Материальный перевес

Наиболее весомое из превосходства в силах, встречающихся в ходе партии.

На пути к достижению главной цели шахматной игры - мату королю противника, каждый из соперников в первую очередь стремится к выигрышу материала (пешки, легкой фигуры, качества и т.д.), что зачастую открывает новые пути к развитию обоснованной атаки, приводящей в конечном счете к победе выигравшей материал стороны.

Пåшечный перевес

Часто встречающийся ÒÈÏ материального перевеса. Его можно интер-претироватъ двояко:

1) общее количество пешек одной ÈÇ сторон больше, чем у против-
ника;

2) количество пешек одной из сторон в какой-либо области доски áольше такого же показателя для противника там же (например, при атаках на определенном фланге в середине игры, когда перевес в пешку делает возможным там пешечный прорыв).

Позиционный перевес

У одной из сторон раоположение фигур в позиции лучше (удачнее), чем у противника. Под этим можно понять следующее:

I) владение какими-либо важными линиями (в первую очередь - вертикалями);

2) фигуры занимают важные с какой-либо точки зрения клетки и
áолее активны, развиты в сравнении с равноценными фигурами противника;

3) король находится в безопасности (надежно прикрыт);

4) слабости защищены лучше, чем у противника либо некоторые из них вообще не атакованы;

5) инициативой владеет рассматриваемая сторона.

Пространственный перевес

Имеет место, когда фигуры одной из сторон м о б и л ь н е е эквивалентных им по ценности фигур противника (например, конь в центре и конь противнике на краю доски).

В общем же случае у одной из сторон есть пространственный перевес, если в каких-либо важных с некоторой точки зрения областях доски фигуры этой стороны имеют большее пространство маневрировать,чем фигуры противника там же.

11. Мотивы (предпосылки) каких-либо действий

I.Мотивы для концентрации.
2. Мотивы комбинации.
3. Мотивы вечного шаха.
4. Мотивы разменов.

Успешность предпринимаемых какой-либо из противоборствующих сторон действий главным образом зависит от того, насколько мотивированны, обоснованны они. По-этому в шахматах бесцельная, недостаточно обоснованная игра, как правило, к добру не приводит (как и недостаточно подготовленные, подкрепленные действия). Таким образом, прежде чем предпринимать какие-либо действия на доске, следует выяснить, есть ли мотивы для их осуществления, не грозит ли все это в конечном счете потерей темпа или инициативы, удудшением позиции или, что хуже, материальным проигрышем.

Мотивы для концентрации

Как известно, концентрация сил (ударов) является подготовительным этапом активных действий группой фигур, направленных на достижение какой-либо заранее поставленной цели, В соответствии со сказанным, мотивы для концентрации могут быть з а щ и т н ы м и и а т а к у ю щ и м и.

Защитные:

1) защита позиции короля;

2) защита удачно расположенных, ключевых Фигур и пешек (скажем, активной фигуры, находящейся в лагере противника или проходных пешек) ;

3) стремление контролировать какие-либо подмножества клеток (области) в собственном лагере, обеспечивающее его неуязвимость.

Атакующие:

1) возможность проведения комбинации;

2) наличие каких-либо слабостей (позиции короля, фигурных, пешечных или пространственных) в лагере противника, что влечет за собой возможность проведения определенных атакующих действий, в результате которых могут быть выиграны темп, фигура

или пешка, захвачена инициатива либо просто улучшена позиция с дальнейшими, лучшими, чем у противника, перспективами на удачное проведение игры.

Мотивы комбинации

Комбинацией называются совместные действия нескольких фигур в виде форсированного варианта с жертвой материала, в результате которого играющий рассчитывает получать какие-либо выгоды (размен активных фигур, завладение важными линиями или полями, расстройство пешечного расположения противника, достижение вечного шаха или мата, выигрыш материала, мат королю противника и т.п.).

Непременным условием для успешного проведения комбинации является наличие у противника каких-либо слабостей.

Мотивами комбинации могут служить:

1) нахождение неприятельских фигур на прямой линии, т.е. мотив чисто геометрического порядка;
2) завлечение неприятельских фигур под двойной удар коня;
3) завлечение короля противника, сочетаемое с таким превращением пешки, при котором возникает решающий удар;
4) стесненное положение неприятельских фигур;
5) все виды ограничения подвижности- отрезание полей, заграждения, близость к краю доски и т.п.
6) материальное превосходство в силах у рассматриваемой стороны в какой-либо области доски (например, пешечное превосходство на определенном фланге).

Мотивы вечного шаха

В шахматных партиях нередко встречается вечный шах -ситуация, когда одна из играющих сторон заканчивает партию вничью,при этом, как правило, "спасая партию". Это ей удается при определенных условиях, возникших на доске (нередко - вследствие проведенной для достижения этой цели комбинации), после каждого хода противника атакуя его короля.

В большинстве случаев возникновения ситуаций, когда возможен вечный шах, тот из играющих, для которого достижение ничьей посредством вечного шаха уже является главной целью, как правило (и как уже ясно из вышесказанного), имеет позицию ничего хорошего для него не обещающую: инициативой владеет противник и его фигуры доминируют по всей доске; еще венного и выяснится, что нет никакого спасения от неминуемого проигрыша. Но именно в этот момент выясняется совсем другое:

1) король противника недостаточно защищен - он вышел из-за пешечного прикрытия, пешечного прикрытия уже не существует либо в какой-то момент король был вынужден начать "марш" не по тому направлению (скажем в центр, в плохо защищенную своими фигурами область доски и т.п.);
2) разрозненность по всей доске фигур противника,их усилий (случай, когда угроз у противника слишком много и его фигуры действуют в отдалении друг от друга).

И тогда, при наличии соответствующих фигур (часто - одной единственной), выясняется главное - от шахов королю противника этими фигурами нет сколь-нибудь вразумительных защит (королю не за чем спрятаться). EÑËÈ же противник будет стремиться каким-либо образом уклониться от известного исхода, то он рискует из делавшей ничью стороны сразу превратиться в проигрывающую.

Мотивы разменов

Общеизвестно, что от удачных, хорошо продуманных разменов многое зависит в шахматной игре.

Вовремя заметить и использовать к месту подвернувшуюся возможность удачного размена, значит сделать все возможное на пути достижения к еще большему в данной стадии партии.

Мотивами для удачных, суливших дальнейшие,лучшие, чем у противника, перспективы, разменов могут служить:

1) размен с последующим выигрышем темпа (обмен своей неразвитой фигуры на развитую фигуру противника);
2) меняться, чтобы без потери времени занять (либо открыть) линию;
3) уничтожить защищающую фигуру противника;
4) не терять времени на отступление;
5) стремление упростить позицию, когда у нас есть материальный перевес.

12. Типы позиций

1. Спокойные позиции.
2. Открытые позиции.
3.Закрытые позиции.
4. Полуоткрытые позиции.
5. Статичные позиции.
6. Динамичные позиции.
7. Опасные позиции.
8. “Смертельные” позиции.
9. Определившиеся позиции.
10. Критические позиции.
11. Стесненные позиции.
12. Цугцванг.
13. Динамическое равновесие.

Спокойные позиции

Это позиции, из которых, при наивероятнейших ходах сторон (т.е. при правильной их игре), невозможны:

1) комбинации с целью достижения чего-либо весомого;
2) размены с какими-либо хорошими шансами для одной из сторон;
3) материальный выигрыш;
4) достижение позиционных выгод на длительное время с их последующим применением для достижения большего в дальнейшем.

Открытые позиции

Позиции, где, как правило, ведется а к т и в н а я к о м б и н а ц и о н н а я борьба, и где, в силу этого, возможности сторон могут быть сравнительно легко оценены, называются о т к р ы т ы м и.

В таких позициях:

- большая часть фигур сторон развиты (расположены они в той или иной степени удачно);
- большинство фигур сторон приведены в боевую готовность и соприкосаютея друг с другом;
- на доске ведется активная борьба за определенные выгоды (позиционные или материальные).

Закрытые позиции

Позиции, для которых больше характерна п о з и ц и о н н а я борьба, и где оценка возможностей сторон сильно затруднена, называются з а к р ы т ы м и.

В таких позициях:

- большая часть фигур сторон расположена на доске далеко не лучшим образом, но пока не ясны пути для дальнейшего улучшения их расположения;
- фигуры сторон не приведены в непосредственное соприкосновение; они находятся в своих лагерях или на их подступах;
- борьба ведется исключительно с целью получения каких-либо позиционных (пространственных) выгод и их постепенного накопления.

Полуоткрытые (полузакрытые) позиции

Позиции, имеющие характеристики как о т к р ы т ы х, так и з а к р ы т ы х (см. выше), но, в силу определенных обстоятельств, не попавшие ни в тот, ни в другой класс, принято называть п о л у о т к р ы т ы м и (п о л у з а к р ы т ы м и).

Этот класс позиций несравненно богаче позициями, чем вышеупомянутые два других класса. Если иметь в виду факт, что понятие “полуоткрытая (полузакрытая)”весьма расплывчато, то этот класс позиций большого интереса не представляет.

Статичные позиции

Это позиции, которые можно оценить главным образом статическими методами (т.е. с помощью статических характеристик).

В таких позициях, как правило, уже достигнутое сторонами на доске сохраняется (при целесообразных, правильных ходах) на некоторое время (количество ходов). Одним словом, в них на некоторую глубину анализа не предвидятся квкиå-либо достаточно обоснованны атакующие, активные действия.

Динамичные позиции

Так называют позиции, по своей структуре противоположные по отношению к статичным позициям (см. выше).

В них возможны следующие ситуации:

1) одна из сторон проводит активные действия (т.е. в позиции последний фиксирован какой-либо из промежуточных ходов комбинации и т.п.);

2) из позиции возможны какие-либо активные, атакующие действия (комбинации и т.п.).

Опасные позиции

Можно так назвать позиции, в которых на определенный ход(ходы) одной из сторон у противника нет сколь-нибудь вразумительных ответов (при анализе дерева игры на некоторую, заранее фиксированную глубину).

При этом, предполагается, что при правильной игре противника достигаемое этим ходом (ходами) преимущество первой стороны-отнюдь не решающее или, другими словами, позиция для противника (при правильной его игре, разумеется) - не проигрышна.

“Смертельные” позиции

Так фигурально называют класс позиций, из которых для одной из сторон имеется хотя бы один ход, который является началом или промежуточным звеном форсированной цепочки ходов, ведущей к выигрышу партии этой стороной.

Определившиеся позиции

Это позиции, оценка которых может быть произведена на основании ряда позиционных факторов без дополнительного вариантного расчета.

Как видим, определение данного класса почти полностью совпадает с определением с т а т и ч н ы х позиций (см. выше).

<u>Критические позиции</u>

Значительно сложнее производить оценку позиций, которые не приняли о п р е д е л и в ш е г о с я характера (см. предыдущий пункт). Эти неясные, с еще не выясненными реальными возможностями позиции называют к р и т и ч е с к и м и. Цель анализа в *ЭТОМ* случае состоят в том, чтобы прийти к позиции, которая носила бы уже не критический, а определившийся характер.

<u>Стесненные позиции</u>

На уровне позиций понятие “стесненности” означает следующее:ударная сила одной из играющих сторон ограничена в своей подвижности (собственные ее фигуры мешают друг другу, противник заграждает и отрезает им путь и т.п.), вследствие чего, как правило, король этой стороны оказывается в опасном (иногда - в безвыходном) положении.

<u>Цугцванг</u>

Класс позиций, в которых в силу определенных обстоятельств (часто вследствие удачной игры одной из сторон) возникла ситуация, когда право хода за противником и, что главное, делая п р о и з в о л ь н ы й из осмысленных ходов, противник, как правило, несет крупные потори(вплоть до проигрыша). Другими словами, на каждый, кажущийся целесообразный ход противника у первой стороны есть по крайней мере один сильный ответ, после которого полностью овладевая инициативой, эта сторона решает исход партии (в большинстве случаев) в свою пользу.

<u>Динамическое равновесие</u>

Так говорится о возникшей ситуации в тех позициях, где возможны активные действия сторон, но на каждый удар есть “контрудар”. Другими слоавми, при достаточной глубине изучаемых вариантов, начинающихся от некоторой позиции из исследуемых,и, разумеется при предполагаемых целесообразных ходах сторон, выясняется, что материальное и позиционное равновесие сохраняется.

<u>13. О дебютных стратегиях</u>

I. Развитие фигур.
2. Замедленное развитие фигур.
3. Несвоевременное (преждевременное) развитие фигур.
4. Централизация.

Общеизвестна роль удачно разыгранного дебюте. Вознаграждением успешно проведенного дебюта может стать закономерная и заслуженная победа, пусть даже в результате тяжелой и изнурительной борьбы за реализацию некоторого перевеса (материального или позиционного) в отнюдь не элементарном эндшпиле. Напротив, пассивная, недостаточно обоснованная игра в дебюте может привести к катастрофе еще в дебюте (не говоря о том, что серьезные промахи в дебюте, как правило, не удается компенсировать даже блестящей игрой в дальнейшем).

<u>Развитие фигур</u>

Является первой и главной из двух важнейших стратегических задач в дебюте.

Задача эта заключается в том, чтобы своевременное, без потери темпа и максимально выгодным образом развить свои фигуры, начиная с первых не дебютных ходов.

В соответствии с конкретными типами дебютов (начальных 10-15 ходов в партии) имеются в той или иной степени стандартизованные методы и способы для гармоничного развития основной части фигур, использующихся в ходе дальнейших событий. Главная же их суть заключается в следующем:

1) начальные ходы пешками (в большинстве случаев - центральными), затем - конями либо наоборот;

2) дальнейшие ходы слонами и ферзем (может быть, и другими пешками), с развитием которых одновременно преследуется цель подготовить рокировку короля;

3) дальше, после стабилизации положения в центре доски (возможные размены пешек и легких фигур, укрепление там своих позиций) , как правило, осуществление рокировки и постепенное включение в борьбу ладьй).

Следует помнить, что чем более о т к р ы т ы й характер носит позиция, тем сильнее проявляется перевес в развитии фигур. А з а к р ы т а я позиция за несколько активных ходов может превратиться в открытую!

Замедленное развитие фигур (отставание в развитии)

Обусловлено слабыми, непродуманными начальными ходами, при которых обычно т е р я е т с я т е м п. Другими словами, в одном из наиболее важных этапов шахматной партии нарушается основной принцип борьбы: не существует отдельных удачных ходов, а существуют удачные серии ходов, координированная игре всеми фигурами, т.е., в дебюте успешность хода нужно рассматривать на фоне происходящих на доске событий и в связи с некоторой поставленной целью.

Н е к о т о р ы е п р и б л и з и т е л ь н ы е р е к о м е н д а ц и и

1) При строжайшем, экономном расходовании темпов каждым ходом должна преследоваться цель развития фигур или борьбы за центр. Особенно ценно, кода одним ходом разрешаются обе задачи;

2) не ходить дважды в дебюте одной и той же фигурой, а мобилизовать за это время другую фигуру;

3) не терять времени на б е с п о л е з н ы е ходы крайними пешками „а“ и “h” .

Несвоевременное (преждевременное) развитие фигур

Другая крайность, связанная с плохо продуманными, необоснованными дебютными ходами.

В этом случае, не достигая на доске ничего определенного или перспективного в результате предыдущих ходов, без какой-либо необходимости в бой перебрасываются все новые и новые силы, что, в конце концов, может обернуться весьма ощутимыми (часто - непоправимыми) потерями как в позиционном, так и в материальном смысле. И опять же нарушается основное правило шахматной игры, которое в дебюте еще и имеет решающее значение - действия фигур должны быть направлены на достижение какой-либо цели и быть планомерными.

Н Å к о т о р ы е п р и б л и з и т е л ь н ы е р е к о м е н д а ц и и

1) не выводить преждевременно ценные фигуры (скажем - ферзя), чтобы противник не мог развивать свои силы с темпом, т.е., нападая на первые;

2) не предпринимать скороспелых, недостаточно подготовленных атак одной - двумя фигурами, так как противник легко отразит их, одновременно продолжая мобилизацию (развитие) собственных сил, мы же рискуем безнадежно отстать в развитии.

Централизация

Является второй из двух важнейших стратегических задач в дебюте. Она заключается в том, чтобы мобилизовать свои фигуры и пешки с целью занятия или контролирования центра доски. Если иметь в виду то, что любая фигура в центре (за исключением ладьи) атакует максимальное число полей (обладает максимальной боеспособностью), а также то, что фигуры именно из центра в кратчайшее число ходов могут быть переброшены на любой из флангов, становится ясно, что контролируемый (занятый) своими фигурами и пешками центр доски является главным плацдармом для проведения дальнейших успешных атак.

Некоторые рекомендации(приблизительные) и замечания.

1) Наиболее удобно, гибко фигурно-пешечное воздействие на центр;

2) именно пешки более всего пригодны для образования центра,
так как они наиболее устоûчивы, однако размещенные в центре
фигуры вполне могут заменить пешки. Заменить обладание центром может и давление, производимое собственными ладьями или слонами на неприятельский центр;

3) игрой в центре (прорыв или занятие центра) можно наказать противника за преждевременные фланговые атаки, в то же время прочность собственных позиций в центре определяет успех предпринимаемых атак на фланге;

4) дебютные фланговые атаки - это важное дополнительное средство борьбы за центр, состоящее в отказе от лобовой атаки в пользу фланговых ударов и обходов. Таким путем мы стремимся уменьшить и ослабить воздействие фигур противника на центр, чтобы добиться там превосходства.

14. О темпе и инициативе в игре

1. Замедление темпа.
2. Выигрыш темпа.
3.Потеря темпа.
4. Развитие инициативы.

Под темпом в шахматах понимается уже сложившийся в результате предыдущих действий противоборствующих сторон ритм игры в данный фиксированный момент.

Первостепенной задачей каждого из играющих является всеми способами и целесообразными ходами поддержать темп игры, не отстать в этом отношении от противника.

Тесно связана с темпом так называемая инициатива в игре.В основе этого понятия лежит следующая шахматная истина: предпочтение в игре всегда отдается тому из играющих,планы действий которого в сравнении с аналогичным планами противника более мощны, живучи, гибки, разносторонни, реальны и т.п. В этом случае считается, что инициатива ведения активных, атакующих действий именно за этой стороной.

Всесторонне развить инициативу, ни на миг не отпуская ее из рук, значит с каждым ходом уверенно приблизиться к заветной цели - победе над противником.

Замедление темпа. В*ЫИГРЫШ* темпа.
Потеря темпа.

Иногда, когда один из соперников не в силах проникнуть в планы другого или же просто не успевает достаточно успешно противодействовать его мощным атакующим действиям, у него получается опоздание в игре в виде целого хода. Тогда говорят о выигрыше темпа его противником или о потеретемпа этой стороной. Отсюда ясно, что любое другое отс т а в а н и е в игре (если оно может быть оценено, что не всегда так) одного из соперников можно назвать замедлением темпа.

Другими словами, если противник не имеет времени для хода, во всех отношениях для него полезного, и притом потому, что почему-либо вынужден к малозначащему в позиционном отношении ходу, в то время, как другая сторона “двигается вперед”, то эта другая сторона явно в ы и г р а л а т е м п.

Н е к о т о р ы е р е к о м е н д а ц и и и з а м е ч а н и я

1) Позиция короля стеснена, он может стать мишенью атак противника - вероятна потеря темпа;

2) потеря темпа возможна, когда вместо того, чтобы перебросить в нужную область доски некоторую фигуру кратчайшим путем, она перебрасывается туда хотя бы ходом позже;

3) лишь в тех случаях можно потерять время т.е. темп), если это дает прочное обладание важными пунктами;

4) оказывается, что можно выиграть темп, потеряв ход, т.е. передав право его выполнения противнику, но в то же время приблизиться к своей основной цели (цугцванг);

5) в дебютах, как правило, п о в т о р н ы е х о д ы одними и теми же фигурами, равно как и п р е ж д е в р е м е н н о е р а з в и т и е ценных фигур, на которые противник может напасть менее ценными фигурами, чреваты потерей темпа;

6) в дебютных позициях з а к р ы т о г о характера допустимы потери темпа (повторные ходы фигурами, движение крайними пешками), при условии, что это оправдано. Это зависит главным образом от того, насколько успешным в данном случае будет контроль над центром.

Развитие инициативы

Тот из противников, кто смог в данной позиции найти оттенки, которые другой не нашел, и использует их, у того и и н и ц и а т и в а. Это так, потому что противник не может играть не учитывая плана своего соперника (он хотя бы видит его ходы на доске). И, начиная с некоторого момента, у него действия (усилия) направлены преимущественно не на реализацию своего плана, а противодействию плану (ходам) соперника.

Владея инициативой, нужно стремиться всячески р а з в и в а т ь ее, продолжая успешно начатое дело и, вместе с тем, создавая все новые и новые угрозы фигурам, пешкам и королю противника. Нужно помнить, что иногда достаточно и з а м е д л е н и я темпа, чтобы противник мог одной успешной к о н т р а т а к о й свести на нет все наши усилия, одновременно п е р е х в а т и в у нас инициативу.

15. Типы действий (игры) сторон в каком-либо промежутке партии

1. Игра позиционная.
2. Игра комбинационная.
5. Игра координированная.

Игра позиционная

Под этим подразумевается серия ходов на позициях, где все хорошо защищено (нет непосредственных угроз взятия фигур и пешек), нет каких-либо активных ходов (скажем-шахов), возможностей проведения комбинаций. Борьба на доске в этой случае ведется главным образом за пространство и удачное расположение своих фигур и пешек (захват определенных клеток, контроль над важными с некоторой точки зрения областями доска, развитие фигур и пешек и т.п.). Заново создаются п л а ц д а р м ы, ф о р п о с т ы, о т к р ы в а ю т с я или з а к р ы в а ю т с я нужные л и н и и- одним словом создаются всевозможные условия для успешного проведения будущих атак.

Игра комбинационная

Серия ходов, где ведется активная борьба на уровне к о м б и н а ц и й, а к т и в н ы х ходов.Ими могут быть: вçятия фигур и пешек, размены, жертвы, связки, двойные удары, шахи и т.п.

Â общем случае, наличие каких-либо мотивов атаки (комбинации, серии активных ходов) в лагере одной из сторон и возможность использования их противником указывает на вероятность ведения на доске к о м б и н а ц и î н н о й игры.

Игра координированная

К о о р д и н и р о в а н и е действий фигур для каждой из играющих сторон имеет решающее значение как при атаке на позиции противника с целью захватить что-либо ценное, важное, так и при защите чего-либо ценного в собственном лагере.

В основе этого понятия лежит факт, что сколь-нибудь значимые результаты являются естественным следствием успешно проведенных координированных действий группы фигур, другими словами, гармоничным взаимодействием этих фигур. Но следует отметить, что здесь слово “взаимодействие" понимается несколько шире, чем скажем то же понятие в отношении фигур в некоторой конкретной области доски. В этом случае имеются в виду совместные действия группы фигур, усилия которых объединены воедино для достижения определенной цели (задачи) и, что очань важно, границы такого взаимодействия часто определяются только размерами доски.

16. Типы фигур центров
1. Фигурный центр.
2. Пешечный центр.
3. Фигурно-пешечный центр.

Одним из наиболее важных дебютных принципов является построение мощного, гибкого центра, ñ целью захватить ñ первых же дебютных ходов инициативу и, если это удалось, начать активную борьбу за приобретение все новых и новых позиционных и, при возможности, материальных благ на доске.

Центры могут быть трех типов:
1) ф и г у р н ы й центр (центр контролируется только фигурами);
2) п е ш е ч н ы й центр (в центре и на его подступах стояк только пешки);
3) ф и г у р н о-п е ш е ч н ы й центр (центр контролируется совместными усилиями фигур и пешек.

Более подробно о центрах говорится в п. “Централизация" (см. §).

17. Типы ходов
1. Слабый ход.
2. Сильный ход.
3. “Остроумный” ход.
4. Стратегический ход.
5. Тактический ход.
6. “Смертельный” ход.
7. “Симпатичный” ход.
8. Целесообразный (полезный) ход.

9. Форсирующий (форсированный) ход.
10.“Тихий” ход.
II. “Выжидательный” ход.
12. Единственный ход.

Слабый ход

1) замедляет темп игры и может привести к его проигрышу;

2) ослабляет инициативу;

3) может сильно ослабить уже завоеванные позиции на доске, сведя на нет потраченные на это усилия;

4)осложняет защиту каких-либо фигур или областей доски;

5) дает возможность проведения противником контрудара.

Сильный ход

1) увеличивает темп игры;

2) развивает инициативу;

3) объект защиты полностью защищается, и тем самым создаются условия для ведения контригры (контратаки);

4) усиливаются позиции собственных фигур или короля на доске;

5) ослабляет инициативу противника, мешая проведению его атаки, прерывая ее или делает ее просто невозможной.

“Остроумный” ход

С первого взгляда такие ходы могут показаться необоснованными, противоречащими возникшей на доске ситуации. Однако, после того, как выяснилось, что делающая такой ход сторона имела на это веские причины или мотивировка хода достаточно убедительна, все становится на свои места. “Остроумные” ходы - это результат достаточно продуманной игры и когда то, на чем он основывается и которого касается, выступает на позиции неявным, скрытым образом.

Стратегический ход

Стратегическими можно считать ходы, обычно сильные, главное значение которых заключается в том, что они фиксируют определенные величины на доске (блокировка фигур, контроль над клетками и т.п.) на время, достаточное для того, чтобы они пригодились в некоторый момент на пути достижения поставленной заранее, но дальней пока цели.

Ясно, что ценность таких ходов не всегда оказывается очевидной в позиции, где он сделан.

Тактический ход

В отличие от стратегических, это ходы в рамках тактических действий, часто направленные на достижение чего-либо конкретного, очевидного в позиции. Другими словами, они делаются из соображений добиться каждым таким ходом некоторого промежуточного, малозначащего (если рассмотреть его отдельно) результата на пути достижения какой-либо, но ближайшей цели, поставленной перед нами.

Такие ходы, как правило, поддаются точной оценке (имеется в виду, прежде всего, производимый ими на позицию конкретный эффект).

“Смертельный” ход

Ход, который сам еще не являясь выигрывающим, содержит неразрушимую, неотразимую угрозу, после осуществления которой на доске достигается выигрыш либо такое материальное (редко-позиционное) преимущество, после чего дальнейшая борьба для противника бессмысленна.

Целесообразный (полезный) ход

Имеются в виду несслабые ходы (но на обязательно сильные), которые делаются в результате такого анализа возникшей на доске ситуации, когда выяснено истинное состояние дел в позиции и выработаны реальные планы на достижение некоторых целей, начиная с самых ближайших.

ФОРСИРУЮЩИЙ (форсированный)ход

Ф о р с и р у ю щ и м и называются ходы особой силы, преследующие цель достижения каких-либо выгод и допускающие, как правило, немногочисленные (нередко-единственный) ответные ходы противнике. Часто всего такие ходы встречаются в комбинациях.

Форсирующими ходами могут быть:

1) двойные удары;
2) шахи;
3) взятия фигур или пешåк;
4) превращение пешек;
5) угрозы взятия (угроза выигрыша материала);
6) угроза мата;
7) угроза занятия важного в каком-либо отношении поля;
8) угроза связки (тем более - связка).

“Тихий ” ход

Понятие, тесно связанное со стратегическими ходами. Под “тихими” ходами опять же понимаются малозаметные в исследуемой позиции, не эффектные на первый взгляд ходы, эффективность которых однако доказывается сделавшей такой ход стороной гораздо позже, когда другими (такими же, может быть) ходами создаются необходимые для штурма некоторой, поставленной еще в исследуемой вначале позиции, цели.

“Выжидательный” ход

Имеются в виду продуманные, не бесцельные, неслабые, хотя и не приводящие, как правило, ни к какому преимуществу (материальному или позиционному) на доске, ходы в позициях, где возможности противников примерно равны, как и их шансы на будущее, но, в силу сложившихся обстоятельств, дающий первым сигнал к активным действиям рискует многим. По-этому и делаются “выжидательные” ходы – при возможности накапливая позиционные преимущества и, если противник начнет активные действия, быть готовым нанести контрудар.

Единственный ход

Может быть:

1) единственным ответным ходом на очень сильный, форсирующий ход противника;

2) единственным полезным ходом из данной позиции среди ходов из некоторого подмножества неслабых ходов, невыполнение которого привело бы к материальному или, в лучшем случае, ощутимому позиционному урону.

Общие замечания к дескрипторам

I. В дальнейшем имеет смысл пересмотреть подход к понятию “фигура связана”, поскольку это не одно и тоже, когда в одном случае некоторая фигура игрока атакует более ценную фигуру противника и по той же линии за этой фигурой противника стоит что-либо ценное или незащищенное, а в другом случае та же фигура игрока атакует менее ценную фигуру противника, за которой опять что-либо ценное или незащищенное.

В первом случае может оыть даже взята более ценная связанная фигура противника, а во втором - важен сам факт связки (ведь даже связанная пешка служит мотивом для разрешающей все комбинации).

2. Заметим, что в силу ряда обстоятельств шахматные понятия, лежащие в основе дескрипторов, в той или иной мере упрощены или даже изменены. Другими словами, они являются “слабыми” эквивалентами соответствующих шахматных понятий.

Имея в виду сказанное, нужно стремиться к расширению и улучшению этих понятий, чтобы можно было в дальнейшем использовать их в качестве полноценных шахматных целей (относится в первую очередь к дескрипторам “Легкая фигура на дырке”, “Изолированная пешка на открытой линии”, “Фаланга пешек”, конфигурации “Конь-пешка”, “Пешки вне атаки слона”, “Возможность пешечного прорыва на каком-либо фланге”, “Конь в центре”,”Пешка в центре”, “Пешка бьет в центр”, “Атака слабых пешек противника”, “Защита своих активных пешек” и т.д.).

3. Дескриптор “фаланга пешек” можно построить аналогично дескриптором блока “Конь-пешка”, то есть следующим образом:
“Пешки (P,*Q*) и *(R, S)* образуют фалангу”. Тогда в функциональную схему должны быть добавлены соответствующие ветви возможности создания фаланги пешкой (R,S) (т.е. – “слева”), а рядом с (043)будет проверяться условие “На клетке *(P+1,Q)* стоит пешка того же цвета”.

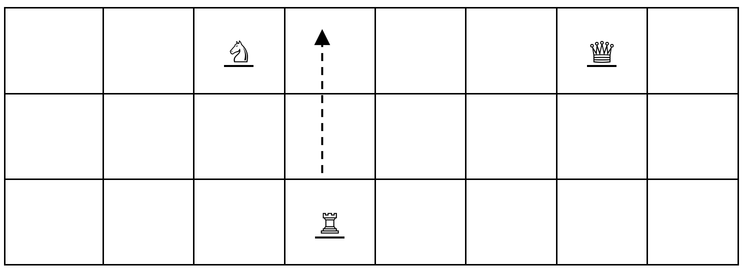
4. Очень важным является вопрос о глубине анализа пути достижения исследуемой цели на шахматной доске. Интересно рассмотреть дело с технической точки зрения, какие трудности (связанные с усложнением анализа на доске) возникают.

<u>Замечания к дескрипторам №№ 19-24</u>
<u>(Блок “Двойной удар”)</u>

1. Следует обратить внимание на тот факт, что возможен двойной удар на пару фигур противника, одна из которых равносильна фигуре игрока, наносящей двойной удар - когда эта вторая фигура противникà связана.

2. В дескрипторах №№ 20-24 могут возникнуть ситуации, когда ñ более ценной, чем атакующая фигура игрока, фигурой под двойным ударом может попасть другая фигура противникà, которая защищена только вышеупомянутой ценной фигурой, и двойной удар фигурой игрока может ок азатьсявозможным.

<u>Пример.</u>

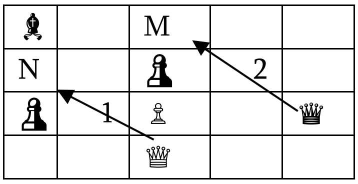


3. Возможны еще ситуации, когда фигура ходит и создает двойной удар на две фигуры противника, которые не могут защитить друг друга, но у противника есть такой ответный ход некоторой третьей фигурой, в результате которого обе фигуры, попавшие под двойной удар, защищаются.

При этом могут возникнуть три случая:

а) у этой третьей фигуры противника есть ход, в результате которого она сразу защищает обе фигуры, попавшие под двойной удар.

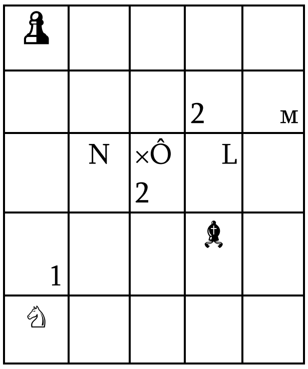
Пример



Первый ходом белый ферзь ходит на клетку *N* и наносит двойной удар по черным слону и пешкам, однако ответным ходом черного ферзя на клетку M все эти фигуры защищаются им;

б) такой возможности, как описанная в предыдущем пункте, у вышеупомянутой трерьей фигуры противника нет, но у нее есть ход, в результате которого одну из фигур, находящихся под двойным ударом, защищает она сама и вместе с этим "открывает" защиту второй фигуры первой.

Пример

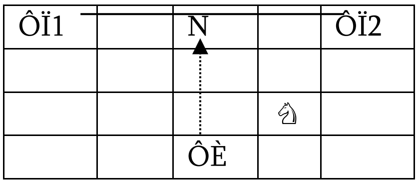


Белый конь ходит на клетку N и наносит удары по незащищенным пешке и слону черных. Однако, следующим ходом фигура черных ЧФ может «открыть» защиту пешки слоном, одновременно защищая слона (если ЧФ-конь, то это можно осуществить с клетки M, если же ЧФ – ладья, то - с клетки L);

в) аналогичная описанной в предыдущем пункте ситуация, в которой вышеупомянутая фигура ходит, одновременно "открывая" защиту одной из находящихся под двойным ударом фигур на другую и защиту некоторой четвертой фигурой на первую из находящихся под двойным ударом.

4. (В дополнение к п.2)

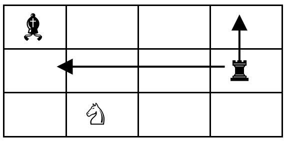
Следует учесть, что при поиске возможностей двойных ударов фигурами, ходящими по линиям (ферзь, ладья и слон) может встретиться такая (и аналогичные ей) ситуация, где



- ФП1 и ФП2 - фигуры противника;
- ФИ- фигура игрока.

Причем ФП1 ценнее ФИ, и ФП2 защищается фигурой ФП1. У фигуры ФИ есть возможность пойти на клетку **N** и оттуда нанести одновременный удар по ФП1 и ФП2. И если у ФП1 нет хода с защитой ФП2 (скажем, по той не линии), то имеет место двойной удар.

5. В случае, когда под двойным ударом находятся ладья и какая-либо другая фигура противника, причем ладья не защищает эту вторую фигуру, в дескрипторе не рассмотрены следующие случаи:



Нужно также учесть, что такие ситуации возможны только при рассмотрении двойных ударов ферзя и коня.

Appendix 8: On the Background of RGT Solvers, OJP and Cognizers

1. Б. К. КАРАПЕТЯН, **ОБ ОДНОЙ УНИФИКАЦИИ ПРЕДСТАВЛЕНИЯ ШАХМАТНЫХ ПОНЯТИЙ,** Mathematical Problems of Computer Sciences, 1986, pp. 181-193.

1. ВВЕДЕНИЕ

Эффективное решение комбинаторных проблем управления связано с использованием различных типов знаний, позволяющих ограничить область поиска решений, определить множество действий, уместных в конкретной ситуации, учесть опыт решения аналогичных задач и т. д. Все указанные аспекты использования знаний характеризуются применением специальных конструкций для задания классов ситуаций с общими свойствами. В [1] рассмотрена возможность описания классов ситуаций посредством формул языка первого порядка. Выделен класс специальных алгоритмов– дескрипторов, вычисляющих формулы языка и предназначенных для распознавания множеств объектов универсума одинаковой полезности, их взаимных отношений, в частности, возможности преобразования одних множеств в другие единообразными средствами. Исследуемые свойства дескрипторов иллюстрируются на примере шахматной задачи, для которой построен универсум, позволяющий получить интерпретацию формул языка первого порядка, адекватную исследуемым вопросам.

В рамках указанного подхода в [2] описана реализация фонда дескрипторов для ЭВМ «Минск– 32». Дескрипторы, составляющие фонд, охватывают широкий круг шахматных понятий, для каждого из которых построено соответствующее формульное описание и составлена фортран– программа, вычисляющая истинность формулы на заданной позиции. Анализ этой реализации выявил ряд недостатков, обусловленных как неприспособленностью языка программирования, так и слабой структурной организацией фонда дескрипторов. Эти недостатки (подробно перечисленные в [3]), ограничивают возможность расширения фонда, усложняют модификацию программ, их сопровождение и перевод на ЭВМ серии ЕС.

Разработка языка описания стратегий (*SDL*) [4], в свою очередь, потребовала создания средств вычисления дескрипторов. В языке *SDL* дескрипторы используются для целей описания спецификаций рассматриваемых стратегий, задания исходных и целевых классов позиций в закономерностях, описания условных выражений управляющих операторов языка *SDL* и др. В настоящей работе описывается программная компонента пакета АКА– У2, предназначенная для вычисления в текущей позиции дескрипторов формул определенного вида.

При разработке описываемой компоненты (названной DESC– VALUE) основное внимание уделено обеспечению простоты модификации и пополнения фонда дескрипторов без существенных программных изменений. С этой целью:

1) Определен конкретный класс формул, допустимых для описания, дескрипторов, а именно формулы языка первого порядка с ограниченными кванторами в предваренной форме.

2) Обеспечена возможность обработки формул без дополнительного кодирования.

3) Разработаны структуры данных для основных элементов универсума, обеспечивающие систематизацию используемых объектов (клетки, позиции и т. д.).

4) Выделено множество первичных шахматных понятий, реализованных функциями, с помощью которых описываются более сложные дескрипторы.

5) Определены объекты специального типа – «доски», позволяющие унифицировать процедуры вычисления атомных формул и основных функций.

6) Разработаны средства диагностики синтаксической корректности формул.

2. ФОРМУЛЬНОЕ ПРЕДСТАВЛЕНИЕ ШАХМАТНЫХ ПОНЯТИИ

1. Шахматные понятия формализуются нами в ограниченном языке первого порядка [5]. Дескриптор формулы некоторого понятия определяет последовательность действий, необходимых для вычисления значения этой формулы в произвольной позиции, т. е., по существу, каждый дескриптор задает конкретизацию формулы в некоторой структуре.

Ниже приводится тип допустимых формул, посредством которых задаются рассматриваемые нами шахматные понятия, и специальная структура для их интерпретации. Описываемая далее программная компонента DESC– VALUE выполняет функцию компилятора для допустимых формул, а именно, обеспечивает конкретизацию формулы в рассматриваемой структуре для заданной позиции и переводит формульное описание понятия на язык, доступный вычислительной машине.

2. Пусть L язык первого порядка [5].

Термом языка L называется выражение, удовлетворяющее следующим условиям:

(i) всякая переменная или константа есть терм;

(ii) если f – n-местный функциональный символ языка L, а $t_1, t_2, \ldots, t_n$ – термы, то $f(t_1, t_2, \ldots, t_n)$ – также терм.

Атомной формулой называется выражение вида $p(t_1, t_2, \ldots, t_n)$, где p – n-местный предикатный символ языка L, а $t_1, t_2, \ldots, t_n$ – термы.

Правильно построенной формулой языка L (ППФ) называется выражение, удовлетворяющее следующим условиям:

(i) всякая атомная формула есть ППФ;

(ii) если .4 и B – ППФ, то $\neg(A)$, $(A \vee B)$, $(A \,\&\, B)$ и $(A \rightarrow B)$ – также ППФ;

(iii) если A – ППФ, то $\forall x(A)$ и $\exists x(A)$ также ППФ.

ППФ находится в пренексной форме, если она:

(i) либо не содержит кванторов,

(ii) либо имеет вид $Qx_1, \ldots, Qx_n\, A$ (Qx_i есть либо $\exists x_i$, либо $\forall x_i$), $x_1, \ldots, x_n$ – различны и A – не содержит кванторов; в этом случае $Qx_1, \ldots, Qx_n$ называется приставкой, а A – матрицей. Можно показать, что для любой ППФ можно построить эквивалентную ей формулу в пренексной форме.

Для формул вида $\forall x(A \rightarrow B)$ либо $\exists x(A \,\&\, B)$, где A – формула с единственной свободной переменной x, введем специальные обозначения $\forall x(A(x)(B)$ и $\exists xA(x)(B)$. Такие формулы будем называть формулами с ограниченными кванторами.

Для описания шахматных понятий мы используем формулы в пренексной форме с ограниченными кванторами.

3. Опишем конкретную структуру **A** для языка L, в которой интерпретируются допустимые формулы. В универсум структуры **A**, кроме индивидов, перечисленных в [1] (константы для обозначения клеток, фигур, позиций и т. д.), добавим индивиды специальных типов – «доски». Доска – это массив размерности 8 на 8, каждый элемент которого может принимать значение 1 или 0.

Если B обозначение переменной или константы типа доска, то компоненты B далее будем обозначать как $B(i, j)$, полагая, что $i, j \in \{1, 2, \ldots, \quad 8\}$.

Переменные и константы типа доска описывают используемые первичные шахматные понятия, которые зависят от геометрических характеристик позиций и от взаиморасположения фигур. Например, понятие четвертая вертикаль описывается константой B типа доска, значения компонент которой следующие:

$$B(i,j) = \begin{cases} 1, \text{если } j = 4, 1 \leq j \leq 8 \\ 0, \quad \text{в противном случае.} \end{cases}$$

К функциям структуры А отнесены базовые функции, перечисленные в табл. 1, а также двухместные функции объединения и пересечения досок, обозначенные символами ∪ и ∩. Термы $B^1 \cup B^2$ и $B^1 \cap B^2$ языка L интерпретируются следующим образом: если O – один из символов ∪ либо ∩, а B^1 и B^2 – доски, то значение терма $B^1\ O\ B^2$ есть доска, компоненты которой определяются дизъюнкцией (если O – это символ ∪), либо конъюнкцией (если O – это символ ∩ соответствующих компонент.

К предикатам структуры **А** отнесены бинарные предикаты равенства и включения, определенные на индивидах типа доска, а также унарные предикаты, выделяющие области изменения переменных.

Таблица 1

Базовые функции

Обозначение	**Тип аргументов**	**Тип значения**	**Примечание**
PLACE(*f*)	фигура	клетка	Координаты фигуры *f* в данной позиции
COLOR(*f*)	фигура	цвет	Цвет фигуры *f*
NEXT(*SQ*)	клетка	доска	Если SQ=(*i*, *j*), то функция описывает вертикали i-1 и i+1
ROUND(*SQ*)	клетка	доска	Если SQ=(*i*, *j*), то функция задает пояс клетки – квадрат стороной 3 клетки с центром (*i*,*j*)
AROUND(*SQ*)	цвет	доска	Если SQ=(*i*, *j*), то функция задает пояс клетки – квадрат стороной 5 клетки с центром (*i*,*j*)
E-COLOR(*C*)	цвет	цвет	Определение цвета фигур противника
MANS(*C*)	цвет	доска	Описание, фигур цвета *C*
PAWNS(*C*)	цвет	доска	Описание пешек цвета С
P-MUVES(*C*)	цвет	доска	Описание ходов пешек
P-HITS(*C*)	цвет	доска	Описание ударов пешек
M-MUVES(*C*)	цвет	доска	Описание ходов фигур
M-HITS(*C*)	цвет	доска	Описание ударов фигур
LINE(*SQ*,*L*)	клетка, линия	доска	Описание вертикали, горизонтали или диагонали с клеткой SQ
C-LINE(*SQ1*,*SQ2*,*L*)	клетка, клетка, линия	доска	Описание отрезка линии между SQ1 и SQ2
FORWRD(*SQ*,*C*)	клетка, цвет	доска	Описание, фигур цвета *C*
QUINING(*SQ*,*C*)	клетка, цвет	доска	Описание поля превращения пешки на *SQ* цвета *C*.
Q-FIELDS*(SQ,C)*	клетка, цвет	доска	Описание критических полей проходной пешки
DISP(*SQ1*,*SQ2*)	клетка, клетка	доска	Описание прямоугольника с вершинами в клетках *SQ1* и *SQ2*
OPP(*SQ1*,*SQ2*)	клетка, клетка	доска	Описание оппозиции по линии *L* клеток *SQ1*, *SQ2*
POPP(*SQ1*,*SQ2*)	клетка, клетка	доска	Описания конфигурации клеток, в которой через ход возможна оппозиция по линии *L*.

Предикатные символы равенства (=) и включения (⊂) языка L интерпретируются следующим образом: если B_1 и B_2 термы, имеющие значение типа доска, то:

(*i*) значение предиката $B_1 = B_2$ истинно, если значения термов B_1 и B_2 покомпонентно совпадают, и ложно, если хотя бы для одной компоненты $B_1(i, j) \neq B_2(i, j)$.

(*ii*) значение предиката $B_1 \subset B_2$ истинно, если для значений термов B_1 и B_2 выполняется покомпонентная импликация, т. е. $\forall\ i, j\ (B_1 \rightarrow B_2)$ и ложно, если хотя бы для одной компоненты $B_1(i, j) = 1\ \&\ B_2(i, j) = \varnothing$.

Унарные предикаты и их интерпретация приведены в табл. 2.

Формулу языка *L,* интерпретированную в структуре **A**, назовем д о п у с т и м о й , если она либо не содержит кванторов и унарных предикатов, либо находится в предваренной форме с ограниченными кванторами, а сами ограниченные кванторы имеют вид $\exists x\ p(x)$ либо $\forall x\ p(x)$, где p – один из одноместных предикатов, перечисленных в табл. 2.

Таблица 2

Унарные предикаты

Предикат	Область истинности
T-MANS	переменные типа фигура
T-COLOR	переменные типа цвет
T- SQUARE	переменные типа клетка
T- LINE	переменные типа линия
T- BOARD	переменные типа доска

4. Базовые функции, включенные в описанную выше структуру **A**„ получены в результате анализа дескрипторов [2]. При этом структура **A** позволяет получить допустимые формулы почти для всех из более 100 приведенных в [2] шахматных понятий (исключение составляют те понятия, которые сильно зависят от динамических характеристик позиции, в частности, количество ударов на клетку). Ниже в качестве примера приведены формульные представления десяти из них. Первые шесть формул определяют пешечные характеристики позиций, последние четыре использованы в [4] для описания стратегии в эндшпиле «король, ладья против короля».

Описание каждого из этих понятий включает:

1) Описание на естественном языке;
2) Список констант, используемых в формуле;
3) Формульное уточнение понятия.

1. Проходная пешка: пешка называется проходной, если впереди этой пешки по вертикали нет пешек противника, и ни одна из этих клеток не бьется пешками противника.

Константы: SQ – клетка пешки, C – цвет пешки, F – вертикаль.

Формула: ((NEXT(SQ) $\cup$ LINE(SQ, F)) $\cap$ FORWRD(SQ, C) $\cap$ PAWNS(E-COLOR(C))) = $\varnothing$).

2. Блокированная пешка: пешка называется блокированной, если перед ней стоит пешка противника.

Константы: SQ – клетка пешки, C – цвет пешки, F – вертикаль.

Формула: $\neg$ (LINE(SQ, F) $\cap$ ROUND(SQ) $\cap$ FORWRD(SQ, C) $\cap$ (E-COLOR(C) = $\varnothing$).

3. Изолированная пешка: пешка называется изолированной, если на соседних с ней вертикалях нет пешек того же цвета.

Константы: SQ – клетка пешки, C – цвет пешки,

Формула: $\neg$ (NEXT(SQ) $\cap$ PAWNS(C) = PLACE(SQ)).

4. Отсталая пешка: пешка называется отсталой, если на горизонталях с данной клеткой и ниже нет пешек того же цвета.

Константы: SQ – клетка пешки, C – цвет пешки, R – горизонталь.

Формула: ((LINE(SQ, R) $\cap$ PAWNS(C)) $\cup$ (PLACE(SQ) $\cap$ FORWRD(SQ, E-COLOR(C)))=PLACE(SQ).

5. Сдвоенная пешка: пешка называется сдвоенной, если на той же вертикали находится еще одна пешка своего цвета.

Константы: SQ – клетка пешки, C – цвет пешки, F – вертикаль.

Формула: $\neg$ (LINE(SQ, F) $\cap$ PAWNS(C) = PLACE(SQ)).

6. Фаланга пешек: пешка образует фалангу, с другой пешкой своего цвета, если последняя стоит на соседней с ней по горизонтали клетке.

Константы: SQ – клетка пешки, C – цвет пешки, R – горизонталь.

Формула: $\neg$ (LINE(SQ, R) $\cap$ ROWND(SQ) $\cap$ PAWNS(C) = $\varnothing$).

7. Отсечение ладьей: короли соперников находятся по разные стороны от вертикали или горизонтали, на которой находится ладья.

Константы: WK– белый король, BK– черный король, WR – белая ладья.

Формула: $\exists$L T–LINE(L)($\neg$(LINE(PLACE(WR), L) $\cap$ DISP(PLACE(WK), PLACE(BK)) = $\varnothing$ &

& $\neg$ (LINE(PLACE(WR), L) = LINE (PLACE(WK)).

8. Мат в эндшпиле «король, ладья против короля»: король черных находится под шахом и у него нет ходов.

Константы: WR – белая ладья, BK – черный король, B – черные.

Формула: $\exists$L T–LINE(L)(M-MUVES(B) = $\varnothing$ & LINE(PLACE(WR), L) = LINE(PLACE(BK), L)).

9. Оппозиция королей: короли находятся в оппозиции, если они стоят на одной линии и расстояние между ними минимально.

Константы: WK, BK – белый и черный короли.

Формула: $\exists$L T–LINE(L)(OPP(PLACE(WK), PLACE(BK), L) = PLACE(WK) $\cup$ PLACE(BK)).

10. Ладья атакована королем противника.

Константы: WR – ладья, B – черные.

Формула: $\neg$(PLACE (WR) $\cap$ M-HITS(B) = $\varnothing$).

3. ОПИСАНИЕ АЛГОРИТМА ФУНКЦИОНИРОВАНИЯ

Разработка компоненты DESC-VALUE проводилась в соответствии с принципами нисходящего структурного программирования, что позволило получить программу с иерархической модульной структурой. Ниже дается описание модулей первого и второго уровней. Модули нижних уровней выполняют технические функции вида сканирования текста, операций со стеком, вычислений функций и, поэтому детально не описываются. Приводимые описания работы модулей включают в себя описания входных и выходных параметров, словесное описание алгоритмов, тексты модулей на PL–подобном псевдоязыке. Используемый псевдоязык состоит из:

а) управляющих операторов DO-WHILE, END-DO и IF-THEN, ELSE-IF, ELSE, END-IF;

б) функциональных операторов, которые описываются на естественном языке и задают обращение к модулям нижних уровней.

2. Модуль DESC-VALUE реализован на языке PL /1 в виде процедуры–функции, входными параметрами которой являются: символьная строка DESC, задающая текст формулы дескриптора допустимого типа (см. 2.2) виде последовательности соответствующих логических символов; описание позиции POS, задающее список фигур текущей позиции.

При обращении к модулю DESC-VALUE в любом месте PL–программы возвращается битовое значение 1, если формула DESC – истинна в позиции POS и 0 в противном случае.

Так как каждая допустимая формула может содержась кванторную приставку, то матрица формулы должна проверяться для всех распределений связанных переменных формулы, определяемых кванторами. Для того, чтобы обеспечить простоту проверки матрицы для конкретного распределения, она переводится в постфиксную форму записи. Алгоритм работы модуля заключается в выделении приставки к матрице формулы, построении постфиксной формы и цикла, на каждом шаге которого вычисляется значение матрицы для текущего распределения переменных и порождается очередное распределение.

Ниже следует текст модуля на псевдокоде:

```
DESC-VALUE: PROCEDURE
    (DESC /*ПРОВЕРЯЕМАЯ ФОРМУЛА*/, POS /*ТЕКУЩАЯ ПОЗИЦИЯ*/)
```

```
    RETURNS (B1T(1) /* ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ,/);
    <ВЫДЕЛИТЬ ПРИСТАВКУ И МАТРИЦУ ФОРМУЛЫ DESC >
    <ПЕРЕВЕСТИ МАТРИЦУ В ПОСТФИКСНУЮ ФОРМУ > ;
    <ОПРЕДЕЛИТЬ НАЧАЛЬНЫЕ ЗНАЧЕНИЯ СВЯЗАННЫХ ПЕРЕМЕННЫХ >
    DO WHILE (< ФОРМУЛА НЕ ВЫЧИСЛЕНА>);
        <ВЫЧИСЛИТЬ ЗНАЧЕНИЕ МАТРИЦЫ ДЛЯ ДАННОГО РАСПРЕДЕЛЕНИЯ
    ПЕРЕМЕННЫХ);
        <ПОЛУЧИТЬ ОЧЕРЕДНОЕ РАСПРЕДЕЛЕНИЕ СВЯЗАННЫХ ПЕРЕМЕННЫХ);
    END–DO;
    RETURN (< ЗНАЧЕНИЕ ФОРМУЛЫ >);
END DESC-VALUE;
```

3. Модуль IN-TO-PST. Входными данными для модуля является текст матрицы формулы в инфиксной форме. Преобразование матрицы в постфиксную форму выполняется на основе следующей приоритетной функции *P*, определенной для символов операций и скобок.

Символ	Приоритет *P*
(,)	0
$\cup$ $\cap$	1
= $\subset$	2
$\vee$ &	3
¬	4

Исходный текст сканируется слева направо. Если очередная лексема является константой типа доска либо функцией, то она становится; очередной лексемой постфиксной формы; остальные символы поступают в стек и далее либо удаляются (если это скобки), либо перемещаются в результатную постфиксную форму в соответствии с их приоритетами.

Ниже приведено описание модуля IN-TO-PST на псевдокоде:

```
IN-TO-PST: PROCEDURE(INMAT/*ИНФИКСНАЯ ФОРМА*/, PSTMAT /*ПОСТФИКСНАЯ
ФОРМА*/)
    I = 1;
    DO WHILE (<INMAT НЕ ИСЧЕРПАНО>);
        LEX = <ОЧЕРЕДНАЯ ЛЕКСЕМА ИНФИКСНОЙ ФОРМЫ>;
        IF LEX = '(' THEN
            <ЗАСЛАТЬ В СТЕК СИМВОЛ '('>;
        ELSE-IF LEX = {<БАЗОВАЯ ФУНКЦИЯ> | <ПЕРЕМЕННАЯ ТИПА ДОСКА> |
                                    <КОНСТАНТА ТИПА ДОСКА>} THEN
            <ПЕРЕСЛАТЬ LEX В ФОРМУ PSTMAT>;
        ELSE-IF LEX = ') ' THEN
            DO WHILE (<ОЧЕРЕДНОЙ ЭЛЕМЕНТ СТЕКА> ≠ '(');
                    <ПЕРЕСЛАТЬ ОЧЕРЕДНОЙ ЭЛЕМЕНТ СТЕКА В ФОРМУ PSTMAT>;
            END-DO;
            <ИСКЛЮЧИТЬ '(' ИЗ СТЕКА>;
        ELSE IF LEX = {<СИМВОЛ ОПЕРАЦИИ> | <ЭЛЕМЕНТАРНЫЙ ПРЕДИКАТ> THEN
            DO WHILE (<СТЕК НЕ ПУСТ> & P(LEX) ≤ P(<ЭЛЕМЕНТА СТЕКА>));
                    <ПЕРЕСЛАТЬ ЭЛЕМЕНТ СТЕКА В ФОРМУ PSTMAT>;
            END– DO;
              <ЗАСЛАТЬ LEX В СТЕК>;
        ELSE <ПЕЧАТЬ СООБЩЕНИЯ ОБ ОШИБКЕ>;
```

```
    END-DO;
    DO-WHILE (<СТЕК НЕ ПУСТ>);
        <ПЕРЕСЛАТЬ ЭЛЕМЕНТ СТЕКА В ФОРМУ PSTMAT>;
    END-DO;
    END IN-TO-PST;
```

4. Модуль MAT-VALUE. Входными данными для модуля MAT-VALUE является текст матрицы формулы в постфиксной форме и описание текущей позиции. Модуль организован как подпрограмма-функция, выдающая значение вычисляемой формулы. Для вычисления значения матрицы используется стек промежуточных результатов, в который заносятся либо указатели на вычисленные ранее доски (операнды для операций ∪, ∩ = и ⊂), либо значения атомных формул (операнды для операций ¬, ∨ и &). Исходный текст сканируется слева направо и если очередная лексема является операндом операций, то в стек заносится указатель на соответствующее значение, если же очередная лексема – операция, то эта операция применяется для верхних элементов стека, которые далее заменяются указателем на полученное значение. Результат вычисления матрицы формулы получается в вершине стека. Операции со стеком выполняются модулями нижних уровней.

Ниже приводится описание алгоритма работы модуля MAT-VALUE на псевдокоде:

```
MAT –  VALUE: PROCEDURE(
    PSTMAT /*ПОСТФИКСНАЯ ФОРМА*/
    POS /*ОПИСАНИЕ ПОЗИЦИИ*/)
    RETURNS (BIT (1) /*ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ*/;
    I = 1;
    DO I = 1 BY 1 WHILE (<PSTMAT НЕ ИСЧЕРПАНА>);
        LEX = <I–ТАЯ ЛЕКСЕМА PSTMAT>;
        IF LEX = (БАЗОВАЯ ФУНКЦИЯ> THEN
            <ВЫЧИСЛИТЬ ФУНКЦИЮ И ЗАСЛАТЬ ЗНАЧЕНИЕ В СТЕК>;
        ELSE-IF LEX = <КОНСТАНТА ТИПА ДОСКА> THEN
            <ПЕРЕСЛАТЬ LEX В СТЕК>;
        ELSE-IF LEX = '¬ ' THEN
            <ВЫПОЛНИТЬ ОПЕРАЦИЮ '¬' В СТЕКЕ>;
        ELSE-IF LEX ∈ {∪, ∩, =, ⊂, ∨, &} THEN
            <ВЫПОЛНИТЬ ДВУМЕСТНУЮ ОПЕРАЦИЮ LEX>;
        ELSE <ВЫДАТЬ СООБЩЕНИЕ ОБ ОШИБКЕ>;
        END-IF;
    END-DO;
END MAT-VALUE;
```

Բ. Կ. ԿԱՐԱՊԵՏՅԱՆ

ՇԱԽՄԱՏԱՅԻՆ ԳԱՂԱՓԱՐՆԵՐԻ ՆԵՐԿԱՅԱՑՄԱՆ ՄԻԱԼԱՐՈՒՄ

Դիտարկված կ վերջավոր քանորակներով առաջին կարգի ստորոգյալների հաշվի բանաձևերով տրված շախմատային գաղափարների նկարագրիչների իրացման համար նախատեսված АКА–у2 *կապոցի ծրագրային բաղադրամասը: Առաջարկված են նկարագրիչների հիմնական տարրերի համար տվյալների բազմություններ, որոնք թույլ են տալիս համակարգել և միալարել ինչպես նկարագրիչների ներկայացումները, այնպես էլ դրանց առանձին բաղադրամասերի մշակման ընթացակարգերը: Տրվում է հետազոտված շախմատային գաղափարների բանաձևերի ցուցակ:*

ЛИТЕРАТУРА

1. *Погосян Э.* М. Адаптация комбинаторных алгоритмов. Ереван, Изд– во АН АрмССР,1983.

2. *Погосян Э. М., Амбарцумян М. А., Аджабян Н. А., Арутюнян Ю. Г., Джндоян Л. О., Карапетян Б. К.* Пакет прикладных программ для исследований методов адаптации комбинаторных алгоритмов управления (АКА 1,2). Отчеты ВЦ АН АрмССР за 1974–1980 гг.

3. *Погосян Э. М.,Калантарян А. Б., Карапетян Б. К.* Требования к программному обеспечению исследований методов адаптации комбинаторных алгоритмов управления, см. настоящий сборник.

4. *Карапетян Б. К* Язык высокого уровня для описания стратегий, см. настоящий сборник.

5. *Шенфилд Дж.* Математическая логика, М., Мир, 1975.

Б. К. Карапетян, ЯЗЫК ВЫСОКОГО УРОВНЯ ДЛЯ ОПИСАНИЯ СТРАТЕГИИ В ИГРАХ,

Mathematical Problems of Computer Sciences, 1986, pp. 167-183.

Введение

Понятие стратегии для позиционных игр определяется как некоторое отображение (вообще говоря не однозначное), множества позиций в ходы. В классической теории игр в основном исследуются условия существования оптимальных стратегий без рассмотрения вопросов их практического использования, в частности, вопросов представления, порождения и хранения стратегий. В то же время именно эти и аналогичные вопросы составляют предмет исследования в программировании игр, где каждая программа, по существу, является конкретной реализацией некоторой стратегии.

Большинство существующих игровых программ используют при выборе хода минимаксную процедуру перебора дерева игры с конкретной оценочной функцией, которая по существу является некоторым квазианалитическим описанием используемой стратегии. При таком подходе возникают трудности модификации стратегий, связанные с большими размерами дерева игры, с невозможностью точного описания структуры позиции посредством оценочной функции.

В последнее время появился ряд работ, в которых исследуется возможность представления и использования стратегий на основе систем закономерностей [1] (продукций) [7, 8] типа <класс позиций ⇒ рекомендуемые действия>, характерных для шахматиста-мастера и существенно сокращающих порождаемые деревья перебора (в частности для эндшпилей). Так в [7] рассмотрена возможность применения системы AL/1 для шахматных эндшпилей. Основу систем AL/1 составляют «таблицы советов» (advice tables), которые состоят из правил следующего типа:

ЕСЛИ <предусловие> ТО <список рекомендаций>.

Для шахматных эндшпилей <предусловие> выделяет некоторый класс позиций, а <список рекомендаций> описывает возможные ходы в этом классе. Использование таблиц советов для выбора хода выполняется простой процедурой: найти первое правило, предусловие которого истинно для текущей позиции, и выполнить соответствующие ходы. Такая управляющая структура оказалась недостаточной для сложных эндшпилей и в [8] предложена система AL3, в рамках которой можно описывать планы и цели, позволяющие управлять процессом выбора очередного правила.

Более универсальный подход предложен в [9], где описывается программа поиска комбинаций, приводящих к выигрышу материала. Планы, порождаемые этой программой, задают последовательность действий, которые необходимо для достижения конкретной цели. Реализация каждого действия включает в себя рассмотрение ответов противника, порождение возможных подпланов, определение так называемых опасных ходов, которые необходимо опровергнуть, и построение планов опровержения таких ходов. При построении стратегий программа производит относительно небольшой перебор неперспективных вариантов. Однако ограничение списка возможных целей и использование для задания искомой комбинации материального выигрыша ограничивают область применения программы именно комбинациями, усложняют ее модификацию для построения более общих выигрышных стратегий.

Шахматная программа PARADISE, использующая для управления поиском хода базу знаний, описана в [10]. Построение стратегии игры основано на статическом анализе позиции, результатом которого является дерево планов, используемое в дальнейшем как основа искомой стратегии. Каждый узел дерева планов либо определяет конкретный ход, либо описывает цель, для достижения которой необходимо обращение к базе знаний. База знаний содержит возможные цели и действия по их реализации в обобщенной форме — так называемые «источники знаний», которые конкретизируются при использовании в текущей позиции. Если построенное дерево планов не реализуется программой, то анализ выполненных действий с использованием много-ступенчатой оценочной функции выделяет некоторую позицию, в которой заново производится построение планов. В указанных выше работах [7-10] не рассмотрена возможность непосред-ственного управления процедурой реализации планов: последовательность шагов реализации стратегий либо зафиксирована [7, 8], либо определяется единообразной процедурой [9, 10].

В отличие от них в настоящей работе описываются средства использования при построении стратегий не только декларативных знаний типа закономерностей (продукций), но и процедуральных знаний, описывающих определенные комбинации выполняемых действий. Стратегия задастся на алгоритмическом языке высокого уровня (SDL) в виде конкретной процедуры, функциональными операторами которой являются закономерности. Реализация заданной стратегии в некоторой позиции осуществляется многоуровневой компиляцией: интерпретацией SDL-описанной стратегии в закономерности и трансляцией каждой закономерности в шахматные ходы.

1. ПРЕДСТАВЛЕНИЕ И ИСПОЛЬЗОВАНИЕ СТРАТЕГИИ В ПОЗИЦИОННЫХ ИГРАХ

Позиционная игра двух противников определяется следующими компонентами:

а) множеством позиций $\mathbf{P}$,

б) множеством ходов M, которые задают переходы от одной позиции к другой;

в) множеством начальных позиций $\mathbf{P}_0 \subset \mathbf{P}$;

г) множеством заключительных позиций $\mathbf{P}_f \subset \mathbf{P}$, для каждой из которых заданы выигрыши игроков.

Игру такого типа можно описать деревом $G = (V, E)$, где множество вершин V соответствует множеству позиций игры $\mathbf{P}$, а множество ребер E соответствует ходам. Если вершинам v_1, $v_2 \in V$ соответствуют позиции p_1, $p_2 \in \mathbf{P}$, то ребро $(v_1, v_2) \in E$ тогда и только тогда, когда существует ход $\tau \in M$, переводящий p_1 в p_2. На концевых вершинах дерева указаны выигрыши каждого игрока. Мы будем рассматривать, конечные игры двух противников с полной информацией и нулевой суммой. Для дерева G выполняются следующие условия:

а) в дереве С нет бесконечных ветвей;

б) в каждой вершине дерева указано, какому из игроков принадлежит право хода;

в) выигрыш одного из игроков в заключительной позиции равен проигрышу другого.

Игроков условно мы будем называть игрок 1 и игрок 2,

Каждая ветвь дерева задает некоторую партию, результат которой определяется соответствующей заключительной позицией. Мы будем рассматривать игры, в которых результат партии может быть трех типов: выигрыш, ничья или проигрыш игрока, сделавшего первый ход в партии.

Стратегией игрока i называется функция S^i: $V^i \rightarrow E$, где $V^i \subset V$ множество позиций с ходом игрока i. Здесь и далее $i, j \in \{1, 2\}$, $i \neq j$. Ход τ игрока i в позиции p определен стратегией S^i, если в вершине $v \in V^i$, соответствующей p, ребро $s^i(v)$ соответствует ходу τ.

Если задана позиция p и стратегии S^1 и S^2 игроков 1 и 2, то однозначно определяется партия $\pi(S^1, S^2, p) = \langle p, p_1, p_2, ..., p_k \rangle$, в которой позиции $p_1, p_2, ..., p_k$ получаются ходами, определенными стратегиями S^1 и S^2.

Пусть p — позиция с ходом игрока i. Стратегия S^i называется выигрышной, ничейной или проигрышной для игрока i, если для любой стратегии игрока j $(i \neq j)$ партия $\pi(S^i, S^j, p)$ является соответственно выигрышной, ничейной или проигрышной.

Выигрышность стратегии является достаточным обоснованием для выбора хода. Если для некоторого множества позиций $\mathbf{P}_S$ игрок i имеет выигрышные стратегии, то в позициях, расположенных выше на дереве игры он будет, по возможности, выбирать ходы, приводящие к позициям из $\mathbf{P}_S$.

Во всех нетривиальных играх деревья игры имеют необозримые размеры, что затрудняет выбор хода. Использование стратегии с известными свойствами уменьшает объем рассматриваемой информации, позволяя оперировать при анализе позиции не отдельными ходами, а поддеревьями и классами позиций. Ниже проблема представления и использования стратегий рассматривается на примере шахмат.

Одним из способов представления стратегий является ее аналитическое задание, т. е. определение хода по некоторой формуле, учитывающей особенности рассматриваемой позиции. Такой способ задания стратегии используется во всех современных игровых программах, использующих оценочную функцию.

Оценочная функция является либо линейной комбинацией некоторого набора предикатов, включенных с весовыми коэффициентами, либо суперпозицией линейных и логических функций. Использование оценочной функции предполагает ее вычисление для всех позиций, возникающих из данной, после чего выбирается ход, приводящий в позицию с наилучшей оценкой (максимальной для белых, минимальной для черных). Предикаты, используемые в оценочной функции, выделяют различные существенные признаки позиций, такие как ценность фигур и их расположение, слабые и сильные поля и удары на них и т. д.

Сложность использования шахматных понятий в оценочной функции заключается в том, что оценочная функция должна быть быстровычисляемой и компактной, а так как при каждом обращении к оценочной функции должны вычисляться все предикаты, то каждый из них должен быть относительно простым. Это вынуждает заменить признаки более простыми, не совпадающими с их формальными аналогами. Кроме этого в шахматной литературе одно и то же понятие используется по-разному в зависимости от стадии игры, от текущих планов и целей. Все эти аспекты использования должны моделироваться в оценочной функции весами предикатов, что трудно осуществимо.

Все это приводит к тому, что ход, выделенный оценочной функцией, не всегда является хорошим в данной позиции. Для обеспечения точности оценки ее использование дополняется переборной минимаксной процедурой, которая строит дерево игры на определенную глубину, вычисляет оценку заключительной позиции и выполняет подъем оценки по уровням дерева по принципу минимакса. Точность полученной оценки зависит от глубины дерева перебора, которую при ограниченных ресурсах можно увеличить отсечением неперспективных ветвей дерева, поэтому сила игры переборных программ существенно зависит от эвристик, применяемых при таких отсечениях.

Влияние различных предикатов оценочной функции на выбор хода переборной процедурой усредняется, поэтому очень сложно определить необходимость изменения весов признаков, их удаления или добавления.

Вопросы накопления знаний, синтеза новых стратегий, использования результатов предыдущих партий посредством оценочной функции также сводятся к изменению весов предикатов и самих предикатов и остаются неразрешенными (такие решения как «служба лучших ходов», «служба лучших ответов» [1] реализованы вне оценочной функции).

Другой способ представления стратегии, требующий минимальных временных ресурсов при использовании, является задание стратегии деревом (p-стратегии) [2]. Если задана позиция p с ходом игрока i, то стратегию Sk можно описать поддеревом Gk дерева G, которое строится следующим образом:

а) Корнем поддерева Gk является вершина v, соответствующая позиции p;

б) Если некоторая вершина v1 с ходом игрока i включена в Gk, то в Gk включается также ребро Si(v1) = (v1, v2) и вершина v2 с ходом игрока j;

в) Если некоторая вершина v1 с ходом игрока j включена в Gk, то в Gk включаются также все ребра (v1, v21), (v1, v22), …, (v1, v2k), которые исходят из вершины v1 в дереве игры G и вершины v21, v22, ..., v2k с ходом игрока i.

Другими словами, из вершин дерева Gk с ходом игрока i исходит ровно одно ребро, соответствующее ходу, определенному стратегией Sk, а из вершины с ходом игрока j исходят ребра, соответствующие всем ходам из этой позиции. Для использования поддерева Gk при выборе хода, достаточно иметь указатель на текущую позицию, который определяет ход игрока i в позициях с его ходом и при каждом ответе противника передвигается на соответствующую вершину. Существенный недостаток такого представления — это размерность соответствующих деревьев. Даже для такого простого эндшпиля, как «Король, ладья против короля», дерево стратегии для большинства позиции имеет порядка 10!0 вершин. Более сложные стратегии приводят к экспоненциальному росту размеров деревьев, и проблемы порождения, хранения и модификации стратегий при таком представлении становятся практически неразрешимыми.

Уменьшения требуемых ресурсов памяти можно достигнуть, используя для описания стратегий понятия, более информативные, чем ход. Такие понятия должны задавать преобразования классов позиций, обладающих некоторыми общими свойствами, в другие классы. Формализация таких преобразований определена в [2] как <P, F`>-стратегии и их слияния, где P — исходная позиция и F — описание класса позиций, достижимых <P, F`>-стратегией. Рассмотрение шахматной литературы показывает широкое использование преобразований этого типа как при обучении игре, так и при анализе и разборе партии. Стратегии при этом описываются в таких терминах, как «оттеснение короля», «фланговая атака», «игра на противоположном фланге», «использование слабых полей» и т. д. При этом в большинстве случаев описание стратегии игры в конкретных классах позиций задается алгоритмически. Примером может служить приводимое ниже описание стратегии игры в эндшпиле «Король, ладья против короля»:

А) Отсечь короля противника ладьей;

Б) Если короли находятся в оппозиции, и ладья может дать шах с безопасного поля, то оттеснить короля к краю доски шахом; если мата нет, то перейти к В), иначе цель достигнута;

В) Если оппозиция возможна после хода противника, то сделать выжидательный ход ладьей и перейти к Б);

Г) Если король противника отступил от линии отсечения, то оттеснить короля ладьей, перейти к Б);

Д) Если короли находятся дальше, чем на 2 хода друг от друга, то подойти своим королем к королю противника, не переходя линию отсечения; перейти к Б);

Е) Во всех случаях, если король противника атакует ладью, отступить ладьей по линии отсечения на противоположный край доски.

Аналогичные описания можно получить для стратегий игры не только во многих других форсированных окончаниях, но и для представления стандартных комбинационных приемов игры (завлечение фигуры под связку, атака связанной фигуры, «мельница», «спертый мат» и т. д.). Рассмотрение таких примеров показывает, что в шахматной теории и практике стратегии описываются в терминах более элементарных действий, способ реализации которых ходами известен, либо описан ранее. Само описание стратегии является алгоритмом, посредством которого можно получить последовательность элементарных действий, необходимых для реализации стратегии.

Алгоритмическое представление стратегий по существу является процедурой на языке высокого уровня, задающей последовательность преобразований позиций, в терминах элементарных действий. Использование такой процедуры возможно с помощью транслятора, который обеспечивает преобразование <позиция> → <ход>. В данной работе описывается язык описания стратегий *SDL* (Strategy Description Language), предназначенный для процедурного представления стратегий и *SDL*-интерпретатор, обеспечивающий реализацию SDL-стратегии для заданной позиции.

2. ОПИСАНИЕ ЯЗЫКА

Язык *SDL* позволяет задавать *T*-слияние стратегий [2], используя фиксированный набор элементарных действий (здесь *T* — это множество позиций, в которых применима данная стратегия).

Входными данными для *SDL*-стратегий (т. е. стратегии, заданной на языке *SDL)* является исходная позиция с ходом белых. *SDL*-стратегия по существу является процедурой, которая задает порядок применения элементарных действий в исходной позиции *p,* необходимый для построения дерева *p*-стратегии. Элементарные действия составляют функциональные операторы языка *SDL,* т. е. операторы, задающие преобразование позиций. Порядок применения элементарных действий задается управляющими операторами языка.

2.1. Представление позиции и классов позиций

Позиция задается как набор фигур, белых и черных, каждая из которых обозначается однобуквенным символом: *K* — король, *Q* — ферзь, *R* — ладья, *B* — слон, *N* — конь и *P* — пешка. Описание позиций задается посредством досок. Каждая доска — это матрица размерности 8x8 нулей и единиц. Для досок определены операции объединения и пересечения, которым соответствует покомпонентная дизъюнкция и конъюнкция соответственно, т. е. если $D_1 = \{d^1_{ij}\}$, $D_2 = \{d^2_{ij}\}$] и $D_3 = \{d^3_{ij}\}$ — доски, то для объединения $D_1 \cup D_2 = D_3$ имеем $d^3_{ij} = d^1_{ij} \vee d^2_{ij}$, а для пересечения $D_1 \cap D_2 = D_3$ — $d^3_{ij} = d^1_{ij}$ & d^2_{ij}. Выражение, составленное из объединения и пересечения досок, наливается первичным выражением. Два первичных выражения, соединенных символом отношения, составляют элементарный дескриптор. В SDL используются отношения =, ≠, ⊂. Элементарный дескриптор — это предикат, равный единице, если отношение выполняется, и нулю в противном случае. Отношение равенства (неравенства) выполняется, если левая и правая части покомпонентно совпадают (не совпадают). Отношение включения выполняется, если выполняется компонентная импликация. Выражение, составленное конъюнкцией и дизъюнкцией элементарных дескрипторов, задает дескриптор класса позиций, т. е. предикат, равный единице для позиций описываемого класса и нулю для остальных позиций.

Элементарные дескрипторы используются для описания шахматных понятий, таких как открытая вертикаль, проходная пешка, оппозиция королей и т. д. Исследование дескрипторов, описанных в [3], позволило выделить набор досок, посредством которых можно описать соответствующие классы позиций. В язык *SDL* включены доски трех типов:

1) геометрические доски, описывающие такие понятия, как вертикаль, горизонталь, диагональ, квадрат проходной пешки, центр доски, королевский фланг, и т. д.

2) фигурные доски, описывающие понятия, связанные с расположением одной или группы фигур, например, доска положения фигуры (единице в, клетке, занятой данной фигурой, и нули

в остальных клетках); доска оппозиции королей, доска критических полей проходной пешки и т. д.

3) доски возможностей, описывающие понятия, связанные с ходами фигур, т. е. в некотором смысле динамические. Сюда относятся доски задающие ходы фигур, описывающие связку, вскрытый шах, «вилку» и т. д.

Каждая доска в *SDL* задается уникальным именем и списком параметров, заключенных в круглые скобки. В качестве параметров досок выступают обозначения фигур. Например, выражение *SQ(W.R)* означает доску расположения белой ладьи; *R(W.R)* — это обозначение горизонтали, на которой стоит белая ладья (т. Е. единицы по описываемой горизонтали и нули на остальных клетках).

2.2. Описание элементарных действий

Элементарное действие задается следующей синтаксической конструкцией:

ACTION (<имя действия>, <исходный класс>, <конечный класс).

Здесь ACTION — ключевое слово, выделяющее данное действие;

<имя действия> — идентификатор произвольной длины, обозначающий одно из действий, зафиксированных в языке;

<исходный класс>—дескриптор, задающий условие применимости данного действия, т. е. множество позиций, в которых действие выполнимо;

<конечный класс> — дескриптор, описывающий результат применения данного действия.

Дескрипторы, задающие исходные и конечные классы позиций, вычисляются для той позиции, в которой применяется данная закономерность. Если для некоторой доски, входящей в дескриптор конечного класса, необходимо указать, что она вычисляется после применения данной закономерности, то перед именем этой доски необходимо поставить букву *N* и точку. Например, исходный класс позиций, в которых короли находятся в оппозиции, можно задать дескриптором

$SQ(W.K) \cup SQ(B.K) = OPP(W.K, B.K),$

где $SQ(W.K)$, $SQ(B.K)$ —доски положения белого и черного короля, а $OPP(W.K, B.K)$ —доска оппозиции. Тогда конечный класс, после шаха ладьей, можно задать дескриптором $SQ(W.K) \cup SQ(B.K) = OPP(W.K, B.K) = L(B. K),$ где L (<фигура >) — доска линии указанной фигуры.

2.3. Управляющие операторы

Управляющие операторы SDL определяют порядок выполнения закономерностей в зависимости от типа достигнутой позиции. В SDL таких операторов два: оператор цикла типа *DO-WHILE* и оператор условного выбора *1F.* Как известно [4, 5] операторы цикла и условного выбора достаточны для записи любой процедуры. Для языка SDL можно показать, что если некоторая стратегия задана как правильная блок- схема [4], функциональные операторы которой выражаются элементарными действиями, то ту же стратегию можно записать с помощью операторов цикла и условного выбора, используя те же, что и в блок-схеме, предикаты и соответствующие закономерности. Выбор всего двух управляющих операторов продиктован стремлением к упрощению́ транслятора SDL. Дальнейшее расширение языка предполагает добавление других удобных структур управления.

Оператор *DO-WHILE* состоит из заголовка цикла, тела цикла и конца цикла. Заголовок цикла имеет вид *DO-WHILE*(<дескриптор>), где <дескриптор> задает класс позиций, для которых выполняется, тело данного цикла. Тело цикла составляют произвольные операторы *SDL.* Конец цикла задается в виде *END-DO.*

Если при выполнении стратегии встретился оператор цикла, то дальнейшее выполнение происходит следующим образом:

1. Для полученной на данном этапе позиции вычисляется дескриптор заголовка цикла. Если он принимает значение 0, то выполняемым, оператором выбирается оператор, следующий за

концом данного цикла. В противном случае выполнение стратегии продолжается с первого оператора цикла.

2. Если при выполнении тела цикла достигнут конец цикла, то выполняются действия, описанные в 1 для полученной позиции.

Иначе говоря, оператор *DO-WHILE* является оператором цикла с предусловием, т. е. с условием, проверяемым перед первым и каждым следующим выполнением тела цикла.

Синтаксис условного оператора задается в следующей форме:

IF <дескриптор> THEN <THEN — операторы >

{ELSE_IF <дескриптор > THEN <THEN — операторы >}

[ELSE <ELSE — операторы >]

END - IF

Здесь квадратные скобки выделяют необязательную часть оператора, а фигурные скобки — часть, которая может отсутствовать или быть повторена несколько раз; дескрипторы описывают классы позиций, в которых должны выполняться соответствующие операторы; <THEN-операторы> и <ELSE-операторы> — последовательности любых операторов, не содержащие условного оператора. Необходимость вложенных условных операторов исключена, так как вложенность определена синтаксисом условного оператора.

Условный оператор выполняется следующим образом:

1. Для текущей позиции вычисляется дескриптор, стоящий после ключевого слова *IF.* Если он равен единице, то выполняются *THEN*-операторы, после чего рассматривается оператор, следующий за ключевым словом *END-IF.* Если дескриптор условия равен нулю, то *THEN*-операторы пропускаются и рассматривается следующая лексема.

2. Если очередная лексема *ELSE-IF,* то оператор выполняется так *же,* как описано в 1.

3. Если очередная лексема *ELSE,* то выполняются *ELSE*-операторы, после чего выбирается оператор, следующий за ключевым словом *END-IF.*

2.4. Операторы STRAT и END-STRAT

Оператор *STRAT* задает имя данной стратегии и ее спецификацию, т. е. описание входов и результатов. Синтаксис оператора *STRAT* следующий:

<имя>: STRAT (IN-CLASS:

FIGURES-W: <список фигур>;

FIGURES-B: <список фигур>;

DESCR: <описание класса >;

OUT-CLASS: <описание класса>).

Здесь: <имя> — произвольный идентификатор; <список фигур> — перечисление фигур соответствующего цвета; <описание класса> — дескриптор, задающий исходный и конечный классы позиций. Дескрипторы, задающие описания классов, подчиняются правилам, указанным при описании действий (2.2).

Выполнение оператора *STRAT* заключается в проверке выполнимости входных спецификаций для заданной позиции. Если эта проверка успешна, т. е. совпадает фигурный состав и выполняется дескриптор, задающий исходный класс, то выбирается первый оператор, следующий за оператором *STRAT,* в противном случае стратегия не выполняется.

Оператор конца стратегии имеет следующую форму: *END-STRAT* <имя> и служит для обозначения конца текста процедуры.

3. КОМПИЛЯЦИЯ SDL-СТРАТЕГИИ

SDL-стратегия, так же как и процедуры любого языка высокого уровня, для использования на ЭВМ требует перевода на более низкий уровень представления, а именно на уровень, доступный данной вычислительной систему. Существуют две возможности компиляции программ с языка *L1* на язык L2: трансляция и интерпретация. Трансляция программы, заданной на *L1* заключается в последовательной замене ее операторов эквивалентными последовательностями команд *L2* и получении в результате новой процедуры, заданной на языке *L2*, которая и будет исполняться на ЭВМ.

При интерпретации каждый оператор процедуры на *L1* анализируется и непосредственно реализуется эквивалентной последовательностью операторов языка *L2*. Каждый, способ компиляции имеет свои преимущества и недостатки. Трансляция требует больших временных затрат для построения результатной процедуры, однако, выполнение полученной процедуры обычно требует меньше времени, чем при интерпретации исходной процедуры. При трансляции неизбежны затраты памяти для хранения полученной программы, при интерпретации память выделяется только для компиляции отдельного оператора интерпретируемой процедуры. Многоуровневая организация процедуры компиляции [6] позволяет более гибко использовать различные режимы компиляции поэтапным переходом с уровня на уровень.

3.1. Уровни представления SDL-стратегии в компиляторе

Многоуровневая организация компиляции характеризуется несколькими уровнями представления стратегий и процедурами компиляции с уровня на уровень. На каждом уровне исходная стратегия описывается на некотором языке, на верхнем уровне — исходный язык представления стратегии, на самом нижнем уровне — язык, доступный вычислительной системе. Для компиляции *SDL*-стратегии наивысший уровень системы —это язык *SDL.* При исполнении *SDL*-стратегии для заданной позиции последовательно выбирается очередная закономерность, которая должна быть реализована в ходах. Исходя из этого можно выбрать второй уровень описания стратегии — уровень закономерности и следующий уровень — описание стратегии в ходах. Наконец низший уровень системы, доступный вычислительной системе, задается алгоритмическим языком *PL/1.*

На уровне закономерностей *SDL*-стратегию можно представить как дерево исполнения, так называемое *E*-дерево [4], ребра которого соответствуют закономерностям и пути которого изображают все возможные последовательности реализации закономерностей без возвратов назад. При выборе хода на основе *SDL*-стратегии для заданной позиции реализуется конкретная ветвь *E*-дерева, так что рассмотрение оставшихся ветвей дерева становится излишним. Исходя из этого, для компиляции стратегии с уровня *SDL* на уровень закономерностей выбран режим интерпретации.

Реализация закономерности для данной позиции в ходах приводит к *p*-стратегии (см. 1), для построения которой необходимо рассмотрение всех ответов черных в каждой возникающей позиции. По существу для исполнения закономерности необходимо сначала получить полное ее описание в ходах, т. е. необходим трансляторный режим компиляции.

Реализация ходов *p*-стратегии на языке *PL/1,* так же, как и реализация *E*-дерева закономерности, естественным образом осуществляется интерпретатором. Описание уровней представления и соответствующих компиляторов приведено на рис. 1.

SDL —представление		**Уровень *4***
	интерпретация	
E—дерево закономерностей		**Уровень *3***
	трансляция	
p—стратегия		**Уровень *2***
	интерпретация	
Операторы		**Уровень *1***

Рис. 1, Многоуровневая организация процедуры трансляции SDL-стратегии на машинный уровень.

3.2. Функционирование SDL-компилятора

Алгоритм работы *SDL*-компилятора приведен на рис. 2. и 3. Для описания алгоритмов мы используем PL-подобный псевдоязык [5], в котором управляющие конструкции имеют обычный смысл, а выражения, заключенные в угловые скобки, соответствуют некоторым компонентам или модулям. На рис. 2 выделены основные компоненты компилятора, а на рис. 3 каждая компонента детализована до уровня модуля.

Первая компонента компилятора, инициализация, обеспечивает ввод исходной позиции и текста *SDL*-стратегии, проверяет применимость заданной стратегии в этой позиции и если позиция принадлежит описанию <IN-CLASS> исполняемой стратегии, то подготавливаются, необходимые переменные и массивы для последующих модулей.

Если в исходной позиции рассматриваемая стратегия не применима, то компилятор печатает соответствующее сообщение и завершает свою работу. В противном случае работает цикл *DO-WHILE,* при каждой итерации которого выполняется очередное действие. Компонента «Выделение очередного оператора ACTION», работает в режиме интерпретации и обеспечивает переход с уровня 4 на уровень 3 (рис. 1).

Реализация выделенного действия, (т. е. выполнение оператора ACTION), происходит в модуле «Построение соответствующей *p*-стратегии. Этот модуль работает в режиме трансляции и строит дерево *p*-стратегии, описывающее в ходах выделенное действие (т. е. на уровне 2, рис. 1).

Полученное дерево используется в следующем цикле *DO-WHILE* для реализации заданной стратегии средствами языка *PL/1.* Модуль «Выполнить очередной ход p-стратегии» интерпретирует выполнение очередного хода в текущей позиции. Модуль «Получить ответ противника» вводит ответ противника, выполняет его и находит вершину *p*-стратегии, соответствующую ответу на введенный ход. Работа цикла *DO-WHILE* заканчивается при достижении концевой вершины *p*-стратегии, т. е. после реализации очередного действия.

После реализации каждого действия повторяется выполнение внешнего цикла, до тех пор, пока не будет достигнут оператор END- STRAT.

```
SDL-COMPILER: PROCEDURE OPTIONS (MAIN);
          <Инициализация компилятора>;
   IF <инициализация не успешна> THEN <Выдача сообщения о неприменимости>;
ELSE DO;
DO WHILE (<не достигнут конец стратегии>);
                 <Выделение очередного оператора ACTION>;
                 <Построение соответствующей p-стратегии>;
                 DO WHILE (<p—стратегия не исполнена>);
                        <Выполнить очередной ход p-стратегии>; <Получить ответ противника>;
END; /*DO*/
END; /*DO*/
END; /*IF*/
END SDL— COMPILER;
```

Рис. 2. Описание SDL — компилятора. Основные компоненты.

```
SDL-COMPILER: PROCEDURE OPTIONS (MAIN);
/•Инициализация компилятора*/
<Ввод позиции POSITION>;
<Ввод текста стратегии >;
<Выделение класса применимости>;
IF < стратегия не применима > THEN
        <выдать сообщение о неприменимости >;
ELSE DO;
        <Подготовка таблиц компилятора >;
         <Выделить очередную лексему стратегии TEMP>
         /*Инициализация завершена успешно*/;
DO WHILE (TEMP ≠ 'END — STRAT');
/*Выделение очередного оператора ACTION*/
DO WHILE (TEMP ≠ 'ACTION' & TEMP´ ≠ 'END — STRAT');
IF TEMP = 'DO' | TEMP = 'END-DO' THEN
                              <Обработка оператора цикла>;
ELSE IF TEMP = 'IF' | TEMP = 'ELSE-IF' | TEMP = 'ELSE'´ THEN
                              <Обработка условного оператора>;
ELSE <Выдача сообщения по ошибке>;
END; /*DO*/
IF TEMP = 'ACTION' THEN DO;
                    <Выделить имя элементарного действия>;
<Вызов модуля транслятора для построения p-стратегии>;
NODE = < Корень полученного дерева >;
DO WHILE (<NODE — не концевая вершина p — стратегии>);
<Выполнить ход NODE в текущей позиции>;
CORRECT = '0'B;
DO WHILE (not CORRECT);
                              <Получить ответ черных>;
IF <ход правильный> THEN DO;
                                        <Выполнить ход в текущей позиции>;
                                        <Найти ход в дереве>;
NODE = <текущая вершина p-стратегии>;
CORRECT = '1'B;
END; /'THEN*/
ELSE <Выдача сообщения об ошибке>;
END; /*DO*/
END; /*DO*/
             <Выделить очередную лексему стратегии TEMP>
END; /*THEN*/
ELSE /*либо достигнут конец стратегии, либо ошибка в стратегии*/;
```

TEMP = 'END - STRAT';

END; /*DO*/

END /*ELSE*/

END SDL —COMPILER;

Рис. 3. Алгоритм *SDL*-компилятора. Основные модули.

3.3. Пример SDL-стратегии

Приведем описание *SDL*-стратегии для эндшпиля «Король, ладья против короля», реализующее стратегию, рассмотренную в [1].

SDL-стратегия, названная К-R-VS-К, описывает стратегию белых в указанном эндшпиле, позволяющую оттеснить короля черных к крайней вертикали и объявить ему мат.

K-R-VS-K: STRAT

(IN-CLASS:

FIGURES - W: K, R;

B: K;

DESGR - ANY;

OUT-CLASS: MV (*B. K.*) = Ø);

ACTION (LINE, ANY, N. L (*W.* R) ⊂ DISP (*W. K, B. K) &* N. L(W. R) ≠ N. L(W. K))

DO WHILE (MV (B. K) ≠ Ø);

IF (OPP (*W. K, B. K) &* SQ (R) ∩ MV *(B. K*) = Ø) | L (R) ∩ MV *(B. K*) = Ø THEN

ACTION (NEWLINE, ANY, (N. L (R) = L (B. K) |

N. L (R) ⊂ DISP (*W. K, B. K)* & N. SQ(R) ∩ MV *(B. K*) = Ø

ELSE-IF (POPP (*W. K, B. K))* THEN

ACTION (DUMMY, ANY, N. L (R) = L (R) & SQ (R) ∩ AL(R, *B.* K) = Ø)

ELSE-IF (SQ (R) ∩ MV *(B. K*) ≠ Ø) THEN

ACTION (ALONG, ANY, N. L (R) = L (R) & SQ(R) ∩ MV *(B. K*) = Ø)

ELSE

ACTION(TOKING, ANY, N. DISP(*W. K, B. K*) ⊂ DISP (*W. K, B. K) & N.* L(R) = L(R))

END — IF;

END - DO;

END - STRAT.

В приведенной стратегии используются следующие доски:

L (FIG) — описание вертикали фигуры FIG;

DISP (FIGI, FIG2) — описание прямоугольника с вершинами в клетках, занятых фигурами F1G1 и FIG2;

MV (FIG) — описание ходов фигуры FIG;

OPP (FIGI, FIG2) — описание вертикальной оппозиции фигур FIGI, FIG2;

ANY — обозначение произвольной доски;

SQ (FIG) — описание положения фигуры;

POPP(FIG1, F1G2) — описание позиций, в которых через ход возможна оппозиция фигур FIGI, FIG2 (предоппозиция [3]);

AL(FIG1, FIG2) — описание позиций, в которых от фигуры FIG1 к фигуре FIG2 можно дойти, сделав более чем 3 хода королем.

Элементарные действия позволяют реализовать следующие преобразования:

LINE— отсечение короля противника по вертикали;

NEWLINE — оттеснение короля противника к краю доски;

DUMMY — промежуточный ход ладьей в случае предоппозиции;

ALONG — отступление ладьи по линии отсечения;

TOKING — оттеснение короля противника своим королем.

Рассмотрим работу *SDL*-компилятора при применении приведенной стратегии для следующей позиции: белые — ладья g6, король *e*6, черные—король *c*5. Для простоты предложены ответы черных, приводящие к короткой партии.

После инициализации компилятор выделяет действие *LINE* и строит дерево *p*-стратегии глубины 3, с первым ходом Кре5 и для каждого ответа черных с соответствующим ходом белых: 1... Кр *b*5 2. Л *c*6; 1... Кр *b*4 2. Л *c*6; 1...Кр *c*4 2. Л *b*6. Далее выполняется цикл реализации (PL-интерпретации) построенной *p*-стратегии в исходной позиции. При этом выполняется и печатается ход Кр *e*5 и компилятор принимает ответ противника. Предположим черные выбрали ход Кр *c*4, тогда следующим выполняемым компилятором ходом будет ход Л *d*6. После того как будет получен ход черных (предположим это ход Кр *b*4), *p*-стратегия отсечения короля реализована и компилятор включает модуль «Выделение очередного оператора ACTON», который сканирует текст исходной стратегии.

Очередным оператором является оператор *DO-WHILE.* Проверяется дескриптор *MV* (*B.K*) ≠ Ø, и так как он выполняется, то рассматривается тело оператора цикла. В данном случае это условный оператор, и так как L (R) ∩ MV *(L (R)* ∩ *MV (B. K) = Ø B. K*) = Ø, то в полученной позиции выполнен дескриптор первого IF-условия и компилятор выделяет для выполнения действие *NEWLINE.* На этом работа модуля «Выделение очередного оператора ACTION» завершается и компилятор строит дерево *p*-стратегии для реализации выделенного действия.

В данном случае оно будет состоять из единственного хода Л *c*6, который и будет выполнен в текущей позиции. Предположим черные ответили ходом Кр *b*3. Так как очередное действие реализовано и не достигнут конец стратегии, то компилятор продолжает обработку стратегии. Оператор *END-1F* пропускается, следующий оператор, *END-DO,* требует проверку дескриптора заголовка цикла и так как для полученной позиции он выполняется, то следующим рассматривается условный оператор (первый оператор тела цикла).

Ни один из дескрипторов IF-оператора не выполняется, что влечет выполнение его ELSE-оператора, т. е. действия TOKING, которое реализуется компилятором как ход Кр *d*4. Если черные ответят Кр *b*4, то компилятор выдаст ход Л *b*6, согласно действию при оппозиции NEWLINE (последовательность шагов выделения этого хода, аналогична описанному выше). Таким же образом на ответ черных Кр *a*6, белые ответят ходом Л *b*1 (действие ALONG) и если Кр *a*4, то Кр *c*4 (действие TOKING). Наконец после ходов Кр *b*3, Л *b*8 (действие DUMMY), Кр *a*4, Л *a*8 X, дескриптор цикла DO-WHILE(MV (B. K) ≠ Ø) не выполняется и следующий за концом цикла оператором стратегии выбирается оператор END-STRAT. Обработка этого оператора приводит к завершению работы компилятора, с выдачей записи сыгранной партии:

Բ. Կ. ԿԱՐԱՊԵՏՅԱՆ

ԽԱՂԵՐԻ ՍՏՐԱՏԵԳԻԱՆԵՐԻ ՆԿԱՐԱԳՐՄԱՆ ԲԱՐՁՐ ՄԱԿԱՐԴԱԿԻ ԼԵԶՎԻ

Առաջարկված է բարձր մակարդակի լեզու (SDL), որը հնարավորություն է տալիս տարրական օրինաչափությունների միջոցով ներկայացնել «դիրքերի դասերի նկարագրություն → հնարավոր գործողություններ» տիպի օրինաչափությունները: Դիտարկվում է երկքայլանի ստրատեգիաների սինթեզման պրոցեդուրա, որը SDL-լեզվով տրված օրինաչափությունների հիման վրա կառուցում է համապատասխան ստրատեգիաներ: Առաջին քայլից հետո ստացվում է կարուցվող ստրատեգիային համապատասխանող տարրական օրինաչափությունների հաջորդականություն, երկրորդ քայլի ընթացքում ամեն մի ստացված տարրական օրինաչափության համար կառուցվում է համապատասխան խաղային ենթածառ:

ЛИТЕРАТУРА

1. *Адельсон-Вельский Г. М., Арлазаров В. Л., Донской М. В.* Программирование игр, М., Наука, 1976.
2. *Погосян Э. М.* Адаптация комбинаторных алгоритмов, Ереван, Изд-во АН АрмССР, 1983.
3. *Погосян Э. М., Амбарцумян М.* А., *Аджабян Н. А., Арутюнян Ю. Г., Джндоян Л. О., Карапетян Б. К.* Пакет прикладных программ для исследования методов адаптации комбинаторных алгоритмов управления (АКА 1,2. Отчеты ВЦ АН АрмССР за 1976—1980 гг.
4. *Лангер Р., Миллс X., Уитт Б.* Теория и практика структурного программирования, М,, Мир, 1982.
5. *Ван Тассел Д.* Стиль, разработка, эффективность, отладка и испытание программ, М., Мир, 1981.
6. *G. Таненбаум Э.* Многоуровневая организация ЭВМ, М,, Мир, 1979.
7. *Bratko I., Michie D.* A representation for pattern-knowledge in chess end games, Advances in Computer Chess, 2, 31 - 56, Edinburgh, 1980.
8. *Bratko I.* Knowledge-based problem solving. In AL3 Machine Intelligence, 10. 73 - 101, 1981.
9. *Pitrat J.* A chess combination program, which uses plans. Artificial Intelligence, 8, 275 - 321, 1977.
10. *Wilkins D.* Using knowledge to control tree searching. Artificial Intelligence, 18, 151, 1982.

3. E. M. Pogossian, АЛГОРИТМЫ АДАПТАЦИИ В КОМБИНАТОРНЫХ ПРОБЛЕМАХ КЛАССИФИКАЦИИ И УПРАВЛЕНИЯ, Computing Center of Academy of Sciences , Moscow, DrSci defense thesis, 1985.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

1. АКТУАЛЬНОСТЬ ТЕМЫ. В работе исследуются возможности построения общих математических методов адаптивного синтеза алгоритмов в комбинаторных оптимизационных проблемах классификации и управления.

Оптимизационные проблемы определяются множеством исходных данных, или входных записей, понятием оптимальных решений при каждой входной записи и множеством всевозможных решений, классами алгоритмов, в рамках которых производится поиск, решений индивидуальных задач и, наконец, критериями качества алгоритмов, на основе которых формируются требования к построению оптимальных алгоритмов. При постановке и решении многих оптимизационных проблем, имеющих практический интерес, весьма часто требуется учитывать следующие условия:

1) На разных стадиях построения решений индивидуальных задач приходится оперировать с различными гипотезами о них;
2) Критерии качества решений индивидуальных задач часто либо не имеют алгоритмического представления, либо трудно вычислимы;
3) Известны или допустимы лишь локальные средства улучшения качества решений индивидуальных задач, что может быть, в частности, следствием реального отсутствия существенных средств улучшения или чрезмерно высокой стоимости их получения.

Проблемы адаптации комбинаторных алгоритмов характеризуются тем, что множество выходных записей каждой из них является множеством записей алгоритмов, а поиск решений производится в ограниченном условиями 1-3 классе алгоритмов, а именно, в классе алгоритмов адаптации. Алгоритмы адаптации предназначены для итеративного локального преобразования текущих гипотез-решений, направленного к последовательному улучшению этих гипотез относительно заданного критерия качества и, в конечном итоге, к достижению искомых решений. Их уточнение проводится нами в классе алгоритмов предельных вычислений. При этом исследования принципиальных возможностей алгоритмов адаптации с целью достижения наибольшей общности результатов проводятся без установления порогов на число допустимых элементарных вычислений при каждом шаге улучшения. В то же время при построении конкретных алгоритмов адаптации указанные условия должны учитываться в обязательном порядке, так как без этого область допустимых решений существенно расходится с требуемой.

Наши исследования проблем адаптации комбинаторных алгоритмов условно могут быть разделены на две части.

В первой из них исследуются *проблемы расшифровки классификаций или описаний*, в которых множества выходных записей состоят из алгоритмов классификации некоторых заранее заданных и, вообще говоря, комбинаторных множеств элементарных описаний. Проблемы расшифровки классификаций имеют общенаучный интерес, поскольку возникают, как правило, с каждой формой деятельности человека. Они непосредственно связаны с проблемами оптимального комбинаторного управления, информационного поиска, синтеза закономерностей и др.

Исследование классов проблей расшифровки классификаций представляет интерес не только потому, что многие проблемы синтеза алгоритмов известны именно в такой форме, но и потому, что существуют основания считать возможным представление каждой комбинаторной проблемы в этой стандартной форме.

Комбинаторность проблем обуславливает высокую специфичность поиска решений для каждой из них, что затрудняет обобщение достижений в отдельных проблемах, их сравнение, анализ и классификацию. Представление в единообразной форме позволяет разработать общий аппарат построения оценок сложности проблем, ввести понятие типа сложности проблем, на основе которой можно надеяться построить адекватную классификацию комбинаторных проблем, исследовать сравнительные характеристики различных классов алгоритмов адаптации и др.

Во второй части работы исследуются вопросы *адаптивной регуляризации комбинаторных проблем*, или соответствующих разрешающих алгоритмов.

В обширном многообразии проблем достаточно эффективные методы решения известны лишь для отдельных классов. При сведении новых проблей к сравнительно хорошо изученным говорят о *регуляризации* этих проблем. Широко известны, например, методы регуляризации некорректных проблем. Вопросы описания классов комбинаторных проблем с приемлемой сложностью разрешения и сведения к ним заданных проблем имеют важное теоретическое и практическое значение. В нашем предположении регулярные комбинаторные проблемы характеризуются наличием данных об общих свойствах множеств входных-выходных записей; наличием решений индивидуальных задач и соответствующих окрестностей входных-выходных записей, свойства которых легко вычисляются; возможностью планирования в множестве окрестностей входных записей разной степени исследованности и др. Уточнение условий локальности усиления, посредством введения специальных порогов на приращение оператора качества алгоритмов и число допустимых вычислений при каждом шаге усиления, определяет одну из разновидностей проблемы адаптации комбинаторных алгоритмов, которая в дальнейшем исследуется для комбинаторных проблем управления. Последние уточняются на основе определения конечных позиционных игр двух лиц с противоположными интересами, в которых понятие оптимального управления (возмущения) расширено на произвольные ситуации, а оператор качества алгоритмов управления определяется либо как временная сложность выбора оптимальных решений в произвольных ситуациях, либо как среднее отклонение решений от оптимальных, при фиксированном времени выбора решений.

Особый интерес к проблемам адаптации алгоритмов классификаций и оптимального комбинаторного управления в последние десятилетия объясняется, в частности, необходимостью автоматизации процессов обработки данных, вызванной экспоненциальным ростом научно-технической информации, требованием принятия решений в сложных для человека ситуациях, ставшие возможными с появлением ЭВМ и робототехнических систем. Теория и методы решения этих проблем в настоящее время находятся на стадии становления и составляют предмет интенсивных исследований на стыке многих научных дисциплин – оптимального комбинаторного управления, распознавания образов, теории игр, программирования игр, информатики, принятия решений, сложности вычислений, синтеза алгоритмов, искусственного интеллекта и др.

2. ЦЕЛЬ РАБОТЫ. Цель работы – построение общих математических методов адаптивного синтеза комбинаторных алгоритмов классификации и управления, а также методов оценивания эффективности таких алгоритмов. Она детализируется следующим образом:

1. Построить математическую модель адаптации классификаций, основанную на методах теории алгоритмов и дискретного анализа и объединяющую основные направления исследований в этой области.

2. Установить общие границы возможностей алгоритмов адаптации классификаций, получить их сравнительные характеристики и условия устойчивости на основе единообразных сложностных критериев.

3. Исследовать связь сложности проблем синтеза классификаций с длинами тестов таблиц, порожденных множеством возможных классификаций. Построить бесстреборные алгоритмы оценивания длин минимальных тестов конкретных таблиц.

4. Получить полное решение следующих комбинаторных проблем; для заданных конечного множества **M** и целочисленных наборов $l_1,\ldots, l_k$ и $m_1,\ldots, m_k$ построить m_j, подмножеств множества **M** мощности l_j, $1 \leq j \leq k$. таких, чтобы все эти множества попарно не содержались бы друг в друге и чтобы для произвольного натурального числа i число всех различных подмножеств мощности i, полученных из этих множеств, было бы наименьшим; для заданных конечного множества **M** мощности n и числа $m < 2^n$ построить систему из m подмножеств M такую, чтобы число всех различных подмножеств мощности i^*. полученных из элементов этой системы, было бы наименьшим, где i^* – минимальное число i от 0 до n такое, что существует хотя бы одна система вышеуказанного типа с числом подмножества меньшим, чем число всех подмножеств **M** мощности i.

5. Для определенного выше класса комбинаторных проблем оптимального управления требуется:

- разработать адекватные критерии эффективности алгоритмов управления и методы их практического измерения;
- исследовать сложностные вопросы разрешимости проблемы адаптации алгоритмов управления;
- определить класс алгоритмов управления, способных к регуляризации;
- сформулировать и исследовать проблему синтеза классификаций, согласованную с исходной проблемой управления.

3. КРАТКАЯ ИСТОРИЯ ВОПРОСОВ. Конкретные методы индуктивного обобщения рассматривались еще в 17-ом веке Ф. Бэконом и в дальнейшем были развиты Д. Миллем в 19-ом веке.

Интенсивные исследования адаптивного синтеза алгоритмов классификации начались с 60–х годов под названием "обучение понятиям", "формирование образов", "индуктивный вывод", "адаптация" и др. Ранние исследования характеризуются обилием эвристик как в алгоритмах синтеза, так и в критериях оценки их эффективности, отсутствием корректных постановок проблемы классификации, стремлением к построение универсальных методов синтеза, малым числом доказанных результатов. Возникает настоятельная необходимость в построении теории адаптивного синтеза классификаций, Первоначальные результаты в этом направлении связаны с именами В. М. Глушкова, Ф. Розенблата, М. А. Айзермана, Э. М. Бравермана, Я. З. Цыпкина, М. М. Бонгарда, В. А. Якубовича, Г. Д. Блока, К. Штайнбуха, В. Н. Вапника, А. Я. Червоненкиса и др.

Поиски общих моделей в рамках функционального анализа привели в 1964 г. к методу потенциальных функций М. А. Айзермана, Э. М. Бравермана, А. И. Розоноэра, объединивших ряд известных схем синтеза персептронного типа. Используя идеи стохастической аппроксимации, Я. З. Цыпкин в 1965 г. построил модель адаптивного синтеза параметрических алгоритмов, включающую. в частности, методы потенциальных функций и статистических решений. Однако существенный класс проблем адаптивного синтеза алгоритмов, не получивших параметрического представления, в том числе проблемы синтеза алгоритмов в форме описаний в языках заданных типов (так называемые структурные или лингвистические методы), индуктивный синтез программ, индуктивный вывод грамматик, расшифровка автоматов, оставался вне этой модели. Вне ее оставались также рассматриваемые Ю. Л. Журавлевым и его учениками алгоритмы синтеза классификаций, основанных на принципе частичной прецедентности, алгоритмы Э. Ханта, Н. Н. Айзенберга, метод эмпирических предсказаний и др.

В 1967 г. Э. М. Голд с целью построения всеобщей теории индуктивного синтеза выделил класс алгоритмов предельных вычислений. Развитие этой теории, интенсивно продолжающееся и в настоящее время, связано с именами Х. Патнама, Л. Блюма, И. Блюма, Я. И. Барздиня, Р. В. Фрейвалда и др.

Стремясь построить минимальное покрытие для возможно широкого класса практически апробированных алгоритмов адаптивного синтеза классификаций, автор в 1975 г. предложил модель индукторов и индуктивных выводов [8, 9], оказавшуюся сравнительно узким подклассом алгоритмов предельных вычислений и удовлетворяющую целям единообразного представления алгоритмов синтеза классификаций известных типов, а также алгоритмов расшифровки автоматов [12, 13, 14, 19, 25]. На основе указанной модели был развит фрагмент теории адаптивного синтеза классификаций [14, 23, 25, 27] – составной части общей теории индуктивного синтеза алгоритмов. В частности, была установлена связь между нижними оценками сложности проблем расшифровки классификаций и длинами минимальных тестов [25], что породило необходимость построения алгоритмов оценки длин минимальных тестов конкретных таблиц.

Понятие теста было введено С. В. Яблонским в 1956 г. Оно нашло широкое применение при диагностике управлявших устройств и построении алгоритмов классификации, в частности алгоритмов частичной прецедентности Ю. И. Журавлева. Асимптотические оценки длин минимальных тестов для произвольных таблиц заданных типов с разбиением на одноэлементные классы получены А. Д. Коршуновым (1970г.), а для произвольных разбиений – А. Е. Андреевым (1961г.).

Автором в [5, 27] предложены беспереборные алгоритмы построения нижних оценок длин минимальных тестов для произвольных конкретных таблиц с произвольным разбиением на классы. Они основаны на полном решении сформулированных выше проблем построения систем несравнимых и произвольных множеств с наименьшим числом подмножеств. В первой из этих проблем искомая система множеств о равными мощностями *1* впервые была построена Макалеем в 1927 г.; им же доказана ее экстремальность для подмножеств модности *1* – 1. Обобщение теоремы Макалея для небинарного случая и подмножеств произвольной мощности получено Клементсом и Линдстромом в 1969 г. Независимо, автором в 1970 г. было получено полное решение указанной проблемы [5] и включено в кандидатскую диссертацию (1970г.).

В 1950г. К. Шеннон сформулировал проблему построения "самоулучшающейся программы", "вырабатывающей собственные принципы игры". А. Сэмюзль в 1959 г. построил программу игры в шашки (достигшую уровня мастера), основанную на принципах параметрической адаптации и полного запоминания прецедентов. Известны исследования по индуктивному синтезу эндшпильных стратегий Х. К. Брутяна, И. Д. Заславекого и Л. В. Мкртчян, по усиление шахматных программ на основе пополнения справочных различных типов Г. М. Адельсон–Вельского, В. Л. Арлазарова, М. В. Донского, М. М. Ботвинника и его учеников, по усилению посредством синтеза закономерностей Ж. Питра, Р. Михальского и др. Возможности усиления алгоритмов управления с использованием описаний проблемной среды в языках, близких к естественным, интенсивно исследуются в ситуационном управлении, а также в других направлениях искусственного интеллекта. Исследованиями Н. А. Шанина, Г. С. Цейтина, С. Ю. Маслова и др. выявлены основные направления усиления алгоритмов поиска выводов в исчислениях. Систематическое математическое исследование взаимосвязи проблем оптимального комбинаторного управления и синтеза классификаций дано Р. Бенерджи (1969г.). Конкретные алгоритмы синтеза образов позиций в шахматах рассматривались Х. К. Брутяном, М. В. Донским, Ж. Питра, А. А. Стогнием, А. Д, Идиным и др. Введение понятия оптимального управления в произвольных ситуациях позволило автору сформулировать проблему оптимального комбинаторного управления при операторе качества типа среднее отклонение решений от оптимальных [25, 27]. Доказательство разрешимости проблемы Шеннона при логарифмических ограничениях на "стоимость" одного шага усиления анонсировано им в [31], а теорема о локально проверяемых достаточных условиях усиления алгоритмов управления – в [27]. Структура данных и алгоритмы, способные к адаптации при их изменении, описаны в [28, 30], а понятие регуляризованной комбинаторной проблема в [32]. Проблема синтеза классификаций при комбинаторном управлении, объединяющая известные работы в этом направлении, и ее сравнительные характеристики опубликованы в [23047]. Построение пакета программ для исследования алгоритмов адаптации на базе шахматной проблемы–посредника и ЭВМ Минск-32 проводилось с 1973 по 1979 г.г. [26]. Оно продолжается в настоящее время на основе ЭВМ серии ЕС. Результаты экспериментальных исследований алгоритмов адаптации представлены в [29].

4. НАУЧНАЯ НОВИЗНА. В диссертации получено обобщение проблем и методов теории адаптивного синтеза алгоритмов классификации. Доказано, что ряд известных ранее разобщенных проблем, в частности более семи типов проблем синтеза образов ситуаций, проблемы расшифровки автоматов, поиска стратегии управления и др., могут быть представлены в единой форме проблемы расшифровки классификаций, а соответствующие алгоритмы, используемые при решении указанных проблем, являются разновидностями специального класса алгоритмов предельных вычислений – индуктивных выводов с индукторами.

Разработан фрагмент теории сложности проблем расшифровки классификаций, включающий теоремы об общих и сравнительных характеристиках индуктивных выводов и индукторов различных классов, об условиях устойчивости указанных характеристик, о связи нижних оценок "энтропийной" сложности инициально пустых проблем расшифровки с длинами минимальных тестов соответствующих таблиц, а также алгоритмы построения нижних оценок длин минимальных тестов произвольных таблиц. Последние основаны на специально развитом аппарате, в рамках которого получено полное решение двух вышеуказанных комбинаторных проблем по безпереборному построение систем несравнимых и произвольных множеств с наименьшим числом подмножеств и вычисление числа этих подмножеств для произвольной заданной мощности.

Сформулировано понятие регуляризованной комбинаторной проблемы, выделены структуры данных и специальный тип алгоритмов, способных к регуляризации при изменении данных типов цели, планы, образы ситуаций, закономерности. Поставлена проблема синтеза оптимальных образов ситуаций, системно согласованная с исходной проблемой регуляризации комбинаторных алгоритмов управления, получены сравнительные характеристики и рекомендации по практическому синтезу образов в проблемах управления. Установлена возможность сравнения эффективности информационных структур управления посредством их моделирования на проблемах-посредниках типа конечных позиционных игр двух лиц с противоположными интересами и нулевой суммой. А для последних получены локально проверяемые достаточные условия усиления алгоритмов управления и доказана разрешимость проблемы адаптации при логарифмических ограничениях на сложность каждой итерации. На базе шахматной проблемы-посредника под руководством автора и с его непосредственным участием разработан и реализован пакет программ, в рамках которого экспериментально подтверждена правильность ряда выбранных принципов регуляризации в алгоритмах управления.

5. ПРАКТИЧЕСКАЯ ЦЕННОСТЬ. Полученное в диссертации обобщение методов теории адаптивного синтеза классификаций и развитый на этой основе фрагмент теории сложности позволяют объединить усилия различных направлений исследований синтеза классификаций; открывают возможности переноса результатов с более развитых областей на сравнительно новые, построения единого аналитического аппарата, выделения непреодолимо сложных (при заданных постановках) проблем синтеза классификаций, изменения постановок таких проблем, либо соответствующего рационального перераспределения усилий.

Проблемы построения систем несравнимых и произвольных множеств с наименьшим числом подмножеств входят в основной фонд комбинаторного анализа. К ним сводятся многие технические и теоретические вопросы диагностики управлявших систем и построения алгоритмов распознавания. Полученные полные решения этих проблем могут найти непосредственное использование в указанных прикладных областях.

Стремительное развитие многочисленных научных направлений, вызванное появлением ЭВМ, обострило потребность в построении объединяющих концепций. Понятие регуляризованной комбинаторной проблемы направлено к выявлению природы и формализации ряда направлений исследований искусственного интеллекта, комбинаторного управления, информатики, сближению их с теориями комбинаторной оптимизации и синтеза алгоритмов. Уточнение структур данных и постановки проблем синтеза в проблемах комбинаторного управления открывает возможности их математического исследования, оценки сложности и выбора практически приемлемых постановок. Предложенные алгоритмы управления могут найти применение в адаптивных системах управления производством и диспетчерскими пунктами, в робототехнических устройствах и др. Постановка проблемы синтеза классификаций, согласованная с критериями оптимальности исходной проблемы управления, открывает возможность оценивания уровня приближенности конкретно построенных классификаций и выбора приемлемых. Разработанный пакет программ, являясь инструментом общего пользования, автоматизирует исследования в области адаптации комбинаторных алгоритмов управления. Пакет может также служить основой при сравнении эффективности информационных структур управления с целью их оптимизации.

6. АПРОБАЦИЯ РАБОТЫ. Результаты диссертации полностью или частично докладывались на семинарах в Ереване (ВЦ АН Арм. ССР и ЕрГУ, отделение физ.-мат. наук АН Арм. ССР), Москве (МГУ. НИИСИ, ВИНИТИ, ИПУ, Общемосковский семинар по искусственному интеллекту), Ленинграде (ЛГУ, ЛЭГИ), Киеве (институт кибернетики); излагались в докладах на конференциях и симпозиумах, Всесоюзных семинарах и школах (Всесоюзной межвузовской конференции по надежности, Ленинград, 1966 г.; Всесоюзной конференции по теоретической кибернетике, Новосибирск, 1969г.; Всесоюзном семинаре по теории графов, Одесса, 1972 г., 1973 г.; Всесоюзном семинаре по комбинаторному анализу, Москва, 1976, 1981 г. г.; Международной конференции по искусственному интеллекту, Тбилиси, 1975 г.; Всесоюзной конференции по Математическому обеспечению моделирования сложных систем, Киев, 1977 г.; Всесоюзной конференции по теории программирования игр, Дилижан, 1976 г.; Всесоюзном симпозиуме по ситуационному управлению, Одесса, 1976, 1978, 1981 г. г.; Всесоюзном симпозиуме по кибернетике, Тбилиси, 1976 г; Всесоюзной школе–семинаре по адаптивным системам, Алма–Ата, 1978 г.; Всесоюзной школе–семинаре по машинным методам обнаружения закономерностей, Рига, 1979 г.; Всесоюзной конференции по семиотическим моделям при управлении большими системами, Клайпеда, 1979 г.; Всесоюзном симпозиуме по теории адаптивных систем, Ленинград, 1979 г.; Всесоюзной школе–семинаре по банкам данных, Цахкадзор, 1980 г.; Всесоюзной конференции по методам математической логики в проблемах искусственного интеллекта, Паланга, 1980 г.; Международном совещании по искусственному интеллекту, Репино, 1980 г.; Всесоюзной конференции по человеко-машинным обучающим системам, Телави, 1979 г.; Всесоюзной конференции по синтезу, тестированию, верификации и отладке программ, Рига, 1981 г.; входили в содержание спецсеминаров, проведенных автором в Ереванском государственному университете в 1975, 1976 и 1978 г. г.

7. ПУБЛИКАЦИИ. По теме диссертации имеется 33 публикации [1-33]. В работе [10] оценка числа I-индукторов принадлежит О. А. Саркисян, в [12, 13, 20, 23] автору принадлежат исходная модель и идеи теорем, а в [2] – участие в разработке алгоритма.

Разработка и реализация пакетов и экспериментов [26, 29] проводилась сотрудниками сектора "Познавательных алгоритмов и моделей" ВЦ АН Арм. ССР Аджабян Н. А., Амбарцумяном М. А., Арутюняном Ю. Г., Джндоян Л. О., Карапетяном Б. К., Симоняном Р. Г. с 1973 г. по настоящее время на основе идей автора, при научном руководстве и непосредственном его участии.

8. ОБЪЕМ РАБОТЫ. Диссертация состоит из введения и пяти глав – объемом в 301 стр., трех приложений объемом в 23 стр., библиографии из 17S наименований и списка использованных сокращений.

СОДЕРЖАНИЕ РАБОТЫ

Во введении сформулированы, основные результаты диссертации, кратко изложена история исследования рассматриваемых вопросов.

1. ГЛАВА 1 посвящена классификации комбинаторных проблем, В § I дается общая постановка исследуемых нами комбинаторных оптимизационных проблем, которые формулируются в классе алгоритмов предельных вычислений с оракулом без указанных выше ограничений типа 3. Далее из них выделяются конечные проблемы класса $\mathbf{L}_1$ с операторами качества типов минимальной пиковой (π) или средней (σ) временной сложности, Последняя понимается как суммарное количество элементарных вычислений и обращений к оракулу в процессе решения индивидуальных задач. Такое уравнивание "в правах" элементарных вычислений и обращений к оракулу позволяет рассматривать в пределах одного и того же класса как проблемы, не требующие обращений к оракулу (инициально полные), так и проблемы с обязательным обращением к нему (инициально пустые). Определяются также проблемы синтеза алгоритмов – индуктивного, дедуктивного и трансформационного.

В § 2 дается постановка проблем расшифровки описаний (ПРО) и адаптации алгоритмов при ограничениях типа 1. Особенностью ПРО является возможность представления множества всех предполагаемых решений индивидуальных задач в форме множества T' разбиений – абсолютных пар множеств, в некотором заранее фиксированном алфавите элементарных описаний M. Тем самым ПРО определяются конкретизацией следующих параметров: конечного множества T' описаний всех возможных решений заданной проблемы и множества M всех элементарных компонент этих описаний, отношения ρ между множествами входных – U, выходных – T' записей, списков C_1, и C_0 – элементарных вычислений алгоритмов и их вопросов к оракулу. В наших исследованиях ПРО рассматривается при операторе временной сложности.

Проблема адаптации комбинаторных алгоритмов при ограничениях типа 1 определяется нами следующим образом: для заданных комбинаторной оптимизационной проблемы L, конечного множества U алгоритмов, предназначенных для решения L, заданных порогов на обязательное приращение оператора качества гипотез-решений и на максимальное число допустимых элементарных вычислений при однократном применении некоторого алгоритма адаптации (усилителя) к произвольному f из U, построить усилитель максимальной кратности применения к каждому f из U.

В § 3 рассматривается возможность сложностной классификации комбинаторных проблем на основе их представления в форме ПРО. Выделяется класс π(σ)- переборных ПРО, к которым причисляются проблемы, для которых π(σ)-сложность, т. е. пиковое или среднее число обязательных вычислений и обращений к оракулу, больше мощности множества всех рассматриваемых элементарных описаний решений. В том же параграфе высокая сложность комбинаторных проблем уточняется через π(σ)-неразрешимость. Для таких проблем предполагается, что π(σ)-сложность даже относительно оптимальных алгоритмов непреодолимо велика. Возникает проблема выбора адекватной меры эффективности работы алгоритмов приближенного решения комбинаторных проблем и построения методов её вычисления. В качестве такой меры в § 4 предлагается пара векторов, первой компонентой которой является усредненная временная сложность решений индивидуальных задач, а второй – числовая оценка отклонений этих решений от оптимальных. Относительно указанной меры вводится понятие оптимального алгоритма приближенного решения и доказывается его существование. Естественные упорядочения алгоритмов по этой мере не являются линейными, что вынуждает сравнивать алгоритмы по одной из её компонент – среднему отклонению, при фиксированной другой.

2. ГЛАВА 2 посвящена исследованию вопроса представления комбинаторных проблем в форме ПРО. В § 1 доказывается следующее утверждение: Для каждой из проблем – расшифровки автоматов по Б. А. Трахтенброту-Я. И. Барздиню, формирования понятий по Э. Ханту, обучения распознаванию образов по Я. З. Цыпкину-М. А. Айзерману и др., построению экстремального распознающего алгоритма по Ю. И. Журавлеву-В. В. Никифорову, обучения распознаванию образов по В. Н. Вапнику-К. Я. Червоненкису можно указать параметры инициально пустой ПРО такие, что соответствующая конкретизация ПРО будет эквивалентна этой проблеме. Аналогичное утверждение, но для инициально полных ПРО, доказывается для проблем синтеза стратегий в проблеме комбинаторного управления и распознавания тавтологий от n переменных.

Как уже отмечалось выше, алгоритмы предельных вычислений являются наиболее общей моделью процессов адаптации, индуктивного обобщения и др. и ПРО сформулированы именно в этом классе алгоритмов. В то же время при анализе возможности представления через ПРО известных в литературе проблем выясняется, что многие из них сформулированы каждая в своем специальном классе алгоритмов, общность которых не является очевидной и требует доказательств.

В § 2, 3 показано, что разрешающие процедуры указанных проблем составляют подкласс алгоритмов предельных вычислений – индуктивных выводов с индукторами, для которых проведена классификация в соответствии с решаемыми проблемами.

Пусть $\boldsymbol{T}$ – множество всех пар непересекающихся подмножеств конечного множества $\boldsymbol{M}$. Произвольное отображение f множества $\boldsymbol{T}$ в $\boldsymbol{T}$ назовем индуктором, если для произвольной пары множеств $<x, y> \in \boldsymbol{T}$ из $f(<x, y>) = <u, v>$ следует, что $|u \cup v| \geqq |x \cup y|$; при этом, если $|u \cup v| = |x \cup y|$, то $x = u$ и $v = y$.

Основное требование, предъявляемое в определении индуктора – увеличение числа классифицированных объектов по сравнению с той классификацией на множестве объектов, которая была задана в качестве исходной. При более сильных ограничениях получаем ряд специфических классов индукторов. В частности, потребовав, чтобы индукторы формировали гипотезы, не противоречащие исходной классификации, получаем класс так называемых согласующих индукторов.

Процесс индуктивного обобщения уточняется посредством понятий индуктивного вывода с обратной связью и без неё. Индуктивный вывод, по существу, есть последовательность применений заданного индуктора к входным записям. Условия применения индуктора заранее определяются и зависят, вообще говоря, от согласованности текущей гипотезы и вновь полученных данных о расшифровываемой паре множеств.

В § 2 уточняются понятия индуктора, индуктивного вывода и критерии их оценки, а в § 3 показывается возможность представления в рамках указанной формализации алгоритмов формирования понятий по Э. Ханту, обучения 3-х слойного персептрона, методов формирования экстремального распознающего алгоритма, тестовых алгоритмов и др. Возможность такого представления для алгоритмов параметрической адаптации к распознаванию образов, расшифровки автоматов и эмпирических предсказаний показана в работах [12, 13, 19].

3. В ГЛАВЕ 3 рассматриваются вопросы оценки сложности проблем расшифровки описаний как в классе алгоритмов предельных вычислений в целом, так и в классах индуктивных выводов.

В §§ 1-3 рассмотрены вопросы построения нижних оценок сложности проблем расшифровки описаний при варьировании определяющих их параметров. Указанные оценки могут быть получены из общих соображений и влияние выбранной концепции алгоритма для них несущественно, в то время как верхние оценки требуют построения конкретных разрешающих алгоритмов и зависят от выбора алгоритмического языка.

В теореме 1.3. § 1 доказывается, что пиковая временная сложность одного из достаточно представительных инициально пустых классов проблем расшифровки описаний может быть оценена снизу длинами специально построенных для таких проблем минимальных тестов. Рассматривается также возможность расширения полученного вывода на класс всех проблем расшифровки описаний.

Полученная связь пиковой сложности с длинами минимальных тестов требует построения методов их оценки для таблиц с произвольным разбиением на классы. Два таких метода описаны в § 3. Они основаны на беспереборных алгоритмах решения проблем о системах несравнимых и произвольных множеств с наименьшим числом подмножеств, описанных в § 2. Сформулируем эти решения.

Набором назовем всякое конечное множество элементов, рассматриваемых вместе с некоторым определенным на нем отношением линейной упорядоченности. Поднабором набора A назовём всякое конечное множество, составленное из элементов A, рассматриваемое вместе с отношением упорядоченности, индуцированным упорядоченностью A. Всякий поднабор A, отличный от A, назовем собственным поднабором A. Через I_n обозначим набор $\{1, 2, \dots, n\}$, а через $\mathbf{R}_n$ множество всех поднаборов I_n.

Элементы $\mathbf{R}_n$ назовём Δ_n-наборами; Δ_n-наборы длины k, $0 \le k \le n$, – Δ^k_n-наборами. Множество $\mathbf{R}_n$ предполагается лексикографически упорядоченным (A≻B). Последовательность Δ_n-наборов $A_1, A_2, \dots, A_m$ назовем " + "-уплотненной, если для $0 < j < m$, $A_j \prec A_{j+1}$ и не существует $A \in \mathbf{R}_n$ такого, что $A_j \prec A \prec A_{j+1}$. Δ_n-набор $B = \{b_1, \dots, b_{k_2}\}$ назовем k_2-расширением Δ_n-набора $A = \{a_1, \dots, a_{k_1}\}$, если $k_2 \ge k_1$ и при любом i от 1 до k_1 будет $a_i = b_i$. Расположенную в лексикографическом порядке последовательность $\Delta_n^{k_2}$-наборов, составленную из всех различных k_2 расширений всех наборов какой-либо последовательности $\Delta_n^{k_1}$-наборов $A_1, A_2, \dots, A_m$, либо какого-либо множества $\Delta_n^{k_1}$-наборов S, где $0 \le k_1 \le k_2 \le n$, назовем k_2-продолжением последовательности $A_1, A_2, \dots, A_m$, соответственно, k_2-продолжением множества S.

Обозначим i-ю компоненту произвольного набора A через $(A)_i$, где $1 \le i \le |A|$, Таким образом, $A = \{(A)_1, \dots, (A)_{|A|}\}$. По определению мы полагаем, что $(A)_0 = (\emptyset)_k = 0$, где A – произвольный Δ_n-набор, $0 \le k \le n$.

Произвольную целочисленную последовательность вида $\chi = (n, l_1, \dots, l_k, m_1, \dots, m_k)$ будем называть характеристической последовательностью, если она удовлетворяет следующим условиям: 1) $n \ge l_1 > l_2 > l_k \ge 0$; 2) $m_i \ne 0$ при $i = 1, \dots, k$. Будем говорить, что Δ_n-наборы A и B несравнимы по включению, если $A \nsubseteq B$ и $B \nsubseteq A$. Каждое множество попарно несравнимых наборов S характеризуется своей характеристической последовательностью $(n, l_1, \dots, l_k, m_1, \dots, m_k)$, где m_k – число наборов длины l_k, а n – мощность исходного множества.

При произвольном k, $0 \le k \le n$, через $\Gamma_n^k(1), \Gamma_n^k(2), \dots, \Gamma_n^k(C_n^k - 1), \Gamma_n^k(C_n^k)$ обозначим "+"-уплотненную последовательность, составленную из всех Δ_n-наборов длины k, а. через $L_n^{n-k}(j)$ при $1 \le j \le C_n^k$ обозначим разность $I \setminus \Gamma_n^k(j)$. Для произвольной характеристической последовательности $\chi = (n, l_1, \dots, l_k, m_1, \dots, m_k)$ уплотненным множеством L_χ типа χ назовем множество Δ_n-наборов вида: $L_\chi = \{L_n^{l_1}(1), L_n^{l_1}(2), \dots, L_n^{l_1}(m_1), L_n^{l_2}(\widetilde{m}_1 + 1), \dots, L_n^{l_2}(\widetilde{m}_1 + m_2), \dots, L_n^{l_k}(\widetilde{m}_{k-1} + 1), \dots, L_n^{l_k}(\widetilde{m}_{k-1} + m_k)\}$, где $L_n^{l_{j+1}}(\widetilde{m}_j) = I_n \setminus \Gamma_n^{n-l_{j+1}}(\widetilde{m}_j)$ и $\Gamma_n^{n-l_{j+1}}$ есть наибольшее в лексикографическом упорядочении $(n - l_{j+1})$-расширением последнего набора последовательности $\Gamma_n^{n-l_j}(1), \Gamma_n^{n-l_j}(2), \dots, \Gamma_n^{n-l_j}(\widetilde{m}_{j-1} + m_j)$, имеющего непустое $(n - l_{j+1})$-продолжение, при $j = 2, \dots, k - 1$, $\widetilde{m}_0 = 0$. Пусть также $N_i(S)$ равно числу всех различных поднаборов длины i наборов S. Тогда решение проблемы о системах несравнимых множеств с наименьшим числом подмножеств дает следующая

Теорема 2. 3. Для произвольной характеристической последовательности χ, если K_χ – класс всех множеств попарно несравнимых наборов типа χ, а L_χ – уплотненное множество типа χ, то:

1) Класс K_χ непустой тогда и только тогда, когда можно построить уплотненное множество L_χ типа χ;
2) Если класс K_χ непустой, то для произвольных i и S, где $0 \le i \le n$, $S \in K_\chi$, число всех различных подмножеств мощности i, полученных из наборов множества L_χ не больше числа всех различных подмножеств мощности i, полученных из наборов множества S.
3) $N_i(L_\chi) = C_n^{n-i} - \sum_{j=0}^{n-l_n-1} C_{n-a_{j+1}}^{n-j-i}$, где $a_j = \left(\Gamma_n^{n-l_k}(\widetilde{m}_{k-1} + m_k)\right)_j$, $j = 1, \dots, n - l_k$, $0 \le i \le l_k$.

Решение проблемы о системах произвольных множеств с наименьшим числом подмножеств дается теоремой 3. Пусть $\{2^n\}$ – множество всех двоичных чисел от 0 до $2^n - 1$ и $\omega: \{2^n\} \to R_n$ – взаимно-однозначное отображение $\{2^n\}$ на R_n такое, что $\forall\alpha(\alpha \in\{2^n\} \,\&\, \omega(\alpha) = A \to A = \{i |\ (\alpha)_i = 0\})$, где $(\alpha)_i$ – цифра в i-ом разряде α, $1 \le i \le n$. При этом старшим разрядом считается первый. Через $c(\alpha)$ обозначим число "1" в записи двоичного числа α. Для произвольной пары $\eta = (n, m)$, $n > 0$, $m < 2^n$ пусть:

$K_\eta = \{S |\ S \subseteq R_n \,\&\, |S| = m\}$, $i_\eta^* = \min_i \left(0 \le i \le n \,\&\exists S\left(S \in K_\eta \& N_i(S) < C_n^i\right)\right)$ и $E_\eta = \{F(2^n - 1), F(2^n - 2), \ldots, F(2^n - m)\}$, где $F(j) = \omega(\beta_j)$, β_j – двоичная запись j при $j = 0, 1, \ldots 2^n - 1$.

Теорема 3.3.

1. Для произвольной пары $\eta = (n, m)$, $n > 0$, $m < 2^n$, число i_η^* существует;

2. $\forall S\left(S \in K_\eta \to N_{i_\eta^*}(E_\eta) \le N_{i_\eta^*}(S)\right)$;

3. $N_i\left(E_\eta\right) = C_n^i - \sum_{j=0}^{n-i-1} C_{n-a_{j+1}}^{n-i-j}$, где $a_j = (\omega(\alpha^i))_j$, $1 \le j \le n - I$, $0 \le i \le n$, $0 \le \alpha^i \le \alpha^*$, $\nexists\alpha(\alpha > \alpha^i \,\&\, c(\alpha) = i)$ и α^* – двоичная запись числа $m - 1$.

4. Если $\sum_{j=1}^{p} 2^{n-j} \le m - 1 < \sum_{j=1}^{p+1} 2^{n-j}$, при $p = 0, 1, \ldots, n - 1$, $\sum_{j=1}^{0} 2^{n-j} = 0$, то $i_\eta^* = p + 1$.

В § 4 исследуется сложность расшифровки описаний в классе индуктивных выводов и индукторов. В частности, в следствии 1 к теореме 6.4 утверждается, что для произвольного индуктора существуют пары множеств (ПМ), индуктивный вывод которых имеет максимально возможную сложность. При этом максимум достигается в одном из классов проблем расшифровки описаний при специальном выборе обучающей последовательности.

Указанное следствие в определенном узком смысле отрицает существование оптимального индуктора в индуктивных выводах. Можно было бы предположить, что оптимальный индуктивный вывод окажется возможным, если брать не фиксированную заранее обучающую последовательность, а такую, которая порождается по определенным правилам в самом процессе вывода.

В теореме 7.4 однако, показано, что для индуктивных выводов, использующих согласующие либо несогласующие индукторы, не существует оптимальных алгоритмов, порождающих вопросы к "учителю".

Далее рассматриваются вопросы сравнительной оценки сложности классов ПМ относительно индуктивных выводов с согласующими либо несогласующими индукторами. Требование согласования встречается во многих алгоритмах индуктивного обобщения и потому заслуживает специального внимания. Согласующие индукторы нами сравниваются с несогласующими. В частности, теорема 8.4 описывает некоторый достаточно естественный класс согласующих индукторов, которые расшифровывают произвольный класс ПМ не лучше несогласующих. В теореме 10.4 показано, каким образом для некоторых классов ПМ (фактически – для произвольных классов) может быть построен несогласующий индуктор, который почти во всех случаях будет расшифровывать этот класс правильнее, чем некоторый согласующий индуктор. В теореме 9.4 утверждается, что возможны классы, оптимальную расшифровку которых можно произвести либо только согласующими, либо только несогласующими индукторами.

В теоремах 11.4 – 15.4 анализируется сложность решения узких классов ПРО, в частности, симметрических и монотонных, посредством индуктивных выводов с I индукторами, без обратной связи и с ней.

В рассмотренных теоремах ПМ предполагалась расшифрованной, если гипотеза полностью совпадала с ней. Возникает естественный вопрос: сохранятся ли преимущества несогласующих индукторов, если в условии расшифрованности требовать лишь частичного совпадения? Точнее, чтобы симметрическая разность между гипотезой и искомой ПМ не превышала заданного порога, зависящего от мощности этой ПМ. В § 5 доказана устойчивость указанного свойства несогласующих индукторов относительно фактически произвольного распределения порогов (теоремы 16, 17). Исключение составляют расшифровки классов ПМ, мощность которых приближается к мощности всевозможных ПМ (теорема 18), либо когда вводится специальное ограничение о согласованности исходных данных с искомой ПМ (теоремы 19–21).

4. В ГЛАВАХ 4, 5 рассматриваются вопросы комбинаторной регуляризации и построения конкретных алгоритмов адаптации. При этом в качестве множества входных адаптируемых алгоритмов, рассматривается множество решений проблем оптимального комбинаторного управления, точнее, модельного представителя подобных проблем – шахматной игры.

Исследуемые нами проблемы оптимального комбинаторного управления (ПОУ) определяются пятерками $<S, C, D, M, Q>$, где S, C, D, Q – произвольные конечные множества, соответственно, ситуаций, управлений, возмущений и целевых ситуаций (выигрышных, проигрышных, ничейных), а M – частичное отображение из $S \times C \times D$ в S, описывающее допустимые последовательности воздействий управлений и возмущений на ситуации (правила игры). Произвольное отображение из $S \setminus Q$ в C называется алгоритмом поиска управлений, если из $f(s) = c$, при $s \in S$, $c \in C$, следует, что $\exists d \in D \exists s' \in S\{(s, c, d, s') \in M\}$. Аналогично определяются алгоритмы поиска возмущений. Предполагается, что управления и возмущения последовательно воздействуют на заданные ситуации для перевода их в целевые ситуации множества Q. Тогда каждая пятерка указанного вида определяет некоторое ориентированное дерево возможностей (дерево игры), в котором вершинами являются ситуации множества S, исходящие ребра в каждой из них указывают на допустимые в них управления либо возмущения (соответственно ярусу дерева), а также ситуации, в которые преобразуются исходные ситуации.

Для произвольной ситуации P дерева возможностей Γ стратегией G из P, или P-стратегией, называется произвольное поддерево G дерева Γ с корнем P такое, что в i-ом ярусе G:

1. При очередности воздействия управлений из каждой вершины этого яруса исходит не более одного ребра, указывающего на одно из допустимых в ней управлений согласно M;

2. При очередности применения возмущений из каждой вершины этого яруса исходят либо все допустимые в ней возмущения, либо ни одно.

P-стратегия называется наиболее выигрышной в ситуации P, если для выигрышной или ничейной P стратегия G обеспечивает выигрыш или ничью независимо от алгоритмов поиска возмущений, а для проигрышной P стратегия G удовлетворяет условию $\min\limits_{x \in G_P} \alpha(x)/\beta(x)$, где G_P – множество всех Р-стратегией с известным выигрышем одного из противников, $\alpha(x)$ – число проигрышных, $\beta(x)$ – число проигрышных или ничейных концевых вершин P-стратегии x. Оптимальным управлением в ситуации P называется первое управление произвольной наиболее выигрышной P-стратегии.

ПОУ определяется как проблема построения алгоритма поиска оптимальных управлений (в произвольных заданных ситуациях множества S) одновременно оптимального относительно заданного оператора качества алгоритмов, в частности, временной сложности.

5. В ГЛАВЕ 4 в соответствии с представлением о регулярных комбинаторных проблемах формулируются исходные предположения о структуре оптимальных алгоритмов управления и основные вопросы их построения, рассматриваются возможности обнаружении относительного усиления и разрешимости проблемы адаптации шахматных алгоритмов.

В § 1 дается описание класса алгоритмов комбинаторной регуляризации и выделяются основные этапы их исследования. В § 2 дается обоснование выбора шахматной игры ($\text{ПОУ}^{\text{Ш}}$) в качестве проблемы-посредника при изучении методов комбинаторной регуляризации для проблем класса ПОУ. Обоснование исходит из общих требований к проблеме-посреднику, а именно: требованиям принадлежать классу исследуемых проблем, допускать исследования в целом, как некоторой замкнутой системы, быть по возможности общедоступной, позволять многократное воспроизведение всех этапов решений, допускать разбиение на подзадачи регулируемом сложности, располагать архивом описаний разных попыток се решений и др., которым ПОУ^{Ш} удовлетворяет.

В § 3 формулируется гипотеза 1 об оптимальном алгоритме f^{*} решения ПОУ^{Ш}, исходя из которой выделяется класс адаптируемых алгоритмов $\mathbf{B}_1$ (входных-выходных записей в ПАКА1). Предполагается, что f^{*} основан на подпрограммах поиска по каждой заданной позиции оптимальных управлений из множества записей w, содержащих все такие управления; при этом релевантность управления к заданной позиции распознается не более чем за $\log_2 r$ элементарных вычислении, где r имеет порядок числа всевозможных допустимых передвижений фигур на доске и не превышает $5 \cdot 10^4$.

Из гипотезы 1 следует, что произвольный алгоритм f_0 из $\mathbf{B}_1$ естественно представить как систему $<\theta(W), f_{01}, f_{02}, f_{03}>$, где $\theta(w)$ – тезаурус (упорядоченное множество W записей типов: алгоритмы, тексты, формулы), f_{01} f_{02} – алгоритмы поиска записей в тезаурусе и стратегий в графе возможностей для ПОУ^{Ш}, соответственно, f_{03} – координатор взаимосвязи f_{01}, f_{02} и входных-выходных устройств. Алгоритм f_0 отображает ситуацию s в управление в S, алгоритм f_{01} –в подмножество записей, f_{02} –пары (S; список записей) в управление в S. При этом как f_{01}, так и f_{02} могут быть разрешающими алгоритмами для ПОУ^{Ш}. Данное представление проблемы оптимизации f_0 порождает подпроблемы ее локальной оптимизации, основанные на изменении одних компонент f_0 при фиксированных других. Наши исследования адаптации в значительной степени связаны с одной из них – проблемой накопления, организации и поиска записей в тезаурусе (НОПТ), в которой при заданных координаторе f_{03} и алгоритме поиска в графе f_{02} требуется выбрать множество записей W', построить такие тезаурус $\theta'(W')$ и алгоритм f'_{01} поиска из $\theta'(W')$ записей по произвольным ситуациям-запросам, что для произвольной другой пары $<\theta(W), f_{01}>$ того же назначения, выполнялось бы неравенство: $< \theta'(W'), f'_{01}, f_{02}, f_{03}> \geq <\theta(W), f_{01}, f_{02}, f_{03}>$, где упорядочение $\geq$ порождается оператором качества алгоритмов в ПОУ^{Ш}.

В § 4 формулируются ожидаемые результата и актуальные проблемы теории адаптации комбинаторных алгоритмов.

Как было отмечено ранее, измерение среднего отклонения при поиске приближенных решений оптимизационных комбинаторных проблем нетривиально и связано с оценкой отклонений решений индивидуальных задач от искомых, как правило, для очень большого числа таких задач. Поэтому практическое использование этого критерия возможно только для таких оптимизационных проблем, в которых, удается указать локально проверяемые методы его оценки. В § 5 описано решение этой проблемы для шахматной игры. Оно основано на предположении о том, что доли выигрышей, ничьих и проигрышей заданного алгоритма поиска управлений f в играх с произвольными другими алгоритмами при фиксированной начальной позиции P прямо пропорциональны долям концевых вершин соответствующих типов в стратегии, порожденной f из P. Тем самым проблема сравнения "силы" двух заданных алгоритмов f_i и f_j сводится к проблеме установления относительного расположения этих алгоритмов в последовательности, полученной линейным упорядочением шахматных алгоритмов по результатам абсолютного турнира. Последний же понимается как круговой турнир между всевозможными шахматными алгоритмами, когда каждая пара алгоритмов встречается в матче при специальных условиях, обеспечивающих отсутствие априорных преимуществ какому-либо из них, вытекающих из характера организации матчей.

Принимая гипотезу о том, что в соответствии с местом i заданного алгоритма f_i в абсолютном турнире можно указать константу b_i такую, что влево от окрестности $[i + b_i, i - b_i]$ f_i только проигрывает, вправо – выигрывает, а в самой окрестности может как выигрывать, так и проигрывать, доказана следующая

Теорема 1.4. Для произвольных алгоритмов f_i и f_j, если f_j выигрывает матч у f_i, и существует такое множество алгоритмов F', что: мощность F' больше b_i, f_j – выигрывает, а f_i проигрывает матч с каждым алгоритмом из F' в абсолютном турнире для $F' \cup \{f_i, f_j\}$ то f_j сильнее f_i.

Аналогичное утверждение справедливо для случая, когда матч f_i с f_j оканчивается ничьей. Тем самым вопрос проверки усиления заданного алгоритма f_j относительно исходного f_i сводится к построению множества F' и оценке параметра b_i. Даже при отсутствии оценки для b_i можно утверждать: чем больше $|F'|$, тем вероятнее, что f_j сильнее f_i. Последнее утверждение дает необходимые основы для экспериментального исследования усилителей.

Отметим, что результаты § 5, 6 справедливы для произвольных конечных позиционных игр двух лиц с противоположными интересами. Их изложение в рамках шахматной игры приведено с целью сохранения цельности всех рассмотрений для конкретной проблемы.

В § 6 доказывается разрешимость проблемы Шеннона, формализованной в виде требования о построении метода адаптации шахматных алгоритмов при "разумных" ограничениях на число $\Delta(t)$ элементарных вычислений при произвольном шаге t адаптации. Соответствующая проблема получается из сформулированной выше проблемы адаптации с ограничениями типа 1 снятием требования о строго последовательном усилении гипотез при каждом акте их преобразования. Разрешимость же доказывается в предположении о том, что класс шахматных алгоритмов задается посредством некоторого исчисления с конечным числом l правил вывода. Последнее позволяет построить процедуру перечисления всех сочетаний из $b = 2 \cdot \max\limits_i b_i$ шахматных алгоритмов и, последовательно проверяя в них наличие более сильного алгоритма, чем текущий алгоритм-гипотеза, в пределе с определенной точностью построить оптимальный. Точнее, пусть c – максимальное число элементарных вычислений при выполнении арифметических операций или сравнении двух чисел, а c_m и c_g – для проведения одного матча и восстановления алгоритма по его номеру, соответственно. Алгоритм ω разрешает проблему адаптации шахматного алгоритма f с точностью m, если предельное решение $\omega(f)$ отлично от истинного не более чем на m абсолютных мест.

Теорема 4.4. Проблема адаптации в классе шахматных алгоритмов, порождаемых l правилами вывода, при $\Delta(t) \geq 2c_m b^2 + 5cb(1 + \log_l(t + b + 2)) + c_g b$ разрешима с точностью $2b$.

Разрешимые проблемы адаптации естественно дополнить требованием выбора оптимальных относительно заданного оператора качества алгоритмов. При использовании временной сложности указанный выбор будет существенно зависеть от соотношения между точностью решения, значением $\Delta(t)$, кратностью повтора порождаемых алгоритмов и др.

Теорема 6.4. Проблема адаптации в классе шахматных алгоритмов, порождаемых l правилами вывода, при $\Delta(t) \geq 8c_m b^2 + 10cb(1 + \log_l(t + 2b + 2)) + 2c_g b$ и однократном последовательном порождении наборов из $2b$ алгоритмов, разрешима с точностью $3b$.

6. В ГЛАВЕ 5 рассматриваются вопросы представления, синтеза и комплексного использования ряда структур, для которых имеются достаточно оснований относить их к знаниям. Для наглядности изложения каждой такой структуре, именуемой "знанием", дается шахматная интерпретация.

"Знания" представлены нами через разновидности записей типа формул, текстов, алгоритмов. В § 1 определяются "знания", используемые при распознавании индивидов одинакового назначения. С этой целью рассматриваются формулы языка первого порядка L с конечным числом функциональных и предикатных символов. Структура A^* для L определяется естественным образом для конечного универсума A, что позволяет каждую формулу языка $L(A^*)$, полученного добавлением к L имен всех индивидов из A, представить в эквивалентной конъюнктивной нормальной форме (к. н. ф.).

В множестве индивидов различаются простые (пешка, клетка) и составные (позиция, стратегия). Для последних вводится понятие разложения на простые. Полное описание заданного индивида a понимается как множество всех неэквивалентных истинных на разложении a к. н. ф. с хотя бы одним вхождением в них имени из разложения a. Частичное описание a или престо описание a, есть произвольное подмножество полного. Описание индивида интерпретируется как сообщение о наличии вполне конкретного индивида. Формула E языка $L(A^*)$ называется описанием в A^*, если E применимо хотя бы к одному индивиду из универсума; множество всех индивидов, к которым E применимо, называется E-классом. Описания определяют дескрипторы выделения индивидов одинакового в определенном смысле типа.

В § 2 определяется частичный порядок θ_1 на множестве всех описаний – отношение выигрышности, а также понятия $<P, F, L>$ стратегий и их объединений. Отношение θ_1 отражает объективно существующий порядок на множестве индивидов, в частности, ситуаций, относительно их возможностей и связей с целевыми ситуациями ПОУ, а также наше наличное знание этого порядка и способность к его конкретному использованию.

$<P, F, l>$ стратегии определяют. P-стратегии, концевые вершины которых имеют описания не хуже F относительно упорядочения θ_1 и одновременно это свойство в каждой такой вершине x сохраняется до глубины не менее l посредством навешивания в них дополнительных x-стратегий. $<P, F, l>$ стратегии и их объединения описывают знания типа комбинация, форсированный вариант, эндшпильная стратегия и др.

"Знания" типа цели понимаются нами как описания ситуаций, определенным образом связанных с достижением целевых ситуаций исследуемой проблемы, Сами же процедуры, определявшие связь рассматриваемых ситуаций с другими ситуациями известной ценности, как описания некоторого смысла этих ситуаций. Индуктивное определение § 3 уточняет понятие цели в ПОУ^{Ш} следующим образом:

1. Произвольное описание E целевых позиций ПОУ^{Ш} одного и того же типа, называется $<E, l^*>$ целью или глобальной целью, где l^* – константа порядка глубины дерева игры;

2. Пусть F – $<F_1, l_1>$ цель, где F_1 – описание ситуаций, а $l_1 \leq l^*$; произвольное описание E называется $<F, l>$ целью или локальной целью, если при мощности E-класса, равной m, можно указать натуральные числа n и l такие, что $n \leq m$ и для каждой позиции P E-класса с частотой n/m существует некоторая $<P, F, l>$ стратегия.

Данное определение целей одновременно является некоторой гипотезой о способе их порождения.

Произвольная конечная последовательность целей, согласованная с отношением выигрышности θ_1, понимается как план. Произвольная $<F, l>$ цель E называется $<F, l>$ образом позиций, если описание E является тупиковым, т. е. никакие подмножества дизъюнктов к. н. ф. E или частей этих дизъюнктов свойством быть $<F, l>$ целью не обладают.

Определение образа позиций характеризует шахматные понятия через цели "в улучшенном" представлении. Данная гипотеза хорошо согласуется с результатами нашего анализа более 200 шахматных понятий и позволяет взглянуть на проблему синтеза образов с позиций, отличных от традиционных.

В определении оптимального $<F, l>$ образа E^*, как следствие гипотезы 1, хотелось бы учесть три не всегда совместимых требования: максимальные мощность и частоту существования $<F, l>$ стратегий для E^*-класса позиций, при одновременно минимальном времени распознавания принадлежности позиций к этому классу. Поэтому E^* в § 4 определяется как $<F, l>$ образ с максимальной частотой существования $<F, l>$ стратегий, и такой, что при минимальной длине записи E^*-класс достигает наибольшей мощности, Для представления знаний о том, какие цели являются необходимыми этапами при достижении заданной, в § 5 вводятся частичное упорядочение θ_2 – "цель-подцель", отношение θ_3 – "часть-подчасть" и др.

Вышеуказанные типы "знаний" в этом или ином виде являются компонентами закономерностей, которые в § 6 определяются как тексты вида $u \Rightarrow v$, где u–распознаватель, а v– следствие закономерности, u и v– формулы в расширении L^0 языка L.

Вопрос о том, является ли та или иная запись тезауруса знанием, решается в контексте структуры конкретного алгоритма управления. Однако представляется, что можно выделить ряд контекстно независимых требований, позволяющих сформулировать автономные проблемы синтеза записей со многими предпосылками быть знаниями. В § 7 рассмотрены возможности постановки таких проблем.

Проблемы синтеза элементов языка рассматриваются в основном для формул типа "образы ситуаций". Отличие нашей постановки проблемы синтеза образов (ФОС3) от традиционной (ФОС2) в том, что, если в ФОС2 основной информацией об искомом образе являются указания о принадлежности отдельных заданных описаний ситуаций к образу, то в ФОС3 в исходную информацию дополнительно включается принцип, по которому ситуации объединяются в один и тот же образ, а именно, их общий $<F, l>$ смысл или процедура перехода от ситуаций образа к целевым ситуациям с описанием F.

Ранее в § 3 главы 3 была установлена определенная связь между параметрами инициально пустых ПРО с нижними оценками $\pi(\sigma)$-сложности этих проблем. Используя выводы соответствующей теоремы 5.3, в § 8 проводится сравнительный анализ проблем формирования образов ситуаций, эквивалентных по множеству искомых решений, но имеющих либо традиционную постановку (ФОС 1, 2), либо порожденную нами из проблем оптимального комбинаторного управления (ФОС3). В частности, можно полагать, что $\pi(\sigma)$-сложность уменьшается при уменьшении отношения числа недопустимых решений к общему числу всех решений, а также при переходе от инициально пустых проблем к полным. Тем не менее имеются основания сложность рассмотренных постановок ФОС считать высокой даже при приемлемых параметрах их представления. Одно из них в высокой стоимости проверки принадлежности элементарных описаний к искомому образу, второе – в большой трудоемкости выбора оптимального образа в множестве возможных. Формулируется более реалистическая постановка ФОС, в которой априори предполагается известной хотя бы одна $<P, F, l>$ стратегия; требуется, при заданных времени распознавания принадлежности к образу и правдоподобии достижения $<F, l>$ целей построить $< F, 1>$ образ с максимальной мощностью $<F, 1>$ класса.

При поиске оптимальных представлений проблем проверка условий эквивалентности может оказаться весьма трудной во многих случаях обусловленной универсальным характером здания этих условий. В теореме 1.8 выделяются два специфичных достаточных условия эквивалентности проблем ФОС1 и ФОС2.

В соответствии с определенными выше "знаниями", алгоритмы управления описанного ранее класса B_1 должны быть детализированы таким образом, чтобы их работа основывалась на совместном анализе и учете всех этих "знаний", что, в свою очередь, должно обеспечить возможность их адаптации. Необходимая детализация и описание процесса использования "знаний" на уровне блок-схемы проведены в § 9. Полученные алгоритмы управления образуют подкласс B_1 и состоят из следующих, основных блоков: 1. формирование области целей, релевантных к заданной ситуации, и вычисление правдоподобия каждой из них; 2. выделение целей наиболее правдоподобных и одновременно выигрышных; 3. формирование справки к анализируемой ситуации на основе поиска "знаний" в тезаурусе; 4. анализ справки с переходом либо к выдаче управления на основе справки, либо к синтезу собственной стратегии; 5. формирование пучка планов на основе разложения текущей цели относительно графов "цель-подцель", "часть-подчасть" и их анализа; 6. непосредственный синтез стратегий по заданной цели; 7. выполнение управления; 8. координация последовательности управлений, возмущений и останова.

В приложении 1 приведено краткое описание проблемно-ориентированных пакетов, прикладных программ АКА-1, предназначенного для исследований с помощью ЭВМ "Минск-32" возможностей доказательства либо опровержения гипотез, связанных с адаптаций комбинаторных алгоритмов управления. Организационно пакет имеет структуру типа библиотеки. Общий объем программ АКА-1 составляет 25.000 операторов на "Фортране" и ЯСК.

В приложении 2 описаны результаты экспериментального исследования на основе пакета АКА-1 возможности распознавания наиболее правдоподобных в анализируемых ситуациях целей из заданного списка. Принцип действия соответствующего алгоритма основан на идее оценивания в отрезке [0, 1] степени выполненности компонент формулы, представляющей цель, а также тех или иных мотивов, связанных с достижением этой цели. Для каждой из 36 рассмотренных шахматных целей построены схемы вычисления правдоподобия, которые являются взвешенными графами типа и/или с определенной упорядоченностью обхода вершин с концевыми вершинами – атомарными формулами, описаниями целей или мотивов их достижения. Число таких формул в реализованных нами схемах достигало 350 при глубине схем порядка 14.

Эффективность работы алгоритма вычисления правдоподобия целей проверялась посредством сравнения оценок правдоподобия заданных целей, построенных соответствующей программой для тестовых позиции, с аналогичными оценками экспериментаторов. Результаты сравнений показали принципиальную возможность применения процедур рассмотренного типа в алгоритмах поиска управлений.

В приложении 3 рассмотрены мотивы выбора закономерностей типа "правдоподобные цели ⇒ рекомендуемые решения" и приведены результаты экспериментов по их алгоритмическому построению.

Эксперименты проводились на основе партий шахматного сборника "Информатор" и показали достаточную осмысленность синтезированных закономерностей для их возможного использования при усилении алгоритмов управления.

Работы автора по теме диссертации

1. Погосян Э. М. О повышении надежности многослойного обучаемого автомата. Тезисы док. Всесоюзной межвузовской конференции по надежности. Ленинград, 1966, с, 123 135.

2. Брутян Х. К., Погосян Э. М. Алгоритм поиска сокращенного описания множеств. Труды ЦНЛНСУ, Ереван, 1970, с. 32-41.

3. Погосян Э. М. О количественных характеристиках тестов. Тезисы док. Всесоюзной конференции по теоретической кибернетике, Новосибирск, 1969, с. 92.

4. Погосян Э. М. О количественных характеристиках тестов. Известия АН Арм. ССР, т. V, № 1, 1970, с. 32–49.

5. Погосян Э. М. О нижних оценках длин минимальных тестов, ДАН Арм. ССР, т. L, №2, 1970, с. 71–74.

6. Погосян Э. М. Δ-графы и некоторые их свойства. ДАН Арм. ССР, т. LIV, № 4, 1972, с. 199–203.

7. Погосян Э. М. I-индукторы и некоторые их свойства, ДАН Арм. ССР, т. LVШ, № 1, 1974, с. 10–14.

8. Погосян Э. М. Сравнительные характеристики согласующих индукторов. ДАН Арм. ССР, т. LX, № 3, 1975, с. 129–132.

9. Погосян Э. М. Индуктивный вывод с обратной связью. ДАН Арм. ССР, т. LX, № 5, 1975, с. 193–197.

10. Погосян Э. М., Саркисян О. А. Об одной формализации индуктивного обобщения, сб. ВЦ АН Арм. ССР, 1975, с. 157–175.
11. Погосян Э. М. Системы множеств с наименьшим числом подмножеств. Комбинаторный анализ, в. 4, МГУ, М., 1976, с. 16–21.
12. Карапетян Б. К., Погосян Э. М. Расшифровка автоматов как разновидность индуктивного вывода. Материалы VIШ Всесоюзного симпозиума по кибернетике, Тбилиси, 9–12 ноября, 1976, с. 79.
13. Карапетян Б. К., Погосян Э. М. Индукторы и их связь с методом эмпирических предсказаний. Вычислительные системы, № 69, Новосибирск, 1977, с. 102–112.
14. Погосян Э. М. К теории автоматического синтеза понятий. Информатика и семиотика, в. 8, ВИНИТИ, М., 1977, с. 125–152.
15. Погосян Э. М. Модельные системы информатики: сравнительный анализ методов теории решения задач и информатики. НТИ, в. 2, № 5, 1977, с. 16–19.
16. Погосян Э. М. Сравнение эффективности информационных структур методом турниров, Материалы Всесоюзной конференции "Математическое обеспечение моделирования сложных систем, Киев, 31 мая – 3. июня, 1977, с. 16–19.
17. Погосян Э. М. К сравнительному изучению алгоритмов индуктивного обобщения. Вопросы кибернетики, сб. "Адаптация в системах со сложной организацией", М., 1977, с. 35–39.
18. Погосян Э. М. Модельные системы информатики: основные критерии эффективности информационных методов и методы их вычисления на задачах-посредниках. Вопросы информационной теории и практики. № 35, ВИНИТИ, М., 1978, с. 75–83.
19. Погосян Э. М. Обучение как разновидность индуктивного вывода, "Техническая кибернетика", № 3, 1978, с. 62–69.
20. Мартиросян А. А., Погосян Э. М. Исследование устойчивости сравнительных характеристик согласующих индукторов. ДАН Арм. ССР, т. LXVII, № 4, 1978, с. 212–215.
21. Погосян Э. М. Итеративное усиление комбинаторных алгоритмов. Тезисы докладов Всесоюзной конференции "Семиотические модели при управлении большими системами", Клайпеда, 7–8 сентября, 1979, с. 56–57.
22. Погосян Э. М. Обучение в системах с комплексным использованием знаний. Тезисы докладов I Всесоюзной конференции "Человеко-машинные обучающие системы", Телави, 31 октября – 2 ноября 1979, с. 62–63.
23. Мартиросян А. А., Погосян Э. М. Об устойчивости сравнительных характеристик согласующих индукторов. Семиотика и информатика, в. 14, ВИНИТИ, М., 1979, с. 212– 215.
24. Погосян Э. М. Об автоматическом усилении стратегий управления. Семиотика и информатика, в. 12, ВИНИТИ, М., 1979, с. 129–134.
25. Погосян Э. М. К классификации проблем расшифровки описаний. ДАН Арм. ССР, т. LXV111, № 1, 1979, с. 29–34.
26. Погосян Э. М., Аджабян Н. А., Джндоян Л. О., Арутюнян Ю. Г., Амбарцумян М. А. Пакет прикладных программ для исследований методов автоматического усиления стратегий управления. Аннотация II Всесоюзной школы-семинара по машинным методам обнаружения закономерностей. Рига, февраль – март, 1979, "Вычислительные системы", № 79, 1979, с. 120.
27. Погосян Э. М. К оценке сложности и приближенной расшифровке комбинаторных описаний. ДАН Арм. ССР, т. LXX, № 2, 1980, с. 72–79.
28. Погосян Э. М. К проблеме адаптации комбинаторных алгоритмов. Тезисы докладов Всесоюзной конференции методы мат. Логики в проблемах искусственного интеллекта и систематическое программирование", Паланга, 3–5 сентября, 1980, с. 148 – 149.
29. Погосян Э. М., Амбарцумян: М. А., Арутюнян Ю. Г., Аджабян Н.А., Программа накопления и использования закономерностей типа: "правдоподобные цели ⇒ рекомендуемые

решения". Тезисы докладов Всесоюзной школы–семинара "Интеллектуальные базы данных", Цахкадзор, 10–18 марта, 1980. с. 31.

30. Погосян Э. М. К параметрам комбинаторной оптимизация и сравнительному анализу проблем формирования образов ситуаций. ДАН Арм. ССР, т. LXXII, № 4, 1981, с. 211 – 217.

31. Погосян Э. М. О разрешимости проблемы адаптации шахматных алгоритмов. ДАН Арм. ССР, т. LXXII, № 5, 1981. с. 269–274.

32. Погосян Э. М. Об алгоритмах комбинаторной регуляризации. Тезисы докладов Всесоюзной конференции "Синтез, тестирование, верификация и отладка программ. Рига, 22–24 сент. 1981, с. 117.

33. Погосян Э. М. Адаптация комбинаторных алгоритмов. Издательство АН Арм. ССР, Ереван, 1983, с. 296.

4.E. M. Pogossian, ON EXPLANATORY DEFINITION OF ENGLISH CLAUSE, 5th Intern. Cognitive Linguistics Conference, VrijeUniversiteit, Amsterdam, July 1996, pp. 209-214

ABSTRACT

An attempt of explicit definition of the scheme of basic affirmative indicative clauses and related syntax is described. Central idea is based on an interpretation of the pragmatic role of some basic constructions of English in the communication process and the fact of partition of all verbs into three classes: DO (active), BE and HAVE (stative) [2].

Feasible adjustments for some concepts are suggested aimed to make syntax presentation more transparent. This explanatory approach may rise the efficiency of English teaching to students already skilled in the grammar of a language to learn English one relaying on the analogies and differences between them, support the parsing of phrases for different computer applications and the linguistic/structural approach to pattern formation and recognition.

1. INTRODUCTION

1.1. Children catch the spirit of their native language very quickly listening to samples of discourse. Even if they make mistakes in the vocabulary, forms of the verbs, etc., they reproduce the main laws of language in the right way. Such efficiency in an inductive way of learning may be explained, particularly, by the highly impressionable abilities of children, as well as by the fact that these basic laws are not too complicated.

For foreigners, educated but deprived of both a natural language studying environment and an ability to memorize intensively the use of the deductive ways of learning, particularly understanding and using the grammar of the language might become effective and helpful also. That process could become more efficient, if the grammar description is organized with explanation of the origin of its rules and shell by shell, to present the information about the structure of the language in the order of priority for communication efficiency. This could be done by the units similar to the ones children acquire in their step by step development

In that process primary and very essential information gives the syntactical structure of the clause. It allows to identify the subjects, objects and relationships between them of some activities in the World , the time, aspect, voice, mode, subordination of components, etc., of that activities. Although the concept of a basic clause must be central for the syntax in the way that the whole variety of clauses could be derived from it in a natural way there is no its explicit and comprehensive definition. E.g. it is very difficult and unexplainable for foreigners to catch the idea of the appearance of the word “did” in interrogative and negative English clauses.

To try to extract the hierarchy of grammar knowledge, we need in a guideline for our choices. For that we first remind ourselves of some pragmatic functions of a language in the process of a human survival as a

Living Reality in the World, referring, for example to [1] , for finding a more detailed analysis of these functions.
Then we give an explicit definition of a scheme of English clause underlining the primacy of its nomination, subject intentions and *do, be, have* relations representation functions. Finally, a justification of some basic constructions of English syntax is suggested followed by a brief illustration of their consequences.

2. LANGUAGE AS A SURVIVAL INSTRUMENT (see also the Introductions to Parts 1,2 of the Monograph)
Living Realities (LR) are those that can preserve themselves, i.e. their main functions, in the extremely changeable and fearfully uncertain World. In that sense, LRs are realizations or implementations of survival methods, i.e. knowledge of how to survive. Thus, "living" of LR as a process is a cognition of the World followed by continuous reconstruction, renovation of LR to react adequately to actual or possible changes in the World.
Knowledge of LR, in fact, include a persistent part in a form of the actual construction of LR themselves - a result of evolution, and more flexible adaptive part in a form of memorized descriptions and instructions acquired mainly during the individually period of life as LR (compare with hardware and software of computers).
That persistent, genetic part of knowledge is so large and complete that in an ideally stable and unchanging environment, LR might have no need to renew it. Only breakdowns in the expectations of LR in anticipated and real events in the World cause the need to memorize the descriptions of situations and their resolution by appropriate actions [1].
The coordinated behavior of LR is also an instrument for survival. Communities of LR make possible results unachievable for individual ones. To coordinate their efforts, LR need in exchanging of relevant knowledge about themselves and the World. For that they prescribe identifiers or names, to the important realities of their life and organize their collective behavior by communication, i.e. transmissions from one to another relevant communication units to identify the situations and own relations of LR to them. Due to the concrete conditions, each community creates its specific communication symbolic system or language, with its own set of useful words (vocabulary) and rules of combination of them - the syntax of the language.
Before discussing these categories of language in more detail, let's remind ourselves that LR symbolize only socially important parts of their knowledge, which are necessary for communication primarily for production or exchange of activities.
The vocabularies of languages of LR and means of composition of their units at each moment of their development are mainly reflect the abilities of languages of LR to represent the parts of the World – realities. What LR communicate, seems, is only the top of the iceberg of their real knowledge.
Communicating LR transmit from one to another the names (the signs) of their knowledge to trigger in the other communication partners a desirable state, which could allow them to recognize and operate in problematic situations appropriately.
That possibly explains the fact that actual results of computer simulations of human intellectual abilities are successful for their "high school", symbolic ones, while are impeded in simulating "primitive", common for LR abilities. Consequently, this fact implies preferences in ongoing attempts of computer simulations [1].
Let us address now to communications in English.

3. A VIEW ON ENGLISH PHRASES
3.1. One of primary functions of English discourse, we assume, is in triggering the participants of the discourse into the target state of the problem situation. That is attained by transmitting to them some phrases of the language, d e s c r i p t i o n s, i.e. systems of names of the World 's realities.
It is important to note that the structure of the basic descriptions is standard [2]:
t w o o r d e r e d n o m i n a t o r s o f r e a l i t i e s a n d **b a s i c r e l a t i o n s** b e t w e e n t h e m comprised from one of the verbs *do, be , have* possibly with certain *modality*.
If situations are described from the human's point of view, *modals* represent human intentions, desires, duties, abilities, etc., to achieve something, to inform others, etc., in the problem situation. In other cases, they are the result of the prescription to the described realities of imagined relations similar to the human ones.

Do, be, have basic relations include information about a type of activity of realities, i.e. indicating the fact of presence of their activity (do) or existence (be), possession, belonging (have) of some realities to others. Historically, these basic relations had, we assume, primary importance in development of human doings, communications and were reflected in the construction of the basic structure of English syntax - scheme of a clause. Other discourse information related to a time or period of events, a manner by which a description is focused, an identification of an action's doer and receiver, etc., is coded by concretizations of that scheme of a clause using forms of times, aspects, moods, voices, etc. It may be considered as a secondary level of information relative to the main one - nomination and the basic relations information.

3.2. Let's illustrate our view by examples of basic Affirmative Indicative clauses at a Progressive Aspect of the Active voice.

We say: *The car* is *stopping near the policeman. Driving car was* stopping *near the policeman at the time. The car* will be stopping *near the policeman. We should* be stopping *near the policeman.*

These clauses have a simple structure: Nominator1 Form(be) Nominator2 or Nominator1 Form(Modal) be Nominator2.

Here: *the car, driving car, we, stopping near the policeman,* are Nominators. They name some realities and their positions in the phrase inform us about Subject /Object subordination between them.

Time and Person information is coded by forms of *be(is, was*) or forms of modals (*will, should*). The type of Aspect (here Progressive) is coded by Present participial of the verb *stop: stopping,* placed at the first position of Nominator2 .

We suppose that a slight generalization of the above structures can be represented as the syntactic scheme of Indicative clauses at the Active voice. Let's substantiate this assertion.

4. THE SCHEME OF BASIC INDICATIVE CLAUSES IN THE ACTIVE VOICE

4.1.1. The basic function of the words – the strings of letters in the alphabets, is nominative. This nominative function we differentiate in accordance with a static or dynamic nature of realities that they describe and a role they play in descriptions: principal or auxiliary. These functions may be inherit or depend from position of the word in a description. Words with inherit functions we differentiate as words of the following types or classes:

Nouns (Nn), Adjectives (Adj), Determiners (Dt), Adverbs (Adv), Prepositions (Pre), Pronouns (Pro), Conjunctions (Cnj) , Verbs (V or v), Modals (M).

E.g. an inherit function of Prepositions is in expressing relationships of things in the space : movement, position, etc. Along with words with only these functions we have ones that beside those functions may play other roles in phrases depending from a position there, e.g. be adverbs, particles of verbs, etc.

Words may have person, degree, time and aspect forms. All F o r m s (Fr) of each type of word may be listed or enumerated by generation rules.

The totality of all the words of these types and their forms is the v o c a b u l a r y of the language.

N o m i n a t o r s, in general, are combinations of arbitrary words of the vocabulary, except modals.

4.1.2.Modals indicate some deontic relation, modality - intention, duty, permission, obligation, etc., between Nominators. Logically, they are the following two-placed predicates: *will (shall), can, may, must, have to, ought to, need, dare.* Some modals have Past forms: *would, should, could, might, had to, needed.*

4.1.3.Each verb V has the following forms (FrV): infinitive (toV), person forms at Present (fPrV), Past (fPtV) and Future (fFtV), present (ingV) and past (edV) participles.

The forms fPr, fPt and fFt we name also Time/Person forms (tp).

The forms of all verbs are given by simple generation rules, except those of about 200 irregular ones, for which forms are listed.

All verbs may be split into three groups:

DO verbs (doV), BE verbs (beV) and HAVE verbs (hV).

DO or a c t i v e v e r b s name some of variety of *do* related to activity, process, change, movement, etc., in the World.

BE verbs and HAVE verbs are named also s t a t i v e v e r b s.

BE v e r b s include *be* and the verbs having a sense “being”

HAVE v e r b s include *have* itself, 19 verbs (see [2]) with a sense of "having ", the verbs with involuntary sense and complete mental action that indicate their presence, existence or possession by the Sbj at the moment of communication.

4.1.4.The first Nominator in a clause, or S u b j e c t (Sbj or sN), may be an arbitrary N without verb or its time/person forms.

Second Nominator, vN , in addition to the Sbj's possible components always includes in its first position some standard combination of forms of a verb to code Aspect, Voice and, for some cases, Time/Person information of the message.

A Nominator complementary to those combinations of verb forms for vN, is named an O b j e c t (Obj) of a clause.

The allowable combinations of the types of words in Sbj-s , vN-s and Obj-s may be listed.

4.2. The scheme (formula, type, code) of affirmative Indicative Basic Clauses of the Active voice, or **aIBC**, is defined as the following *positional* writing:

sN M dbh vN

or

sN dbh vN

where sN, M, dbh, vN are variables, sN- for a Nominator in the first or in the Subject position, M- for modals or their forms in the second position, dbh- for one of verbs: *do, be, have* or their forms in the third position, and vN - for verb Nominators in the forth position, correspondingly.

A Time/Person information of the event is coded by using an actually present in the scheme the first component of the triple M dbh V.

dbh component of the aIBC is missed if aIBC expresses a Simple (General) Aspect of the event .

Verb Nominator vN always contains at its first position forms of verbs which provide nominative, Aspect and Voice information but also may code Time/Person one, e.g. for Simple Aspect.

4.3.1. The focus of a message in any clause is the Subject. sN provides its identification by its position and other externally recognizable characteristics.

M codes information of the internal state of the Sbj.

dbh and forms of verbs in the vN specify the relation of the Sbj to the World (W). Namely, dbh identifies the type of that relation in terms of the basic or archaic ones: *be*- for underlying Sbj's being or existence, *do* - to indicate presence of some activity of the Sbj in the W, and *have*- for saying about Sbj's possession of something of the W.

Note, that these archaic relations *do, be* and *have* also correspond to the three fundamental instruments of human survival: human's production (*do*), human's own evolution at genetic or knowledge-based ways (*be*), and exchange activities for needed items (*have*).

Forms of verbs at the first position of the vN provide nominative and Aspect /Voice information for a concrete activity of the Sbj . They also may code Person /Time information if both M and dbh are missed.

4.3.2. Let's note that the difference of the aIBC from known scheme: (Noun phrase) (Verb phrase) [2], is in extraction of the M and dbh from Verb phrase and positioning them at the scheme in a explicit way.

As we can see also the aIBC concept will coincide in fact with the concept of clause in [2], if we accept that active verbs may be always interpreted as a transitive. (The notion of a transitive stative verb seems very unnatural. I.e. we consider active verbs transitive and the stative ones intransitive.) Even if this assertion turns out to be false for some verbs, following it in the presentation of the syntax will clarify some arguments for its use as the first level of approximation.

4.4. Before completing the definition of aIBC for compound and complex clauses, we illustrate at first aIBC for a Simple Aspect of Active voice , then demonstrate advantages of the aIBC presentation for some syntactic constructions of the Indicative mood.

5. THE SCHEME OF BASIC AFFIRMATIVE CLAUSES FOR A SIMPLE ASPECT

Basic clauses in a Simple Aspect of the Active voice may be naturally interpreted in frame of the aIBC. Indeed, the interpretation of phrases like: *Driving at high speed m a y be dangerous*. *We should have a great benefit. He will do the work.,* where *be, have* or *do* are used explicitly, is the following: corresponding *dbh* component in the scheme is missed due its actual redundancy - repetition in the vN.

Phrases such as: *A boy runs. The car moves. I do it*., etc., which underline an activity and use active verbs, may be interpreted as equal to phrases: *A boy* does *run. The car* does *move. I* do *do* it*., with* a scheme : sN

fPr(do) (do)N. In actually used phrases redundant *do* is missed and fPr functions are drived to the verb in the vN.
Phrases: *We exist many years*. *It reminds me of the school.,* with stative BE verbs, different from *be,* may be interpreted as equal to phrases: *We* are *exist many years*. *It* is *remind me the school.,* with archaic scheme : sN fPr(be) vN, and an actual one: sN fPr(v)N, with missed the redundant *be* in the dbh position.
At last, the phrases: *He belongs to the group. I hear you. We believe him.,* with stative HAVE verbs, different from *have,* have an archaic scheme: sN fPr(hv) vN , and an actual one: sN fPr(v)N, with missed the redundant *have* in the dbh position.

6. AN OUTLINE OF SYNTAX FOR THE BASIC INDICATIVE CLAUSES.

6.1 We are going to demonstrate that English syntax of Indicative clauses may be presented in compact and enough complete form if operate with aIBC .
Let's start with consideration of Aspect forms of the Active voice. To this end we will base ourselves on their transparent explanation in [3].
Information on activity or events in the World, coded in phrases, in their essential include the following:

- Time period T of an activity: Present, Past or Future,
- an Aspect of the activity relatively to the focus point T [3].

There are the following types of Aspects:

- Progressive: the activity is continuing now at the focus point,
- Perfective: the activity started at some period before the focus point in the past and just ended,
- Perfective +Progressive: the activity started at some period in the past before the focus point and is continuing now,
- General, or Simple: the activity exists also after the focus point of consideration..

The Aspect information at time T is coded in the vN part of aIBC at the following simple way:

- for Progressive Aspect: vN = ingV Obj,
- for Perfective Aspect: vN = edV Obj,
- for Perfective +Progressive Aspect: vN = edBE ingV Obj,
- for Simple Aspect: vN = v Obj or vN = tpV Obj (if there is no modal in the clause)..

Correspondingly *dbh* is equal to the following :

- for Progressive Aspect *dbh* = be,
- for Perfective and Perfective/Progressive Aspect *dbh* = have,
- for Simple Aspect *dbh* is omitted, as a rule.

Time /Person information is coded by actually present first component of the triple M dbh V in the clause.
Let's remind that only DO verbs usually have progressive Aspect.
6.4. For clauses: *A boy can make a shelf. A boy makes a shelf. A boy has made a shelf.,* clauses in Passive will be: *A shelf can be made by a boy. A shelf is made by a boy. A shelf has been made by a boy.*
As we can see a Sbj is "passive" in the Passive voice (in opposite to the Active voice), i.e. Sbj is not a Doer of the action but only Receiver of it.
There's no normally Passive for clauses with stative verbs [2, p.226]. E.g. for clauses like: *George has sat down on my hat. He is my friend.,* etc. Haven't Passive, too, clauses with Perfective /Progressive Aspect, like : *A boy has been making a shelf*., [2].
The Passive is formed by specific coding of an Aspect information for active verbs. Namely, it is using Aspect forms of the verb *be* in Active voice and edV participial of the current verb. Thus, aIBC in Passive will be :

- for Progressive Aspect: sN [M] be ingBE edV [*by*] Obj ,
- for Perfective Aspecte: sN [M] have edBE edV [*by*] Obj,
- for Simple Aspect: sN [M] be edV[*by*]Obj.

In Passive clauses a part:" *by* Obj ", often is omitted and *by* may be replaced by prepositions *: with, in, at*, etc.[2].
6.3. For further consideration we'll use the following notification.
Let denote by Y an actually present first component of the triple M dbh V in a aIBC by Yn the negative of Y .
We present a clause in a form C = sN Y C', too, where C' complement sN Y to the complete aIBC .
Time /Person information in clauses is coded by Y .Negative forms Yn for each Y are generated by known easy rules.
Let's remark that for all Active and Passive voice clauses Y is equal either a Modal, or BE ,or HAVE. The only exclusion are clauses in a Simple Aspect of Active voice when used verbs are different from BE and HAVE. At those clauses dbh component are omitted for reasons interpreted above . In further analysis and definitions for such clauses below we will suppose that Y denotes omitted verb DO (see also the *Note* at the end of this paragraph).

6.4. N e g a t i v e clauses, or **nIBC ,** are defined as clauses formed by replacing Y by Yn in clauses sN Y C'.

6.5. Interrogative, or q u e s t i o n , clauses (**qIBC** *)* for the first degree of their approximation qIBC are the following:

- C? and nC? with rising intonation , if we are pronouncing them.
- Y Sbj C' ?. and Yn Sbj C' ?
- Question tags : Sbj Y C', Yn ProSbj ? and Sbj Yn C', Y ProSbj ?

Here ProSbj means Pronoun corresponding to Sbj.

- .Wh Y C' ? or Wh Yn C' ? if the question is about the Sbj.

Here Wh corresponds to one of Sbj question words: Who, What, Which, Whose .
If Wh =Whose and N(Sbj) is the Noun of the Sbj, then q aIBC will be: Wh N(Sbj) TFrV C'?.

- Wh Y Sbj TFrV C'/P ?, if the question is about any part P of the TFrV C'.
  Here Wh is one of the corresponding P question words: Who, What, Which ,Whose, Where, When, Why,How, How many, How long, etc., C'/P is the rest of C' without P. [2]. For Yn a definition is similar.

6.6. *Note.* To construct interrogative or negative for the above exception clauses in a Simple Aspect of Active voice standard syntax recommends a use of DO for the verbs of all types: active and stative. In our logic BE and HAVE verbs had to use omitted tpBE and tpHAVE, correspondingly, instead of tpDO. For example, the interrogatives for: *We exist many years*., might become: *Are we exist many years?, or We exist many years, aren't we*?, instead of: *Do we exist many years?*, or: *We exist many years, don't we?.*
At the same logic: *He belongs to the group.*, might become:. *Has he belong to the group?* or: *He belongs to the group, hasn't he?,* instead of :. *Does he belong to the group?* and: *He belongs to the group, doesn't he?*
As we can see in these examples the use of DO, in opposite to BE and HAVE, adds some new sense of activity absent in the initial aIBC. This fact may be an argument also for splitting the verbs into DO, BE and HAVE classes.

7. INDICATIVE CLAUSES

7.1. We have defined above all types of **basic** clauses for the Indicative mood. Let's make an inductive step to complete that definition for complex and compound Indicative clauses. Here we outline it based on the examples from [4] only. The full definition will demand more details.
Given affirmative or negative IBC clause C (aIBC or nIBC) we name:
phrases *that C, what C, how C,* etc., as a N o u n p h r a s e C (NnPh C),
phrases *that C, which C, who C, whom C,* etc., as an A d j p h r a s e C (AdjPh C),
phrases *when C, where C, because C, if C, so that C,* etc., as an A d v p h r a s e C (AdvPh C).
C o m p l e x c l a u s e s are defined as the following:
if B and C are affirmative or negative IBC clauses in a form: sN [M] dbh vN, then the substitutions:

- of the Noun (Nn) in the Subject sN of the B by Noun phrases: *what C, or how C*, etc., or Noun in vN of B by Noun phrase: *that C,*
- Noun, Adj Noun or Noun Adj in B by Adj phrase: Noun AdjPhC,

- Adv in some couples with V, toV and ingV in B by Adv phrase AdvPhC

are c l a u s e s , or c o m p l e x c l a u s e s, of affirmative or negative types in a correspondence with the type of B.
B is named a principal clause and C - a subordinate one.
Correspondence of Times and Aspects of the component clauses of a compound clause is subject of special consideration.
This definition is expanded for other basic forms of B and C in an evident way.

7.2.Given affirmative or negative clauses B, C, a text B Cnj C is a c o m p o u n d c l a u s e where Cnj-s are Conjunctions: *and, but,* etc..
It will be affirmative or negative clause depending upon the combination of the types of B and C.

7.3.*Thus, I n d i c a t i v e c l a u s e s are affirmative, negative and question basic, complex and compound clauses.*

8. SUBJANCTIVE and CONDITIONAL CLAUSES. COORDINATION of TIMES and ASPECTS

Given affirmative or negative clauses B and C, conditional clause is a compound one with Adverbial clause:" *if* C ", in Present Simple and principal one in Future Simple.
Example.
Thus, in conditional clauses component ones have the same Aspect but Times are shifted for one.
Subjunctive clauses preserve the same structure but relates it to the Past of Simple or Perfect Aspects. Correspondingly, instead of *will* they use *would* , i.e. instead of Future Time they use Future in the Past .
Example.

9. CONCLUSIONS

9.1. We suggest to replace accents and modify some definitions in English basic syntax to try to make more explainable its constructions. Namely, we expand nominative role of verbal phrases, more concentrate attention on the role of modals and verbs DO, BE and HAVE,
slightly reclassify stative verbs and equalize them with intransitive ones. This modifications allow to set forth the core of syntax in compressed and logically clear form as well as to give complete definition for clauses in all their variety.
As a result the essential of the whole syntax in the form, compatible for its father systematic enrichment, may be easy cached by experienced in a symbolic analysis students from the first steps of their learning and used as a powerful deductive instrument for selfcorrections.
9.2.The above view of syntax, if would be correct, could imply a natural order in English presentation. Namely, first presenting the scheme of a clause, then step by step its forms, followed by indication of the parts of the language that ought to be regularly memorized: vocabulary, exclusions, prepositional verbs, etc. In Appendix we give some examples of that.
9.3.Author has not illusions to cover by some formulas an alive language in whole. But he is convinced in importance of attempts to extract the pragmatic nature of language expressions. He risked to try it for English based on his languages learning experience in an essentially deductive ways and the years of efforts to explain the mechanisms of knowledge formation [5].
Author only tried to demonstrate the advantages of this approach for some syntax constructions in an attempt to attract language professionals to make the final verdicts.
This work was inspired by the ideas of Terry Winograd discussed at the seminars with Arman Kuchukyan, as well as the lessons and support of his teachers and friends from language department of the American University of Armenia. Specially he thanks Patricia Boyle, Maria Bobrova and Virgil Strohmeyer for their exclusive efforts and patience to advance such terrible student as me.

REFERENCES
1. Winograd T., Flores F.. Understanding Computers and Cognition. A New Foundation for Design, Stanford, 1989.
2. Close R. A.. A Reference Grammar for Students of English. Longman Group, 1975, 352pp.
3. Asher A. Think About Editing. Haile & Haile Pub.,1993, 334pp.
4. Eckersley C.E.. Essential English for Foreign Students, Sofia, 1967.
5. Pogossian E.. Adaptation of Combinatorial Algorithms. Pub. by Academy of Sciences of Armenia, 1983, 293pp.(in Russian).
6.Reid W. Verbs and Nouns Number in English. A functional explanation. Langman Publ.,1991.
7. A. Martirosian , E.Pogossian. "Learning". Reference book on Intellectual Systems. Radio i Svjas Publishing Company, Moscow, (in Russian), vol. 2, pp. 206-231, 1990.

5. Martirosian A., Pogossian E. **"Learning by Revelation".** Reference book on Intellectual Systems. Radio i Svjas Publishing Company, Moscow, (in Russian), vol. 2, pp. 206-231, 1990

Глава 6: Обучение

А. А. Мартиросян, Э. М. Погосян

6.1. Модели обучения

Неформальные модели

В психологии под обучением понимают способность к приобретению ранее неизвестных умений и навыков. В ИС неформальное понимание обучения трактуется аналогично. Говорят, что ИС обучилась чему-либо, если она стала способной к выполнению некоторых процедур или решению некоторых задач, которые до этого была неспособна выполнять. Конечно, такое определение широко и расплывчато. Под него, например, подходит случай, когда в память ИС вкладывается готовая программа, которой ранее в ней не было. Поэтому часто специально подчеркивается, что обучение в ИС происходит таким образом, что она самостоятельно извлекает новую информацию из исходной или текущей информации. Другими словами, предполагается, что в процессе деятельности ИС анализирует имеющуюся информацию и на основе анализа извлекает из нее полезные закономерности.

При имитации в ИС процедур обучения активно используются модели обучения, известные в физиологии и психологии. Первые программы, демонстрирущие возможность обучения, — программы моделирования условных рефлексии гомеостатических процессов [Гаазе-Рапопорт и др., 1987], опирались на чисто физиологические модели обучения, разрабатывавшиеся в школе Павлова. Позже такие жесткие модели уступили место моделям, опирающимся на *ассоциативнуй модель обучения,* согласно которой всякое обучение есть установление ассоциативных связей в нейроноподобных сетях.

На смену ассоциативной модели обучения пришла *лабиринтная модель, опирающаяся* на идеи когнитивной психологии. Модель предполагает, процесс обучения состоит в эвристическом поиске в лабиринте возможных альтернаатив и оценивании движения по лабиринту на основе локальных критериев. Наиболее исследованными на сегодняшний день являются модели, относящиеся к обучению по примерам.

Формальные модели

Обучение как математическая задача может быть отнесено к классу *оптимизационных проблем поиска описаний.*

Индивидуальная оптимизационная задача L есть пятерка

$$<X_L, У_L, \rho_L, F_L, J_L>$$

где X_L и $У_L$ — множества входных и выходных записей; $\rho \subseteq X_L \rtimes Y_L$ отношение (или функция ρ: $X_L \rightarrow У_L$); F_L—множество отношений $f_L \subseteq X_L \rtimes Y_L$ для всех $f_L \in F_L$, называемых описаниями; J_L — оператор качества для F_L показывающий для каждого $f_L \in F_L$ степень его близости к ρ_L. Задача состоит в отыскании оптимального по J_L описания f_L из F_L.

Спецификация задачи часто оказывается неполной. Например, оператор качества может быть плохо формализуемым, информация об отношении ρ_L может задаваться только примерами пар (x_1, y_1), (x_2, y_2), ...» (x_n, y_n), для которых $x_i \rho y_i$, и т. д. Если спецификация полная, то обучение не нужно, так как получается традиционная оптимизационная задача. Для задач, относимых к обучению, характерна неполнота спецификации.

Множество индивидуальных оптимизационных задач *{L}* с одними и теми же компонентами *X, У, F* и *J* называется *оптимизационной проблемой.* Задача синтеза (поиска) описаний для проблемы состоит в построении алгоритма ω который по спецификации произвольной $L \in \{L\}$строит решение $\omega(L) = f^*$. При

ггом описание должно быть синтезировано с возможно меньшими вычислительными затратами, т. е. алгоритм ω должен быть в этом смысле оптимальным.

Для комбинаторных проблем задачи $L \in \{L\}$часто ранжируются по числовому параметру size(L), называемому *размером задачи* и являющемуся мерой сложности входной спецификации. Так, размером задач для проблемы нахождения кратчайшего пути в графах при условии полной спецификации может служить число вершин графа. При наличии размера у индивидуальных задач понятие оптимальности ω можно понимать стандартным образом: вводится *функция затрат* $C(\omega, L)$ и *функция сложности* $C(\omega, n) = max\{ C(\omega, L) \mid size(L) \leqq n \}$. Алгоритм ω будет оптимальным, если не существует алгоритма *ω'*, такого, что C (ω', n) ≦ C (ω, n) для всех *n* (и для одного из *n* это неравенство строгое) [Гэри др., 1982].

Для задач обучения естественное понятие размера часто отсутствует. Трудно представить, например, что могло бы служить в качестве функции size при

синтезе описаний функций, спецификации которых задаются в виде бесконечной последовательности пар <аргумент-значение>. Отсутствие размера является одной из причин того, что сложность *С* в задачах обучения обычно плохо формализуема.

Известные методы решения задачи синтеза можно классифицировать по способу спецификации проблем, типу разрешенных алгоритмов ω, классу исследуемых проблем, критерию оценки синтеза и т. д. В настоящее воемя решение задач обучения характеризуется тремя подходами [Погосян, 1983].

1. В *теории статистических гипотез* [Кендел и др., 1960] рассматривается множесво *М* реализаций некоторого случайного объекта с распределением вероятностей *p(x)* на *M*. Пусть *W* — произвольное подмножество *М* и *{Н}* — некоторое множество гипотез фиксированного типа, связанных с вероятностью *p(x)* и характеризующих ее. Требуется на основе выборки (обучающей последовательности) из *М*, полученной в соответствии с *p(x),* выбрать наиболее подходящую гипотезу из *{Н}*.

В качестве гипотез могут быть, например, следующие утверждения: «*p(x)*

пуассоновское расределение» или «распределение *p(x)* — нормальное и имеет заданные

средние и дисперсию» и т. д. Статистические методы применимы к случайным величинам и обнаруживают их специфические, статистические характеристики, которые часто являются симптоматическими по отношению к основным, глубинным закономерностям исследуемых явлений [Дружинин, 1973]. В [Гаек и др., 1984] объединяются методы статистической теории на стадии построения гипотез с логическими методами на стадии обоснования и построения следствий из этих гипотез. Другие методы теории статистических гипотез, применяемые в проблеме обучения, можно найти в [Вапник и др., 1974; Фукунага, 1979].

2. В теории *параметрической адаптации* [Цыпкин, 1968, 1970; Вапник и др., 1974] полагается, что множество *F* описаний, среди которых ищется f*, может быть охарактеризовано вектором параметров и выбор сводится к поиску экстремума оператора качества, задаваемого функционалом вида

$$\boldsymbol{J}(\boldsymbol{c}) = \int_{\boldsymbol{x}} \boldsymbol{Q}(\boldsymbol{x},\boldsymbol{c})\boldsymbol{p}(\boldsymbol{x})\boldsymbol{dx} = \boldsymbol{M}_{\boldsymbol{x}}\{\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{c}\boldsymbol{0}\}$$

Здесь $x= (x_1 ,.,, x_n)$ — вектор дискретного или непрерывного случайного процесса с плотностью распределения *p(x)*; $c=(c_1 ..c_m)$ — вектор, компоненты которого характеризуют выбранное решение (описание); *Q(x, c)* — функционал вектора *с,* зависящий от *x*; *Mx* — математическое ожидание. Экстремум *J (c)* находится из уравнения drad *J(c) =0*. Поскольку в общем виде это уравнение не имеет аналитического решения, то можно переходить к разностному уравнению

- $C[t] = c[t-1] — Г[t]\ grad_c\ J(c[t-1])$,

где *Г* — матрица *m* x *m,* элементы которой, вообще говоря, зависят от текущего значения *c [t—1]*. Надлежащий выбор матрицы *Г* должен обеспечить сходимость *c[t]* к оптимальному значению *c**.

Если *p(x)* неизвестна и ее нельзя предварительно восстановить, а также при отсутствии явно заданного функционала *J(c)* переходят к другому разностному уравнению, которое по наблюдаемым значениям *x, c* и $grad_c\ Q(x[t], c)$ позволяет определить изменение вектора *c [t]:*

- $C[t] = c[t-1] — Г[t]\ grad_c\ Q(x[t], (c[t-1])$),

В этом случае соответствующие итеративные алгоритмы называются *адаптивными* или *обучающимися.*

Алгоритмы параметрического обучения и их приложения в системах классификации, обучающихся моделях, антенных и кодирующих устройствах, фильтрах и т. д. описаны в [Айзерман и др., 1970; Цыпкин, 1970; Вапник и др., 19741.

3. *Теория индуктивного вывода* [Gold, 1967; Angluin et. al., 1983] представляет собой дискретную математическую модель обучения по примерам. Множества *X* и У счетные, искомое описание ρ в общем случае специфицируется посредством (потенциально бесконечной) последовательности троек вила (x_1, y_1, *al*), (x_2, y_2, a_2), ..., таких, что $a \in \{0, 1\}$ и $x_i \rho y_i$ тогда и только тогда, когда a=1 (т. е. тройки (x_i,y_i,a,) представляют примеры и контрпримеры р). В качестве *P* выбирается множество процедур, например, формальные грамматики, общерекурсивные функции и т. д.

6.2. Обучение по примерам

Типы задач

Система (человек или машина) может получать новые знания многими способами. Можно, например, вывести нужную информацию как логическчое следствие имеющихся знаний, получить ее модификацией имеющихся .чплиий, рассчитывая на «аналогичность» ситуаций, попытаться вывести общин закон из имеющихся примеров. Приведем некоторые задачи, традиционно относимые к задачам обучения по примерам (ОП).

1. *Прогнозирование.* Дана последовательность чисел: 3, 5, 7 ... Чему равен следующий член этой последовательности?
2. *Идентификация (синтез) функций.* Имеется некоторый «черный ящик» относительно устройства которого можно судить по его поведению, подавая на вход сигналы и получая в ответ выходные. Требуется по этой информ сформулировать описание работы анализируемого устройства. Главное отличие этой задачи от предыдущей состоит в синтезировании общего закона, а не в прогнозировании его частного проявления. К задачам идентификации относится, например, расшифровка структуры конечных автоматов [Трахтенброт и др.,1970], индуктивный синтез программ на языке ЛИСП [Biermann, 1978].
3. *Расшифровка языков.* Поиск правил синтеза текстов некоторого языка на основе анализа конкретных текстов на этом языке (расшифровки кодов, систем письменности и т. д.). Задачей такого же типа является обучение распознаванию образов.

Близкие по духу задачи рассматриваются в теории грамматического вывода [Biermann et al., 1972a].

1. *Индуктивный вывод.* В широком смысле в это направление вписываются все рассмотренные выше задачи. В узком смысле индуктивный вывод почти совпадает с проблемой расшифровки языков.
2. *Синтез с дополнительной информацией.* В качестве дополнительной информации может использоваться структура примеров, их родовидовая принадлежность и т. д. К дополнительной информации относят также контрпримеры. Контрпримеры часто помогают в решении задачи (например, в алгоритме Уинстона, решающем задачу об арках [Уинстон, 1978]). Дополнительная информация используется в других задачах синтеза. Например, возможность синтеза программ по парам вход —выход существенно расширяется, если с каждой парой вход — выход задается «траектория» ее вычисления (последовательность состояний программы без учета их тождественности) [Summers, 1977].

Все рассмотренные задачи в зависимости от предмета исследования (множество или отображение) относятся к одной из двух категорий: синтезу языков или синтезу функций. Отметим два свойства задач ОП (общие для функционального и языкового синтеза): 1) все они являются задачами нахождения описаний; 2) задаваемая в виде примеров входная информация (обучающая выборка) является недостаточной для однозначного формирования требуемого описания. В этом смысле задачи обучения некорректны. Например, при идентификации вычислимой функции по входным-выходным данным (x_1, y_1,), (x_2, y_2,), ...существует бесконечное число вычислимых функций, графики которых содержат точки (x_i, y_i,), и нет логических оснований предпочесть одну из них другой.

Итеративные алгоритмы обучения

В большинстве задач ОП множество примеров потенциально бесконечно. Его означает, что хотя в каждый конкретный момент времени имеется конечное множество примеров, но число их может увеличиваться неограниченно. Например, при расшифровке конечного автомата можно подавать на его вход новые входные сигналы, получая в ответ новые выходные. Новые данные могут помочь отсеять некоторые описания и может наступить момент, когда останется единственный вариант, который не будет опровергаться последующими примерами. Такие ситуации возможны, хотя обычно не существует алгоритма, способного определить момент стабилизации решения.

Пусть задана (потенциально бесконечная) последовательность чисел y_0, y_1... Задача состоит в нахождении полинома P, такого, что y_0, y_1... совпадает со значениями $P(x)$ при $x=0, 1, ...$ Решение задачи будем искать с помощью итеративной процедуры, работа которой на k~ом шаге состоит в обработке очередной порции примеров — последовательности $y_0, y_1 ... y_к$ и выдаче на ее основе некоторого полинома P_k (гипотеза об искомом полиноме P). В качестве $P_к$ выберем первый в фиксированном заранее перечислении всех возможных полиномов полином степени, меньшей или равной k, такой, что значения его при $x = 0, 1, ..., к$ совпадают с $y_0, y_1, ..., y_k$ (возможность перечисления полиномов и существование $P_к$ очевидны). Если пример y_{k+1}, рассматриваемый на следующем шаге итерации, будет таким, что $P(к+1) = y_{k+1}$ (т. е. гипотеза не "опровергается), то $P_{к+1}$ будет совпадать с $P_к$. Если $P_к$ и далее не опроверга- ется, то оно будет искомым полиномом. С другой стороны, поскольку последовательность $y_1, ..., y_к, ...$ «полиномиальная», то после некоторого числа опровержений подходящий полином будет найден. Таким образом, после «достаточно большого числа примеров последовательность гипотез стабилизируется» на полиноме, удовлетворяющем условию задачи. Однако определить достаточное
число примеров алгоритмически невозможно, хотя очевидно, что если вместе с примерами в качестве дополнительной информации задана степень искомого полинома P, то последняя проблема разрешима.

Существует более изящный и менее сложный интерполяционный алгоритм для решения этой задачи, однако прием с перечислением «пространства гипотез» обладает большей универсальностью. Например, задача синтеза конечных автоматов по результатам экспериментов решается аналогично [Трахтенброт и др., 1970], причем число примеров, необходимых для стабилизации гипотез, можно подсчитать, если заранее задано количество состояний искомого автомата.

Таким образом, согласно наиболее распространенной в настоящее время парадигме [Gold, 1967] корректными являются методы генерации гипотез, **ко**торые в пределе (при исчерпании всех примеров) приводят к решению задачи.

Итеративные процедуры обладают существенно большими вычислительными возможностям чем обычные алгоритмы. В теории алгоритмов эти процедуры получили название *алгоритмов предельных вычислений* , АПВ, представимых в виде lim F(x,y) при y$\rightarrow\infty$, где F- некоторая общерекурсивная функция [Фрейвалд, 1974].

Предположение о предельной стабилизации гипотез является основой гипотетико-дедуктивного подхода [Kugel, 1977], согласно которому решение задачи ОП включает четыре этапа: 1) наблюдение: сбор и накопление исходных данных (примеров), 2) обобщение: выдвижение «разумной» гипотезы *H* об искомом описании; 3) дедукцию: выдвижение различных следствий из *H* или прогнозов на основе *H* , 4) подтверждение: проверка прогнозов на совместимость с результатами наблюдений — оценка гипотезы *H*; если *H* подтверждается, то остается в качестве текущей типотезы и весь процесс повторяется сначала, в противном случае гипотеза *H* заменяется новой. Считается, что описанный процесс находит искомое описание, еслии оно было выдвинуто в качестве гипотезы на каком-либо этапе и при следующих прохождениях не заменялась на новую.

В процессе выдвижения гипоотез выясняются «разумные» способы выдвижения и критерии подтверждения гипотез.

Спецификация задач обучения по примерам

Для спецификации задачи ОП необходимо уточнить следующие ее характеристики: [Angluin et ai., 1983].

1. Класс искомых описаний. Поскольку речь идет о математической теории, го обычно это множества (языки) и функции.
2. «Пространство гипотез», т. е. множество формальных выражений, соответствующих возможным описаниям. Каждое из искомых описаний должно иметь в пространстве гипотез хотя бы одного своего представителя. Обратное неверно пространство гипотез может представлять более широкий класс описаний, чем искомые.
3. Множество примеров для каждого описания, а также разрешенные последовательности этих примеров, называемые допустимыми представлениями этого описания.

4. Критерий успешности вывода, т. е. определение того, в каком случае гипотеза, на которой слабилизировался процесс решения задачи, считается приемлемой.

Например, *задача синтеза (идентифицации) вычислимых* (т. е. частично рекурсивных функций может бьп идентифицирована следующим образом. Класс искомых описаний есть множество всех вычислимых функций; пространство гипотез –множество всех синтактически правильно построенных программ (скажем Лисп- программ); множество примеров функции f — все возможные пары натуральных натуральных чисел $<x,f(x)>$, такие что f определена в точке x. Допустимым представлением является любая бесконечная исчерпывающая последовательность примеров (повторения допускаются). Наиболее часто упоминаемый критерий успешности состоит в том, что программа P. накоторой стябилизировался процесс нахождения описания множества $\{<x, f(x) > | x \in dom\, f\}$, должна заканчивать работу во всех точках $x \in dom\ f$ и для всех этих точек выдавать значений $f(x)$ (P может быть определена и в таких точках, где f не определена). Говорят, что такая программа P идентифицирует функцию f. Другие встречающиеся критерии — точная идентификация (т. е. с учетом точек неопределенности /), идентификация с точностью до конечного числа точек и т. д.

Наиболее распространенными спецификациями задач *расшифровки языков* являются следующие. Класс описаний — множество формальных языков в фиксированном алфавите Σ; пространство гипотез — конкретный способ представления этих языков. Например, для задачи синтеза регулярных языков это могут быть регулярные выражения, конечные автоматы, леволинейные и праволинейные грамматики, а также формализмы, описывающие более широкие классы языков. Допустимым представлением языка L является любая бесконечная исчерпывающая последовательность слов из L. Такое представление языков называется позитивным. Если вместе с примерами слов из L в последовательности встречаются маркированные контрпримеры, то такое представление называется позитивно-негативным. Если в представлении присутствуют все возможные контрпримеры (т. е. последовательность включает все слова в алфавите Σ, маркированные знаками + или -), то представление называется полным. Критерием успешности является точное соответствие найденного описания множеству позитивных примеров. В отличие от синтеза функций «сверхобобщение» обычно не допускается, поскольку это приводит к тривиальным решениям. Другими критериями могут быть совпадение с точностью до заданного числа слов, совпадение с точностью до любого конечного числа слов и т. п.

Задача расшифровки языков рассматривается и в вероятностной постановке. *Стохастический язык* — это множество слов в некотором языке с распределением вероятностей на своих элементах. Пространство гипотез составляют стохостические грамматики (т. е. грамматики с приписанными правилам вероятностями). Допустимым представлением является последовательность слов из языка. Подразумевается, что эта последовательность случайная, т. е. входящие в нее слова встречаются с соответствующей им в языке вероятностью. Наиболее распространенный критерий успешности – идентификация стохастической грамматики, порождающей заданный язык с вероятностью 1. Множество описаний, пространство гипотез и множество допустимых представлений задачи обычно стандартным и естественным образом вытекают из ее неформальной постановки; вариативность критерия успешности намного больше. Укажем несколько критериев, ослабляющих требование стабилизации гипотез. Выполнение условия на окончательную гипотезу требуется лишь «для всех гипотез, начиная с некоторого места». Например при синтезе функций с критерием успешности «по поведению» требуется, чтобы, начиная с некоторого места, все гипотезы P_i, P_{i+1}, $P_{i+2}, \ldots$, идентифицировали f. Таким образом, условие $Pi = P_{i+1} = \ldots$ снимается: идентифицировать функцию становится легче. При частотной идентификации требуется, чтобы частота (точнее, ее нижний предел) «правильных» гипотез была не меньше заранее заданного числа е. Такие же критерии рассматривались в задачах языкового синтеза.

Несколько другой характер имеет отказ от условия стабилизации гипотез в *задаче прогнозирования вычислимых функций.* Здесь требуется найти не общее описание, а различные следствия из него. Соответственно пространство гипотез составляют не программы для вычислимых функций, а их результаты, т. е. множество всех натуральных чисел. Прогнозирование функции f считается успешным, если, начиная с некоторого места *i*, все выданные числа-гипотезы h_i, h_{i+1}, h_{i+2}, ... совпадают с числами f(i), F(i+1), …, заданными на вход итеративного алгоритма $H{:}H(f(0), f(1), \dots f(i+j)) = h(i+j) = f(i+j)$ для всех $j \geqq 0$.
Особенность задачи прогнозирования заключается в том, что ее формальные уточнения существенно зависят от принятой вычислительной модели итеративных алгоритмов.

Для того чтобы охарактеризовать эту зависимость и сравнить рассмотренные варианты задач языкового и функционального синтеза, достаточно ппедставить АПВ в виде трехленточной машины Тьюринга (М) со следующими особенностями. Первая (входная) лента предназначена для записи закодированных в алфавите машины бесконечных последовательностей (допустимых представлений множества примеров). Головка машины, работающая с этой лентой, является только считывающей и может перемещаться только вправо — по команде управляющего устройства. Вторая (рабочая) лента предназначена для вычислений на каждом шаге итерации. При этом входными параметрами являются очередной элемент допустимого представления, считанный со входной ленты, и записи, оставшиеся на рабочей ленте после предыдущего шага итерации — «накопленный опыт» машины. Головка рабочей ленты может и читать, и писать, может двигаться и вправо, и влево. Время от времени управляющее устройство дает команду головке третьей (выходной) ленты записать на ней некоторое число, которое интерпретируется как номер выдаваемой гипотезы в фиксированном заранее пересчете пространства гипотез. Последняя головка — только пишущая и двигается только вправо. Печать номера на выходной ленте считается завершенной, если в конце его напечатан специальный знак—(разделитель). Программа управляющего устройства должна быть такой, чтобы печать каждого начатого номера завершалась.

Интерпретация работы такой машины для задач ОП очевидна. Гипотезу, выданную машиной *M* после считывания первых *n* элементов входной последовательности $al\ a_2, \dots, a_{n>}$, будем обозначать $M\ [al, a_2, \dots, a_n]$.

Исследуем возможности таких машин. Пусть S есть некоторая спецификация задач ОП, т. е. $S = \langle D, H, P, C\rangle$, где *D* – множество описаний; *H* — пространство гипотез; *P* - множество допустимых представлений; *C* – критерий успеха. Обозначим через $Inf_s\ (M)$ множество описаний из D, которое (после соответствующих кодировок допустимых представлений и пересчета пространства гипотез) машина *M* в состоянии синтезировать при заданных *H, P* и *C.* Другими словами, $Inf_s\ (M)$ — это множество индивидуальных задач из S, решаемых машиной *M.* Таким образом, при разных $S\ Inf_s\ (M)$ может быть множеством языков, синтезируемых *M* по их полному представлению, множеством функций, идентифицируемых *M* по входным-выходным данным и т. д. Класс $Inf_s\ (M)$ является мерой универсальности *M* при решении S. Множество описаний D' называется *идентифицируемым* (синтезируемым) при спецификации S, если существует машина *M,* такая, что $D' \subseteq Inf_s\ (M)$. Отметим, что идентифицируемость того или иного множества не зависит ни от алфавитов машины *M,* ни от способа кодирования информации, ни от способа пересчета (нумерации) пространства гипотез и самого его выбора. Независимость от выбора и способа нумерации пространства гипотез нуждается в оговорках, однако достаточно, чтобы *H* было множеством, обладающим так называемой геделевой нумерацией. Это следует понимать в том смысле, что класс $Inf_s = \{D' \mid D' \subseteq Inf_s\ (M)$ для некоторого $M\}$ инвариантен к изменению этих параметров.

Для большинства задач ОП не существует универсальных способов их решения (т. е. не существует АПВ, решающего все индивидуальные задачи).

Первый результат такого рода был получен в [Gold, 1967]: любое множество языков, содержащее все конечные и хотя бы один бесконечный язык, на может быть синтезировано по позитивному представлению. Отсюда следует **не**синтезируемость по позитивным примерам множества регулярных языков. Подобные утверждения были затем доказаны для задач синтеза и прогнозировании вычислимых функций с различными критериями успешности, синтеза рекурсивных языков по полному представлению и др. [Бардзинь и др., 1972; Blum et al. 1975, Angluin 1980a; Мартиросян 1986].

Сравним различные варианты спецификации задач ОП. Будем различии, два типа машин. Машины первого типа, называемые всюду определенными, обладают тем свойством, что любой элемент входной последовательности когда нибудь считывается и между любыми последовательными считываниями машина выдает на выходную ленту одну гипотезу. Другими словами, функция $M [al, a_2, ..., a_n]$ определена для всех $al, ..., a_n$ и любого n. К машинам второго типа отнесем те, у которых $M[al ..., a_n]$ не определена хотя бы для одного набора (al ... $..a_n$). Оказалось, что если S — спецификация задачи синтеза (языков или функций), то при проверке S-синтезируемости множества описаний можно ограничиться только всюду определенными машинами. Другими словами, по любой машине M можно построить всюду определенную машину M', такую, что $Inf_s (M) \subseteq Inf_s (M')$.

Исключением из правила являются задачи прогнозирования. Рассмотрим три ее варианта, отличающихся друг от друга ограничениями на использование не всюду определенных машин. В первом варианте разрешается использовать только всюду определенные машины: $U\in Inf_{s1}$ тогда и только тогда, когда существует всюду определенная машина М, такая, что для любого $f\in U$

имеем $M[f(0), ...f(n)] = f(n+1)$

для всех n, кроме конечного числа. Во втором варианте M не обязана быть всюду определенной, но если $U\in Infs1(M)$, то, начиная с некоторого $n=n0$, должно быть $M[f(0), ...f(n)] = f(n+1)$, причем для всех $\kappa<n$ требуется, чтобы функция $M[f(0), ...f(n)]$ была определена. В третьем варианте разрешается, чтобы для некоторых κ (но не более конечного числа) $M[f(0), ...f(n)]$ была не определена.

Для классов Inf_{s1}, Inf_s и Inf_{s2} введем специальные обозначения NV, NV', NV'' (NV означает next value); имеют место соотношения $NV \subset NV' \subset NV''$, где $\subset$ - строгое включение.

Рассмотрим несколько разновидностей задачи синтеза функций. Пусть S есгь спецификация, при которой множеством описаний являются все вычислимые всюду определенные (общерекурсивные) функции, пространством гипотез — множество всех программ на каком-либо языке, единственным допустимым представлением функции f — последовательность $f(0), ...f(n)$ … Критерий успеха определим как точное угадывание: $M [f(0), ...f(n))] = const$ для всех n, начиная с некоторого n_0, причем программа Р, имеющая номер $M [f(0), ...f(n))]$, вычисляет именно функцию f. Задачу S обычно называют задачей синтеза общерекурсивных функций. Множество Inf_s обозначают через GN (от Gödel Number).

Если ослабить критерий успешности, сняв условие стабилизации, то получим другую задачу — поведенческого синтеза общерекурсивных функций. Здесь |требуется, чтобы для всех i программы, имеющие номера М [f(0), …f(n),…f(n+i)] вычисляли одну и ту же функцию f (но не требуется, чтобы все программы были одинаковые). Множество Inf_s в этом случае обозначается GN^∞.

Соотношения между рассмотренными задачами такие [Подниекс, 1974]: $NV \subset NV' \subset GN \subset GN^\infty \subset NV''$. Результаты сравнения других вариантов спецификации можно найти в [Подниекс, 1975; Klette et al., 1980; Jantke et al.., 1981; Blum et al. 1982; Angluin et al 1983; Case et al., 1983; Мартиросян, 1986] и др. Наиболее типичны случаи, когда имеет место строгое включение или когда и выдачи оказываются несравнимыми (т. е. соответствующие классы Inf_s несравнимы по теоретико-множественному включению). Например, задачи частичного прогнозирования функций всюду определенными и не всюду определенными машишами (соответствующие спецификации достаточно очевидны) несравнимы при разных заданных частотах: при $\delta<\varepsilon$ $NV(\delta)$ и $NV'(\varepsilon)$ несравнимы по включению. [Подниекс, 1974]. Отсюда, в частности, следует, что нельзя гарантировать повышение частоты верных прогнозов за счет отказа от всюду определенности машин. Тем не менее теория индуктивного вывода не разбивается на независимые подтеории: не только методы, но и многие результаты для всех этих задач совпадают.

Возможности АПВ

При исследовании зависимости АПВ типа представления примеров [Gold, 1967] оказалось, что позитивное представление языков недостаточно информа тивно для синтеза большинства типов языков. Однако в [Angluin, 1980a, b; Shinohara, 1982; Nix, 1983] приведены интересные классы языков, синтезируемых по позитивным примерам. В задаче синтеза стохастических грамматик ситуация иная: произвольный стохастический контекстно-свободный язык может быть с вероятностью 1 идентифицирован (т. е. для него будет найдена стохастическая контекстно-свободная грамматика) по случайной последовательности слов из языка, не содержащей явно негативных примеров [Horning, 1969]. Различные варианты языкового синтеза по позитивному представлению рассматривались также в [Angluin, 1980a, b, 1982; Osherson et al., 1982; Angluin et al., 1983]. Если заранее известно, что перечисления порождаются только примитивнорекурсивными функциями, то можно идентифицировать любые рекурсивные языки [Gold, 1967]. Этот же результат справедлив и для функционального синтеза [Blum et al., 1975; Мартиросян, 1986]: множество всех вычислимых функций идентифицируется по примитивно-рекурсивному перечислению своих графиков. Если допускать любые эффективные (алгоритмические) перечисления, то выигрыша не получаемся: множество вычислимых функций идентифицируется по эффективным перечислениям тогда и только тогда, когда оно идентифицируется по всем возможным представлениям. Другие возможные эффекты от ограничений подобного рода описаны в [Weihagen, 1978]. В [Blum et al., 1975] показано, что описание, на котором стабилизируется АПВ, можно сделать независимым от того, с каким из допустимых перечислений оно работает. Этот результат останется справедливым в любом случае, если класс допустимых перечислений замкнут относительно эффективных преобразований [Мартиросян, 1986].

Ограничения на процесс решения задач ОП могут вытекать как из специфики данной задачи, так и из общих для обучения требований, например *согласованности.* Машина M называется согласующей, если всякая гипотеза, выданная M, согласована с тем начальным отрезком допустимого представления, на основе которого она генерирована.

Приведем содержательные примеры, показывающие, что требование согласованности ограничивает возможности АПВ. Пусть фиксирована некоторая спецификация $S = <D, H, P, C>)$ задачи ОП и M — некоторый АПВ. Пусть $d \in D$– описание и $p_d \in P$ – допустимое представление d (для простоты предположим, что критерий успешности C включает условие стабилизации гипотез). Возможны три исхода при работе M с p_d: 1) M не стабилизируется ни на одной гипотез, 2) M стабилизируется на гипотезе, удовлетворяющей С, и 3) M *стабилизируется* на гипотезе, не удовлетворяющей C. Машину M будем называть *надежной,* если ни для одного $d \in D$ и $p_d \in P$ третья из этих возможностей не реализуется. Иначе говоря, M либо идентифицирует данное d, либо не приходит ни к какому выводу. Заметим, что надежные машины наследуют свойство независимости от порядка. Всякая согласующая машина (АПВ) надежна. Без нарушения общности верно и обратное: для любой надежной машины M можно построить «равномощную» ей согласующую машину M'. Таким образом, класс описаний, синтезируемых надежными и согласующими машинами, один и тот же.

Надежные АПВ обладают рядом замечательных свойств. Например, если $U1, U2 \in Inf_s R$, то $U1 \cup U2 \in Inf_s R$, где $Inf_s R$ – класс множеств, S-идентифицируемых надежными АПВ. Это свойство аддитивности справедливо и для бесконечного| числа слагаемых. Далее, если $U \in Inf_s R$ и — «похожее» на множество описаний (например, все описания из U' получаются применением к описаниям из U некоторой стандартной процедуры или описания из U' отличаются от описаний из U каждый раз только конечным числом примеров и т. д.), то часто удается по надежной M, идентифицирующей U, построить надежную M', идеитифиющую U'.

Другим интересным свойством надежных алгоритмов в задачах синтеза функций (точнее, для задач, в которых критерий успешности допускает «сверхобобщение данных») является то, что вместе с синтезом описания всегда обеспечить синтез любого его подописания, охватывающего более узкое множество примеров, чем исходное. Это свойство равномерности вне связи с надежностью рассматривается в [Фрейвалд и др., 1975]. Ограничение надежными АПВ имеет и методологические преимущества. Оказалось [Мартиросян, 1986], что классы множеств, идентифицируемых надежными АПВ в разных вариантах задачи функционального синтеза, часто совпадают: $Inf_s R = Inf_{s'} R$. Все приведенные результаты являются следствием требования надежности, т. е. в общем случае они неверны. Дополнительные сведения о надежности (согласованности) можно найти в [Angluin et al., 1983].

Иногда рассматривается понятие *относительной надежности* АПВ. Если D' — подмножество описаний (индивидуальных задач), то M называется D'-надежnой, если условие надежности выполнено для всех описаний $d \in D$. Например, если S есть задача синтеза вычислимых функций, T — класс всех всюду определенных вычислимых (общерекурсивных) функций, то *T-надежность* машины M означает, что по любому перечислению графика общерекурсивной функции машина M должна либо не стабилизироваться вообще, либо синтезировать в пределе алгоритм вычисления этой функции. T-надежные машины могут идентифицировагь такие множества общерекурсивных функций, которые не могут быть надежно идентифицированы. Поэтому не следует ограничиваться согласующими алгоритмами.

Еще одним часто встречающимся понятием является *консервативность* АПВ. Консервативный АПВ изменяет выдаваемую гипотезу только в том случае, если вновь полученный элемент допустимого представления не согласован с ней. Консервативность не всегда можно обеспечить: например, существует множество вычислимых функций, такое, что U идентифицируется согласующей машиной, но не идентифицируется машиной, являющейся к тому же консервативной [Kugel, 1977; Мартиросян, 1986]. Аналогичное утверждение справедливо и для языкового синтеза [Angluin, 1980a]. Другие естественные ограничения обсуждаются в [Jantke et al., 1981].

В задачи индуктивного вывода входит также характеризация возможностей различных классов АПВ, т. е. определение множеств описаний, синтезируемых для задач ОП [Weihagen, 1978; Angluin, 1980a; Zeugmann, 1983*]. Большинство таких характеризаций сложно и содержательно не интерпретируется. Остановимся на тех, которые являются исключением из этого правила. Класс NV (класс множеств общерекурсивных функций, прогнозируемых всюду определенными АПВ) совпадает с классом всех эффективно перечислимых множеств общерекурсивных функций [Бардзинь и др., 1972]. Класс общерекурсивных функций, идентифицируемых согласующими и одновременно" консервативными АПВ, совпадает с NV [Мартиросян, 1986].

Дополнительная информация о решаемой задаче часто имеет существенное значение. Например, любую вычислимую функцию можно идентифицировать,
Если известно, что ее допустимое представление осуществляется примитивнорекурсивной функцией [Gold 1967; Blum 1975]. В качестве дополнительной информации может служить и родовидовая принадлежность индивидуальной задачи. В
[Jantke, 1978] показано, что можно таким образом подобрать систему вычислимых функций $U_1, U_2, \ldots U_n, \ldots,$ что любая общерекурсивная функция f может быть идентифицирована по своему графику и подходящему n такому, что $f \in U_n$ Дальнейшее исследование синтеза с дополнительной информацией можно найти в [Freivald et al., 1979].

При создании эффективных процедур или методов необходимо исследовать вопросы сложности решения задач ОП. Если $S = \langle D, H, P, C \rangle$ —некоторая система задач ОП, s — индивидуальная задача из S (задаваемая фикцией какого-либо описания d), то обозначим через $CP(M, p_d)$ количество данных, требуемых машине M при работе с представлением p_d задачи с до момента стабилизации гипотез (CP — от Convergence point). Будем говорить, что M_1 эффективнее по данным, чем M_2, если $Inf_s(M_2) \subseteq Inf_s(M_1)$ и $CP(M, p_d) \leq CP(M2, p_d)$ для любого преставления p_d индивидуальной задачи s из $Inf_s(M_2)$. В [Gold, 1967] показано, что метод перечисления пространства гипотез оптимален по этому критерию.

Желательно иметь понятие сложности, зависящее только от конкретной задачи, а не от ее представления, полагая, например, $CP(M, s) = CP(M, s) = max\{CP(M, p_d) | p_d \in P\}$. Но при таком подходе в большинстве случаев $CP(M, s) = \infty$. Этот же недостаток присущ многим другим определениям сложности, имеющимся в литературе. К их числу относятся количество различных гипотез, выданных при работе с p_d до момента стабилизации, количество изменений гипотез до этого же момента и т. п. [Бардзинь и др., 1972, 1974; Фрейвалд, 1975; Кинбер, 1977; Feldman et al., 1977]. В [Фрейвалд, 1975; Daley, 1977; Daley et al., 1983] предложен аксиоматический подход к определению сложности задач ОП. Большинство результатов по оценке сложности носят негативный характер: алгоритмов, которые были бы существенно лучше переборного, не существует [Gold, 1967].

Случаи, когда эффективные АПВ можно построить, чаще всего оказываются неинтересными с практической точки зрения. В теории индуктивного вывода особое внимание уделяется изучению предельного поведения синтезирующих алгоритмов, хотя в большинстве задач имеющееся реально множество примеров всегда ограничено. Такая ситуация способствует разрыву между теорией и практикой построения обучающихся систем.

---

* Zeugmann Th. Aposteriori characterizations in inductive inference of recursive functions// Elektron. Informations-verarb. Kybern. (EIK). — 1983. P. 559—594.

6.3. Обучающиеся системы

Общие методы выдвижения гипотез

Способ выдвижения гипотез тесно связан с умением сравнивать гипотезы: лучшей считается гипотеза, которая «проще» и «более совместима с исходными данными», чем другие. Содержание, которое в каждом конкретном случае вкладывается в эти понятия, может меняться. Можно считать, что заданы два отношения и β, называемые отношениями предпочтения и совместимости, такие, что

$$h_1 \ \alpha \ h_2 \approx h_1 \text{ проще } h_2$$

$$h \ \beta \ V \approx h \text{ совместима с обучающей выборкой } V.$$

Тогда естественно полагать, что гипотеза h_1 лучше гипотезы h_2 при выборке V, если $h_1 \ \alpha \ h_2$ и h β V. Например, в задачах синтеза языков часто применяются следующие отношения:

$h_1 \ \alpha \ h_2 \leftrightarrow L(h_1) \subseteq L(h_2)$; $h \ \beta \ V \leftrightarrow V^+ \subseteq L(h)$ и , $V^- \cap L(h) = \emptyset$, где V^+ и V^- — множества позитивных и негативных примеров из V, $L(h)$ - язык, определяемый гипотезой *h.*

Если для любых h_1 и h_2 из $h_1 \ \alpha \ h_2$ следует, что h_1 лучше h_2 при любой выборке *V, с* которой h_1 и h_2 совместимы, то отношение предпочтения α называется независимым от выборки. Для задач синтеза функций независимым от выборки отношением предпочтения является, например, упорядочение программ по их длине.

Если независимое от выборки отношение предпочтения вычислимо, можно предложить простую схему АПВ (Р), применяемую во многих приложенииях:

Шаг 0. Вычислить наилучшую при данной выборке гипотезу. Потребовать новый пример. Перейти к шагу 1.

Шаг *n* (*n* >0). Если текущая гипотеза совместима с примерами, потребовать новый пример и перейти к шагу *n*+1. В противном случае перейти к шагу 0.

Эта схема будет сходиться к наилучшему описанию, если таковое действительно существует (в смысле отношения α). Отметим, что и промежуточные гипотезы будут наилучшими для каждой данной выборки описаниями.

Рассмотрим задачу синтеза конечных автоматов по позитивным и негативным примерам. Будем говорить, что автомат A_1 лучше автомата A_2, если число состояний у A_1 меньше, чем у A_2. Ясно, что нахождение наилучшего автомата возможно: упорядочив все автоматы по количеству состояний, выберем первый такой, который принимает данное множество позитивных и отвергает данное множество негативных примеров. Применение схемы Р приводит к обычной схеме «идентификации по перечислению пространства гипотез», что позволяет синтезировать конечный автомат для любого регулярного языка [Gold, 1978]. Отметим, что задача нахождения автомата с минимальным числом состояний, совместимого с данным множеством примеров, NP-полна.

Аналогичное отношение предпочтения можно сформулировать для задачи синтеза языков по позитивному представлению: гипотеза $h_V \in H$ считается наилучшей для обучающей выборки *V*, если $V \subseteq L(h_V)$ и для всех $h \in H$, таких, что $V \subseteq L(h)$, выполнено $L(h_V) \subseteq L(h))$, где $L(X)$ обозначает язык, определяемый гипотезой *X.* Такое отношение предпочтения является независимым от выборки и даже вычислимым для многих случаев. Для некоторых из них найдены полиноминальные алгоритмы для вычислений h_V [Angluin, 1980b, 1982b; Shinohara, 1982].

Рассмотрим задачу синтеза конечных автоматов по позитивным и негативным примерам. Будем говорить, что автомат A_1 лучше автомата A_2, если число состояний у A_1 меньше, чем у A_2. Ясно, что нахождение наилучшего автомата возможно: упорядочив все автоматы по количеству состояний, выберем первый такой, который принимает данное множество позитивных и отвергает данное множество негативных примеров. Применение схемы Р приводит к обычной схеме «идентификации по перечислению пространства гипотез», что позволяет синтезировать конечный автомат для любого регулярного языка [Gold, 1978]. Отметим, что задача нахождения автомата с минимальным числом состояний, совместимого с данным множеством примеров, NP-полна.

Аналогичное отношение предпочтения можно сформулировать для задачи синтеза языков по позитивному представлению: гипотеза $h_V \in H$ считается наилучшей для обучающей выборки *V*, если $V \subseteq L(h_V)$ и для всех $h \in H$, таких, что $V \subseteq L(h)$, выполнено $L(h_V) \subseteq L(h)$, где $L(X)$ обозначает язык, определяемый гипотезой *X*. Такое отношение предпочтения является независимым от выборки и даже вычислимым для многих случаев. Для некоторых из них найдены полиноминальные алгоритмы для вычислений h_V [Angluin, 1980b, 1982b; Shinohara, 1982].

Большое распространение получили отношения предпочтения, основанные на использовании теоремы Байеса. Пусть $Pr(h)$ есть вероятность появления гипотезы h в (вероятностном) пространстве H; $\Pr(V)$ вероятность появления выборки V в пространстве примеров; $Pr(V/h)$ — вероятность того, что h совместимо с V. Тогда вероятность

$$Pr(h/V) = [Pr(h)\ Pr(V/h)] / [Pr(V)]$$

может быть использована для определения отношения α: $h_1 \alpha h_2$ тогда и только тогда, когда $Pr(h_1/V) \geq Pr(h_2/V)$. Задача состоит в максимизации $\Pr(h/V)$ или, что то же, максимизации $Pr(h)\ Pr(V/h)$. Ясно, что это отношение предпочтения и является независимым от выборки и схема F не работает.

Конкретные методы подсчета вероятноегей зависят от предметной области [Horning, 1969; 1969; Вапник и др., 1974; Фукунага, 1979].

Связь отношения предпочтения с теоремой Байеса обсуждается в [Solomonoff, 1964]. Более подробный анализ отношений предпочтения можно найти [Biermann et al., 1972a, b; Angluin et al., 1983].

При поиске метода решения задач ОП большое значение может иметь структурированность пространства гипотез. Отношение предпочтения можно рассматривать как один из способов структуризации. В противоположность простому перечислению, структурированность позволяет исключить не одну, а группу ошибочных гипотез.

Рассмотрим структуру пространства гипотез, часто применяющуюся в системах формирования понятий и индуктивного вывода. На множестве всех конъюнкций атомарных формул введем отношение предпочтения, определяемое следующим образом: K_1 α K_2 тогда и только тогда, когда для некоторой подстановки σ множество атомов из $\sigma(K_1)$ есть подмножество множества атомов из $\sigma(K_2)$.

Будем говорить в этом случае, что K_1 «общее» K_2 (например, $A\ (x)\ \&B\ (x)\ \&C(y)$ общее $A(a)\&B(a)\&C(f(b)))\&D(g(y)))$. Задача обучения (формирования понятия) заключается в нахождении минимальной общей (по α) конъюнкции для заданного множества фактов, представленных атомарными формулами, не содержащими переменных. Если некоторая конъюнкция не согласована с каким-либо фактом, то и все менее общие конъюнкции можно исключить (при наличии отрицательных примеров можно исключать и более общие конъюнкции). Такого рода пространства гипотез, и более сложные, теоретически исследовались в [Plotkin 1970, 1971; Reynolds, 1970]. Структуры, основанные на этой простой идее, используются во многих алгоритмах формирования понятий [Hayes-Roth et al., 1978; Vere, 1980; Dietterich et al., 1981].

Структурированность пространства гипотез позволяет вводить в обучающую выборку специальные диагностические компоненты, анализирующие причины, по которым гипотеза оказалась несовместимой с данными. Результатом работы диагностических компонентов должно стать выявление структурных единиц гипотезы, обусловивших ее несостоятельность.

В [Shapiro, 1981] эта идея применяется к задаче индуктивного синтеза теорий по фактам. Пространство гипотез считается состоящим из всех возможных конечных множеств аксиом A_1 & ... $A_k \rightarrow A_0$ (A_i — атомарные формулы или их отрицания), т. е. фактически из всех возможных Пролог-программ; обучающая выборка состоит из атомарных формул (без переменных), помеченных символами И («истина») или Л («ложь»). Задача заключается в нахождении теории (Пролог-программы), в которой все позитивные примеры (формулы, помеченные символом И) были бы выводимы, а все негативные (помеченные Л) — невыводимы. Если текущая гипотеза оказалась несовместимой с фактами, то диагностическая процедура в интеракивном режиме выясняет, какая из аксиом, входящих в гипотезу, может быть причиной неудачи. Пролог-программы, содержащие эту аксиому, исключаются из дальнейшего рассмотрения.

Наибольшее число практических разработок основано на методах выдвижения гипотез, использующих идею локальной оптимальности гипотез. Представление о задачах обучения как об оптимизационных задачах характерно для многих подходов к проблеме обучения и в особенности для стохастических подходов с аналитическими функционалами качества [Цыпкин, 1970; Вапник и др., 1974]. Суть метода состоит в том, что выдвинутая гипотеза остается неизменной, если по мере поступления новых данных она продолжает оставаться оптимальной в своей окрестности. В противном случае она заменяется наилучшей в своей окрестности гипотезой.

В [Cook et al., 1976] можно найти пример такого «подъема на холм» для дискретной стохастической задачи — идентификации стохастических контекстно-свободных грамматик по случайным позитивным данным. Вводится комбинаторная мера $M[G, V]$, отображающая сложность грамматики G и одновременно степень ее совместимости с обучающей выборкой V. К текущей гипотезе можно применять определенные трансформации. Грамматики, получающиеся однократным применением какой-либо трансформации к текущей гипотезе, образуют ее окрестность. Если некоторая грамматика из этой окрестности оказалась лучше по мере M, она выбирается в качестве очередной гипотезы. В качестве первой гипотезы выбирается грамматика, порождающая в точности заданную обучающую выборку V (этой грамматике приписывается большая сложность и большая степень совместимости с V).

Рассмотренные до сих пор методы не всегда эффективны, поскольку ориентированы прежде всего на то, чтобы не пропускать в процессе перебора гипотез, совместимых с обучающей выборкой. Методы другого типа основаны на процедурах, которые исходя из обучающей выборки конструируют те свойства, которым должна удовлетворять разумная гипотеза. Такая идея использована, например, в эвристическом методе синтеза конечных автоматов по входным-выходным данным реализуемых ими функций [Biermann et al., 1972b]. Из всех возможных разбиений множества состояний канонического древесного автомата выбирается то, при котором в один блок попадают состояния, неразличимые для входных слов длины не более k (параметр k задается пользователем). Получаемый в результате склейки k -неразличимых состояний автомат может оказаться недетерминированным, однако при увеличении числа примеров и параметра k в пределе будет получен корректный детерминированный автомат. Этот метод был затем обобщен для синтеза грамматик [Brayer et al., 1977; Levine, 1982].

Обучение распознаванию образов

Вопросы обучения по примерам сравнительно хорошо исследованы для проблемы поиска алгоритмов классификации (ПАК). Пусть в проблеме ПАК X_L –конечное множество (универсум), $n = |X_L|$, $Y_L = \{0, 1\}$, $T=\{(X_1,X_0) \mid X_1, X_0 \subseteq X_L \ \&\ X_1 \cap X_0 = \emptyset\}$ — пары множеств (ПМ) из X_L (частично определенные предикаты на X_L); $T^n = \{(X_1,X_0) \mid X_1, X_0 \subseteq T \ \&\ X_1 \cup X_0 = X_L\}$ — абсолютные ПМ (предикаты на X_L), $T' \subseteq T^n$; ρ_L — конкретный предикат из T', известный только оракулу (или учителю); Q_L — заданный частично определенный предикат из T, связанный с ρ_L, (дополнительная информация); F_L — множество алгоритмов f: $X_L \to Y_L$, предназначенных для моделирования ρ_L и именуемых алгоритмами классификации X_L; J_L — оператор качества алгоритмов из F_L, например J_L = $<J_0,\ T_0,\ S_0>$, где $J_0(f) = \sum_{x\in X_L}\big(f(x) - \rho_L(x)\big)^2$ — хеммингово расстояние классификации, полученной $f \in f_L$ от истинной ρ_L; $T_0(x)$ — максимальное по всем $x \in X_L$ время вычисления значений $f(x)$ (время распознавания); $S_0(f)$—память, используемая при хранении f в машине. Требуется построить оптимальный по J_L алгоритм классификации f_{L^*} из F_L.

Класс A = {ПАК} определяется множеством всех конечных множеств X_L. Если f_{L^*} может быть построена без обращения к учителю с использованием лишь свойства Q_L, то проблема ПАК называется инициально полной, в противном случае — инициально пустой.

Синтез классификаций по примерам для инициально пустых ПАК рассматривается в основном в теории распознавания образов и называется обучением распознаванию. В теориях статистических гипотез и параметрической адаптации обучение распознаванию формулируется как разновидность проблемы минимизации среднего риска [Вапник, 1974], получаемый из проблемы ПАК при следующих дополнительных требованиях:

1) X_L есть множество n-мерных векторов с распределением вероятностей $P(x)$; каждый вектор представляет результат измерений множества объектов n датчиками с конечной и дискретной шкалой измерений;

2) компоненты произвольной абсолютной пары множеств $V = (X_1,\ X_0)$ на X_L являются реализациями случайных величин B_1^V и B_0^V соответственно;

3) на множестве X_L существуют совместные распределения вероятностей $p(\omega, x)$, $\omega = 0, 1$, которые задают вероятности событий типа: появился вектор $x \in X_L$ и $x \in X_1$ ($x \in X_0$); $p(\omega,\ x)$ характеризует множества элементов из X_L, о принадлежности которых будет задан вопрос, учителю; при этом $p(\omega, x) = p(x)p(\omega|x)$;

4) каждая из рассматриваемых абсолютных пар множеств задается характеристической функцией $F(x, c)$ при некотором c, где $x \in X_L$, c — m-мерный вектор параметров;

5) оператор качества решений задается в виде

$$J(c) = \int \big(\omega - F(x,c)\big)^2 dp(\omega, x) \qquad (1)$$

Требуется найти минимум по c функционала $J(c)$, если неизвестна функция распределения $p(\omega, x)$, но задана случайная и независимая выборка $x_1\omega_1$, $x_2\omega_2$, ..., $x_L\omega_L$ — обучающая последовательность.

Так, для персептрона оператор качества решений имеет вид

$$J(c) = \int \left(\omega - \theta \sum_{i=1}^{m} c_i \varphi_i(x) \right)^2 dp(\omega, x),$$

Где $\theta(x)$ – пороговая функция, $\varphi_i(x)$ — некоторые преобразования на X_L (А-элементы).

Существуют три пути решения задачи (1). Первый относится к теории статистических гипотез и связан с идеей восстановления функций распределения $p(\omega, x)$ векторов первого и второго класса, а затем построением на их основе дискриминантной функции. Поскольку построение функций распределения является более сложной задачей, чем построение искомой дискриминантной функции данный путь часто бывает нерационален.

Второй путь связан с организацией рекуррентной процедуры поиска параметра c, дающего минимум (1). Исходя из процедуры $c[t]$ для (1) имеем

$$c[t] = c[t-1] + \Gamma(t)\, grad_c\, \xi(\omega(t) - F(x(t), c[t-1])). \qquad (2)$$

Выбирая различные выпуклые функции получаем ряд известных алгоритмов [Цыпкин, 1968, 1970; Вапник и др., 1985*], в частности обучения трехслойных персептронов и метода потенциальных функций [Айзерман и др., 1970]. В общем случае решение в соответствии с (2) гарантируется при неограниченном увеличении обучающей выборки. Достаточные условия сходимости (2) при конечных выборках и оценки длин этих выборок приведены в [Вапник и др., 1974].

Третий путь решения (1) относится (как и второй) к теории параметрической адаптации и связан с идеей замены (1) так называемой функцией эмпирического риска

$$J_{\text{эмп}}(c) = \frac{1}{l} \sum_{i=1}^{l} (\omega_i - F(\chi_i, c))^2, \qquad (3)$$

построенной по случайной независимой выборке x_1, ..., x_l и определяющей частоту неправильных классификаций на этой выборке. Метод состоит в поиске $c_{\text{эмп}}$, обеспечивающего минимум эмпирического риска. Теория призвана выявить условия, при которых переход от (1) к (3) возможен. Основой их является требование

$$p\{sup_c |J_{\text{эмп}}(c) - J(c)| > \varepsilon\} \underset{l\to\infty}{\longrightarrow} 0 \qquad (4)$$

равномерной сходимости частот $J_{\text{эмп}}$ к вероятностям $J(c)$ по классу S событий, каждое из которых определяется параметром c и указывает на множество ошибочно классифицированных векторов x при решающем правиле $F(x, c)$.

Необходимым и достаточным условием выполнения (4) является стремление к нулю последовательности

$$\frac{H^S(1)}{1}, \frac{H^S(2)}{2}, \dots, \frac{H^S(l)}{l}, \dots$$

при неограниченном увеличении длины выборки l. Здесь $H^S(l)$ —энтропия класса S решающих правил $F(x, c)$, определяет математическое ожидание логарифма от числа возможных разбиений $\Delta^S(x_1, ..., x_l)$ выборки длины l в классе S. На практике вместо энтропии класса используется логарифм функции

$$m^S(l) = \max_{x_1,\dots,x_l} \Delta^S(x_1, \dots, x_l)$$

называемой функцией роста класса S, и соответствующее достаточное условие равномерной сходимости

$$\lim_{l\to\infty} \frac{lg_2 m^S(l)}{l} = 0 \qquad (5)$$

В [Вапник и др., 1974] доказано, что функция роста либо тождественно равна 2^l, либо мажорируется степенной функцией $1.5\frac{l^{n-1}}{(n-1)!}$, где n – минимальное число, при котором нарушается равенство $m^S(l) = 2^l$. При этом во втором случае имеет место (5). Тем самым число m-1 — мера разнообразия или емкость класса решающих правил — определяет достаточные условия выполнения (4). В этом случае говорят об ограниченной, или конечной, емкости класса правил.

* В списке литературы неправильно указан 1984 г.

Таким образом, параметрическое обучение будет успешным, если емкость класса, в котором ищется решение, конечна, а само решение доставляет минимум эмпирического риска, т. е. по возможности с меньшим количеством ошибок разделяет элементы обучающей выборки разных классов.

Из разнообразных методов минимизации эмпирического риска чаще других используются два.

Метод обобщенного портрета [Вапник и др., 1974]. Строится разделяющая гиперплоскость, минимизирующая число ошибок обучающей выборки. Если это число не равно нулю, строится вторая гиперплоскость, минимизирующая остаточные ошибки. Если ошибок еще много, то строится еще одна гиперплоскость и т. д. Алгоритмы этого, типа реализуют разные идеи кусочно-линейной аппроксимации.

Алгоритмы частичной прецедентности [Журавлев, 1978]. Для случая двух классов и *n*-мерного булева пространства E_n, все векторы которого считаются допустимыми, этот алгоритм задается с помощью следующих процедур:

F_1 — разбиения произвольной пары множеств (X_1, X_0) на обучающую (X'_1, X'_0) и контрольную (X''_1, X''_0), такие, что $X_1 = X'_1 \cup X''_1$ и $X_0 = X'_0 \cup X''_0$;

F_2 — выбора опорных множеств, указывающих совокупности признаков для анализа векторов;

F_3 — вычисления функций близости на множестве векторов;

F_4, F_5, F_6 — вычисления оценок (при сравнении вектора с вектором и вектора с классом векторов при заданных опорном множестве или классе опорных множеств);

F_7 — определения решающего правила;

F_8 — оценки оптимальности построенных распознающих алгоритмов.

Процедуры F_1— F_8 перерабатывают произвольную заданную пару множеств в распознающий алгоритм в такой последовательности:

1. Процедура F_1 разбивает заданную пару множеств (X_1, X_0) на (X'_1, X'_0) и (X''_1, X''_0).

2. Процедуры F_2 — F_7 по обучающей паре множеств (X'_1, X'_0) задают параметрическое множество распознающих алгоритмов $\{R\}$, каждый из которых определяется однозначно фиксацией конкретных значений конечного числа параметров. Будем считать, что области изменения параметров также конечны. Произвольный алгоритм распознавания R, полученный указанным образом, каждый булевый вектор S из E_n сопоставляет с одним из чисел: 0, 1, 2. При этом, если

$$R(S) = \begin{cases} 1, \text{то } S \text{ принадлежит } I \text{ классу} \\ 2, \text{то } S \text{ принадлежит } II \text{ классу} \\ 0, \text{то } S \text{ не распознано} \end{cases}$$

Таким образом, каждому алгоритму R соответствует пара множеств

$$(X_1^R, X_0^R), \text{такая, что } S \in X_1^R \leftrightarrow R(S) = 1, S \in X_0^R \leftrightarrow R(S) = 2, S \in E_n \backslash X_1^R \cup X_0^R \leftrightarrow R(S) = 0.$$

3. Процедура F_8 выбирает такой алгоритм R^* из $\{R\}$, при котором достигается минимальное число ошибок на контрольной паре множеств (X''_1, X''_0). Алгоритм R^* объявляется экстремальным.

Представление проблемы обучения распознаванию как проблемы минимизации среднего риска предполагает пригодность статистической параметрической модели для исследования проблемы распознавания, в частности возможность представления компонентов искомой пары множеств в языке статистической теории (т.е. через понятия о среднем, дисперсии, корреляции, типе распределения и т.п.), что не всегда выполняется.

Например, допущение о том, что X1 может быть описано через указание векторов средних значений и их дисперсий, практически означает следующее. Выбрана система n датчиков (измерителей, предикатов), которые преобразуют реализации некоего случайного объекта B_1 в множество векторов X_1 таким образом, что средние значения по каждой компоненте этих векторов описывают особенности B_1 достаточно существенные, чтобы на их основе, также по их отклонениям от средних (дисперсии) отделять B_1 от B_0. Другими словами, датчики должны быть выбраны для измерения значений тех признаков случайного объекта B1, которые существенны для отделения B_1 от B_0, т. е. предполагается знание существенных для описания B1 признаков. Однако известно, что выбор существенных признаков, по которым B_1 и B_0 могут быть разделены, является трудной проблемой, сравнимой с проблемой распознавания.

Если система существенных признаков задана, то статистические методы способны лишь устранить возможные случайные отклонения от средних, измеряемых по этим признакам, значений, т. е. устранить помехи при условии, что измерители способны оперировать сигналом.

Поскольку статистический подход в проблемах распознавания часто связывается с восстановлением функции распределения вероятностей и, как правило, нормального распределения, которое определяется заданием среднего значения и дисперсии, то вышесказанное характеризует требования, необходимые для успешного применения этого подхода: 1) распознаванию подлежат множества реализаций случайных объектов B_1 и B_0; 2) задана система п измерителей, которые преобразуют множества реализаций случайных объектов B_1 и B_0 в множества n-мерных векторов X_1 и X_0 соответственно; 3) измерители таковы, что анализ средних значений и дисперсий полученных измерений в совокупности достаточен для отделения B_1 от B_0, т. е. измеряются «существенные» характеристики B_1 и B_0. Эти требования, как и требования параметрического представления класса распознающих алгоритмов, во многих проблемах распознавания невыполнимы, и проблема задается в своем исходном комбинаторном представлении в форме ПАК.

Комбинаторность проблем выражается, в частности, в следующих особенностях.

1. Высокая степень индивидуальности, различие как проблем из A, так и элементов из X_L при L ∈ A. Вследствие этого построение решения f^*_L и вычисление $f^*_L(x)$ при $x \in X_L$ связано с длительным процессом изучения и приспособления к специфике L и x, требующим дополнительной информации о них извне.

2. Оператор J может быть плохо формализуем или трудновычислим, вследствие чего непосредственное использование известных методов оптимизации, основанных на аналитических свойствах, становится невозможным.

Для комбинаторных задач распознавания известны логические, лингвистические, структурные методы, обучение понятиям, методы формирования понятий Ханта [Хант, 1970], методы растущих пирамидальных сетей [Гладун, 1970] и др. Отметим алгоритм «Кора» [Бонгард, 1967] формирования понятий в виде дизъюнкции элементарных разделяющих функций. Для построения последних из множества заданных признаков выделяются достаточные признаки для элементов обучающей выборки каждого класса, которые выполняются только на элементах этого класса и в совокупности покрывают весь этот класс. При опознании нового элемента подсчитывается число выполненных достаточных признаков каждого класса и решение принимается по большинству из них. В [Вапник и др., 1974] показано, что если значение эмпирического риска близко к нулю, то вероятность неправильной классификации с помощью алгоритма «Кора» уклонится от эмпирической оценки не более чем на

$$\varepsilon = (6t \ln n - \ln \eta) / l,$$

где n — размерность пространства векторов выборки; t — число достаточных признаков; η — верхняя оценка (4), $0 < \eta < 1$.

К особенностям логических методов обучения или индуктивной логики относятся использование малого числа примеров для формирования понятий обобщение не только по признакам, но и по структуре и именам [Поспелов Д., 1986]. Один из них, основанный на принципах индуктивных рассуждений Бэкона-Милля, развивается в [Финн и др., 1981]. Логические методы формирования понятий являются основными для систем, основанных на знаниях. Они широко применяются в ситуационном управлении [Поспелов Д., 1981, 1986] и при моделировании шахматной игры [Адельсон-Вельский и др., 1978; Погосян, 1983].

Синтез классификаций для инициально полных ПАК исследуется в терминах комбинаторной оптимизации (распознавание слов с заданными свойствами) [Гэри и др., 1982]. Если итеративность в распознавании образов обусловлена, как правило, ограниченностью доступной для анализа информации, то при комбинаторной оптимизации — ограниченностью вычислительных средств и требованием быстрейшей окупаемости затрат при построении гипотез. При конкретных всюду определенных предикатах Q_L получаем инициально полные проблемы ПАК, эквивалентные ограниченным проблемам распознавания слов со свойством Q_L [Гэри и др., 1982]. В частности, при X_L — множестве n-вершинных графов, $Q_L(G)$ — предикате «граф G имеет гамильтонов цикл» получаем ограниченную проблему L распознавания гамильтонова цикла в графах с числом вершин не более n. Инициально полные проблемы формирования образов ситуаций в конечных играх типа шахмат сформулированы в [Погосян, 1983]. Ряд задач по построению классификаций в ситуационном управлении при заранее заданных критериях качества управления также относятся к инициально полным [Поспелов Д., 1986].

Формирование понятий

Описания в процессе обучения формулируются обычно в виде языковых текстов, интерпретируемых в данной проблемной области, а не в терминах, скажем, разделяющих гиперплоскостей, содержательная интерпретация которых условна. Задачи такого рода будем называть з а д а ч а м и с и н т е з а (ф о р м и р о в а н и я) п о н я т и й. Неформально их определить можно следующим образом [Michalski, 1983].

Дано множество наблюдений (фактов) F, представляющих специфические знания о некоторых объектах, ситуациях, процессах и т. д.; множество исходных предположений, выражающих ограничения, которым должно подчиняться искомое описание; контекстная информация, включающая предположения, налагаемые на F и F', конструкции, рекомендованные для генерации кандидатов на роль результирующего описания, критерий успешности, характеризующий обязательные для результирующего описания свойства, а также любые другие знания, релевантные предметной области. Предполагается, что языки, в которых вписывается данная информация, фиксированы. Требуется сформировать удовлетворяющее критерию успешности описание H, такое, что все факты из F выводятся в качестве следствий из H: $H \models F$. Заметим, что отношение $\models$ не обязательно совпадает с отношением следования в исчислении предикатов и может задаваться как контекстная информация. Критерий успешности для задач формирования понятий включает ряд трудноформализуемых семантических и прагматических условий, налагаемых на H, таких как, интересность, новизна, наличие практических следствий и т. д. При этом классификационный аспект H - это разбиение объектов на два класса — принадлежащих и не принадлежащих H может оказаться не самым важным. Например, при формировании математического понятия АРИФМЕТИЧЕСКАЯ ОПЕРАЦИЯ в описываемой далее программе АМ [Lenat, 1976] важнее, к каким следствиям оно может привести, а не какие объекты следует считать операциями, а какие — нет. Таким образом, понятие задачи формирования понятий шире, чем задачи обучения распознавания образов.

Различают два типа задач формирования понятий: *обучение по примерам* (называемое также *обучением с учителем*) и *обучение на основе наблюдения* [Michalski, 1983]. При обучении по примерам множество F представляет собой набор практически однородных описаний объектов (примеров и контрпримеров формируемого понятия): $F=\{e_i^+\}\cup\{e_i^-\}$. Требование «выводимости» $H \models F$ приводит к тому, что должны выполняться условия: $\forall i(H(e_i^+))$ (полнота); $\forall i\ (H)=(e_i^+))$ (совместимость).

При обучении на основе наблюдения факты из множества F могут иметь разную структуру, условия полноты и совместимости не выполняются и целью индуктивного предложения H является только описание («объяснение») фактов из F. Такая заадача сложнее предыдущей. Примерами обучения на основе наблюдения является большинство задач, решаемых экспертными системами, поскольку экспертные решения часто основываются на обобщении или «объяснении» имеющихся данных (формирование теорий для множества химических соединений [Buchanan et al., 1978] или чисел [Lenat, 1976, 1984; Lenat et ah, 1978], обнаружения закономерностей в данных [Soloway et al., 1977; Hayes-Roth et al., 1978; Pocorny, 1980; Гаек и др., 1984], выявление естественной классификации объектов [Michalski et al., 1981]).

При условии неполноты исходной информации центральную роль в решении задач формирования понятий должны играть процедуры обобщения (индукции). Примерами такого рода являются процедуры получения решающих правил для шахматных эндшпилей и миттельшпильных комбинаций [Michalski et al., 1977; Wilkins, 1982].

Рассмотрим обучающуюся систему [Уинстон, 1978], показывающую, как с помощью различных эвристик можно решать и задачу обучения на основе наблюдения, и задачу обучения по примерам. Эта система представляет собой четырехступенчатую совокупность процедур, позволяющих роботу ориентироваться в мире трехмерных геометрических тел (кубиков).

Первой ступенью иерархии программ служит подсистема анализа двухмерных изображений сцен, составленных из кубиков. Целью подсистемы является распознавание геометрических тел, составляющих сцену, и ее описание в терминах базовых свойств и отношений, таких, как КЛИН(x) (т. е. «x представляет собой объект, имеющий форму клина»), СЛЕВА (x, y) («x слева от y»), СПРАВА (x, y), НАД (x, y), ПОДДЕРЖИВАЕТСЯ (x, y), ПЕРЕД(x, y), СОВМЕЩАЕТСЯ (x, y) и т. д. Полученное описание представляется в виде сети, вершины которой соответствуют объектам и их свойствам, а дуги (называемые указателями) — отношениям. Задача этого этапа укладывается в рамки модели обучения на основе наблюдения.

Программы второй ступени производят разбиение множества объектов на группы, с которыми можно обращаться как с самостоятельными объектами. Эта «естественная классификация» производится без учителя на основе поиска объектов, связанных цепочками указателей или находящихся в одинаковых отношениях к некоторому другому объекту, связывающему члены потенциальной группы.

Программы третьей ступени предназначены для формирования с помощью учителя новых понятий, имеющих более сложную природу, чем «естественный классификации». Центральной идеей здесь является использование контрпримеров.

Программы четвертой ступени позволяют роботу на основе накопленных знаний планировать целесообразное поведение в мире кубиков.

Рассмотрим механизм «естественной классификации» объектов. Первыми кандидатами в группу являются объекты, связанные друг с другом цепочкой указателей типа ПОДДЕРЖИВАЕТСЯ и ПЕРЕД. Не все такие цепочки порождают группу. Например, если некоторая часть сцены представляет собой композицию, состоящую из высокой башенки (четыре больших кубика) с широкой площадкой наверху и второй маленькой башенкой (три маленьких кубика) на этой площадке, то несмотря на то, что все элементы сцены связаны отношением ПОДДЕРЖИВАЕТСЯ, система выделит две группы (большую и маленькую башни), а не одну. Это объясняется различием в размерах объектов (система располагает указателями БОЛЬШОЙ, МАЛЫЙ). Таким образом, система сначала выдвигает гипотезу о группировании объектов, а потом проверяет ее.

Другой способ выдвижения гипотезы основан на наблюдении, что некоторое число объектов одинаково относятся к другому объекту. Ножки стола типичная группа, образованная отчасти из-за того, что они одинаково относятся к крышке стола, и отчасти из-за того, что все они являются разновидностями брусков и все стоят.

Проверка гипотезы производится следующим образом. Формируется список «общих отношений» в группе, т. е. отношений, в которых участвует более половины объектов группы. Для каждого объекта подсчитывается доли общих отношений, в которых участвует данный объект. Если a — максимальная доля из всех имеющихся, то объекты, доля которых составляет менее 80% от a удаляются из группы и процесс повторяется снова до тех пор, пока оставшаяся группа не будет состоять из очень похожих друг на друга объектов. Информация о группировках включается в сеть, описывающую сцену, для дальнейшего использования на третьей и четвертой ступенях. Не может существовать универсальная процедура группирования, так как потребности различных программ, вызывающих процедуры группирования, могут быть различными. Например, группой можно считать тела, составляющие при подходящем совмещении некоторую заранее заданную фигуру.

Процедура обобщения по примерам работает следующим образом. Пусть системе предъявлена конструкция из трех прямоугольных брусков, соединенных в виде буквы П (назовем эту конструкцию аркой). В качестве первой гипотезы о понятии «арка» система выбирает описание сцены, полученное на предыдущих этапах. Однако по одному примеру трудно судить, какие из соотношений, присутствующих в описании, обязательны для арки, а какие факультативны. Для этого система использует новые примеры. Пусть вторая предъявленная сцена отличается от первой тем, что верхний брусок снят и положен перед двумя вертикальными. Такой пример маркирован учителем как отрицательный. Система сравнивает описание второго примера с описанием первого и находит. что единственным их различием является удаление из сети указателя ПОДДЕРЖИВАЕТСЯ. Отсюда делается вывод, что указатель имеет статус обязательного. Это выражается тем, что в сети, соответствующей текущему описанию (гипотезе), он заменяется на указатель ДОЛЖЕН ПОДДЕРЖИВАТЬСЯ. Программы нахождения аналогий и различий в сетях описаний играют на этом этапе важнейшую роль. Если описание негативного примера имеет несколько отличий от описания текущей гипотезы, то эти программы находят различие, фиксируемое на более высоком уровне сети, и объявляют его причиной расхождения, давая тем самым одному из отношений в текущей гипотезе статус обязательного. Результат сравнения сетей запоминается, поскольку может возникнуть необходимость возврата для пересмотра гипотезы и причине расхождений (из-за ее несовместимости с последующими примерами). Если текущий пример был позитивным, то различие в сетях обобщается с тем, чтобы обобщенное описание покрывало и старую гипотезу, и новый пример. Так, если в первом примере верхний брусок (горизонтальную часть) заменить треугольной призмой, то эта конструкция также будет представлять собой арку. Обучающаяся программа выводит из этого, что форма горизонтальной перекладины несущественна, т. е. свойство ПРИЗМА будет исключено из гипотезы.

Из сказанного ясно, что качество обучающих примеров и порядок их подачи имеют большое значение. Наиболее информативными оказываются примеры, близкие к типичным для данного понятия, но не являющиеся таковыми.

Пример системы Уинстона показывает, что вместе с общелогическими правилами необходимо использовать специфические для данной предметной области знания. Осознание этого факта привело к появлению систем, в которых знания отделены от управляющей программы (хотя могут воздействовать на нее). Управляющая программа играет роль интерпретатора знаний и может бытьболее или менее универсальной для разных предметных областей.

Примером обучающейся системы, основанной на использовании эксплицитно представленной базы знаний, является программа AM (Автоматический Математик) [Lenat, 1976, 1984]. AM является системой формирования теорий прямом математическом смысле слова (предметной областью является именно математика). База знаний, содержащая вначале только пару десятков элементарных математических понятий, в процессе работы системы обогащается новыми, имея возможность выводить также «теоремы», связывающие эти понятия. AM не имеет подсистемы дедуктивного вывода и не может оперировать понятием доказательства, поэтому ее теоремы носят эмпирический характер.

Каждое понятие В AM представляется фреймом с набором характеристик. Каждая характеристика соответствует какому-либо вопросу, который может быть задан относительно этого понятия, например: NAME («как называется понятие?»), DEFINITION («как -определяется понятие?»), INTEREST («чем оно интересно?»), GENERALIZATION («частным случаем каких понятий является данное?»), CONJECTURES («какие выводы можно сделать?») и т. д. Всего в AM фиксировано 25 таких характеристик. Не все характеристики могут быть заполнены, и задача системы состоит не только в формировании новых понятий, но и в уточнении уже сформированных.

Кроме фреймов, система располагает примерно 100 эвристиками, имеющими вид продукций <ЕСЛИ (условие), ТО (действие)>, в которых часть ЕСЛИ представляет собой некоторый набор предикатов, а часть ТО — набор акций, каждая из которых -может быть либо предложением рассматривать новую задачу, либо инструкцией, каким образом нужно определять новое понятие или заполнить аспекты уже сформированного. Пример такой продукции:

ЕСЛИ текущей задачей является <исследовать примеры понятия *F*>

И понятие *F* есть оператор с областью определения *A* и изменения *B*

И известно более чем 100 примеров из *F*

. . . .

И хотя бы один пример $b = f(a)$ является особым в *B* (например, является экстремальным в *B*),

ТО для каждого такого *b* сформировать следующее понятие:

NAME: <прообраз *b* по *F*>

DEFINITION: $\lambda aF(a)$ есть *b*

GENERALIZATIONS: *A*

. . . .

INTEREST: любой вывод с использованием *F* или F^{-1}

И причиной для формирования этого понятия считать: «Стоит исследовать те $a \in A$, которые имеют необычные *F*-значения»

И добавить к списку задач задачу: заполнить аспект CONJECTIL RES.

Продукции определяют поведение системы на нижнем уровне. Вопросы глобального управления, в том числе выбор продукций из множества применимых, выбор задач, на решение которых следует ориентироваться в данный момент, решают независимые процедуры, имеющие более общий характер, прототипом которых являются структуры [Ленат, 1975].

Особенностью АМ является автономность: система не нуждается в информации извне, поскольку сама порождает материал для своего исследования. К недостаткам следует отнести ее неспособность менять свои эвристики (обучаться новым продукциям).

Необходимость в эффективных обучающихся процедурах ощущается практически в любой области искусственного интеллекта. Диапазон реализованных систем чрезвычайно широк: от процедур пополнения базы знаний в универсальном решателе задач [Minton, 1985] до прогнозирования результатов на лошадиных бегах [Salzberg, 1985]. Много внимания уделяется обучению ври решении трудных комбинаторных проблем, в частности в шахматном програмировании [Science, 1986].

Синтез программ по примерам

Большинство работ в этой области посвящено синтезу программ в ограниченном (но функционально полном) подмножестве языка Лисп. Примитивами Лиспа, используемыми для этих целей, являются операции разбиении спмсков car, cdr и их суперпозиции, называемые базовыми функциями; операция сцепления списков cons; предикат atom; атомы Т, nil, а также условные н функциональные вызовы. В общем случае синтезируемые программы имеют следующий вид (для двухместной синтезируемой функции):

$$\begin{aligned}(Fzx) = (\text{cond }&((p_1x)\ (f_1xz))\\ &\ \ \ldots\ldots\\ &((p_kx)\ (f_kxz))\\ &(\text{T }(Hx(F(bx)\ (Gxz))))),\end{aligned} \qquad (6)$$

где H и G -— программы такого же типа, что и F (в частных случаях H или G могут опускаться); b — некоторая базовая функция; f_i—либо константа nil, либо комбинация car, cdr и cons, либо вызов.

Основной прием состоит в том, что для каждой пары вход-выходных списков (x, y) единым способом выписывается бесцикловая программа (называемая траекторией или полутраекторией вычисления y по x). Затем траектории, полученные по разным (x, y), сравниваются на предмет выявления какой-либо закономерности в их структуре, позволяющей сформулировать условные и функциональные вызовы и записать общую для всех (x, y) программу. Таким образом, при известных траекториях вычисления задача сводится к восстановлению циклов в программе.

Возможность получения траекторий в Лиспе основана на следующих ограничениях: ни один атом во входном S-выражении x не встречается дважды; все атомы выходного S-выражения y, за исключением nil, встречаются в x; единственной операцией, применяемой в процессе конструирования y по x, является присоединение некоторого атома или подсписка x к имеющемуся и данный момент выходному S-выражению.

При этих условиях траектория может быть однозначно вычислена функцией ST:

$$\boldsymbol{ST}(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \boldsymbol{p}(\boldsymbol{x}), \text{если } \boldsymbol{y} = \boldsymbol{p}(\boldsymbol{x}), \text{где } \boldsymbol{p} - \text{базовая функция} \\ \textbf{nil}, \text{если } \boldsymbol{y} = \textbf{nil} \\ \textbf{cons}\ \boldsymbol{ST}(\boldsymbol{x}, \textbf{car}(\boldsymbol{y}))\boldsymbol{ST}(\boldsymbol{x}, \textbf{cdr}(\boldsymbol{y}) \text{в противном случае.}) \end{cases}$$

Например, при x=(AB) и y=(BA) траекторией будет

$ST(x, y)$= (cons (cadr (cadr x)) (cons(car x)nil))

(согласно принятому в Лиспе способу написания (cadr x) обозначает базовую функцию (car(cdr x))).

Одна из первых методологий индуктивного синтеза программ изложена в [Summers, 1977]. В основе ее лежат не эвристики, а алгоритмы со строго характеризуемой областью применимости.

Ясно что, если по последовательности примеров (x_1, y_1), ..., (x_n, y_n) можно построить базовые функции b_i, такие, что предикаты (ATOM b_i x_j) ложны для $i \neq j$ и истинны для $i = j$, то программа

$$\begin{aligned}F(x)\ (\text{cond }&((\text{ATOM}(b_ix))\ (f_1x))\\ &\ \ldots\ldots\\ &((\text{ATOM}(b_nx))\ (f_nx)),\end{aligned}$$

где f_ix есть ST(x_iy_i), будет удовлетворять всем (x_iy_i).

Функции b_i выполняют роль указателей различий, вычленяя из входного списка ту часть, которая является атомом в случае x_i и не является атомом для остальных x. Например, для списков $x_1=(A)$ и $x_2=(A\ B)$ в качестве b_1 можно брать функцию (cdr x). Нахождение b_i в общем случае не представляет труда, однако для дальнейшего очень важно, чтобы они вычислялись каким-либо общим способом. В [Summers, 1977] приведен такой алгоритм вычисления b_i и доказывается, что, если, начиная с некоторою $i = k$ функции b_i и функции f_i могут быть выражены рекуррентными соотношениями рекуррентными соотношениями $b_{k+n}(x) = b_n(b(x))$ и $f_{k+n} = \mathrm{C}(f_n(b(x)), x)$, то функция F представляется в виде программы:

$F(x)$ = (cond ((ATOM(b_1x)(f_1x))
((ATOM(b_2x)(f_2x))
. . . .
((ATOM(b_kx)(f_kx))
(T ($C(f(bx)\ x)$)).

Эта теорема и ее обобщение устанавливают связь между рекуррентными соотношениями (повторяющимися шаблонами в траекториях) и рекурсивными схемами программы. Задача синтеза программ сводится к нахождению регулярных сегментов в траекториях примеров. Для этого можно применять, в частности, алгоритм унификации [Kodratoff, 1979]. Выбор унифицируемых траекторий в данном случае осуществляется с помощью эвристик.

В [Summers, 1977] предлагается и другой способ. В функциях f_i выбирается какой-либо константный список. Он заменяется на новую переменную z, начальное значение которой полагается равным замещенному списку. Затем ищется рекуррентное соотношение. Например, траекториями, полученными по примерам nil $\rightarrow$ nil, (A) $\rightarrow$ (A), (AB) $\rightarrow$ (BA) и (ABC) $\rightarrow$ (CBA), будут f_1 = nil, f_2 = (cons(car x) nil), f_3 = (cons (cadr x) (cons (car x)nil)), f_4 = (cons(caddr x) (cons (cadr x) (cons(car x) nil))). После введения новой переменной вместо nil траектории f_3 и f_4 будут иметь вид

(f_3xz) = (cons(cadr x) (cons(car x)z)),

(f_4xz) = (cons(caddr x) (cons(cadr x) (cons(car x)z))).

Рекуррентным соотношением между f_3 и f_4 будет

(f_4xz) = (f_3 (cdr x)) (cons (car x) z).

Было доказано, что если найденное соотношение сохраняется и для следующих вход-выходных пар, то можно синтезировать корректную программу (без функции H):

(Px) = (Fx nil),

(Fxz) = (cond ((ATOM x) z),

(T (F cdr x)(cons(car(x)z)))).

Изложенный метод получил развитие в последующих работах. В [Biermann, 1978] синтезируются программы, представляющие собой набор функций F_i вида

(F_ix)==(cond((p_1x) (f_1x))
((p_2x) (f_2x))
. . . .
(T ($f_{k+1}x$))),

где каждая f_j есть одна из функций nil, x, (F_j car x)), (F_j(cdr x)), (cons(F_mx) (F_nx)), P_j имеет вид (ATOM (b_jx)) и для всех j $b_{j+i} = b_jw$, где w — базовая функция, не равная тождественной. Кроме того, ни одна из функций $F_ш$ не должна вызываться более чем одной функцией f_j. Такого рода программы называются р е г у л я р н ы м и (поскольку могут быть представлены конечными автоматами).

Предложенная в [Kodratoff, 1979] техника синтеза является усилением метода Саммерса. В данном случае регулярности ищутся не между траекториями, вычисляющими y_i и y_{i+k}, а между y_i и частью y_{i+k}. 3

Среди списков y_{i+k} ищется такой, который можно разбить на подсписки $y^1{}_{i+k}$, $y^2{}_{i+k}$ и $y^3{}_{i+k}$ такие, что $y^2{}_{i+k}$ связано некоторым рекуррентным соотношением с y_i. После этого ставятся две новые задачи синтеза: по паре (x_i, $y^1{}_{i+k}$) и по паре (x_i, $y^3{}_{i+k}$). Показано, что если H есть программа, синтезированная по первой подзадаче, а G — по второй, то результирующая программа будет иметь вид (6) с одним дополнительным оператором (Px) = (Fxz_0), где z_0 — выбираемая в процессе работы алгоритма константа (результатом работы синтезированной программы считается P). Например, по парам x_1==(ABC), y_1 = = ($ABCCBA$) и x_2 = ($ABCB$), y_2 = ($ABCDDCBA$) можно построить разбиение y_2 на части y_2^1=(*A*), y_2^2= (*BCDDCB*), y_2^3=(*A*), причем траектория вычисления y_2^2 совпадает с траекторией y_1 при подстановке cdr *x*. Поскольку y_2^1 и y_2^3 выражаются через *x* как (car *x*), то результирующая программа будет иметь вид

(*Fxz*) = (cond ((ATOM *x*) *z*),

(*T* (*Hx*(*F* cdr *x*)(*Gxz*))))),

(*Hxz*) = (cons (car *x*) *z*),

(*Gxz*) = (*Hxz*),

(*Px*) = (*Fx* nil).

В [Jounaud et al., 1977, 1979] также применяется техника разбиения на три части, однако авторы используют несколько иной базис примитивов Лиспа: lcar(ABC) = (A), lrac(ABC) = (C), cdr*(ABC) = (BC),* rdc*(ABC) = (AB),* а также их композиции. По данному примеру $x \to y$ алгоритм разбивает *y* на три части: *u* append *(py c sy),* где *c* есть наибольший подсписок *y,* совпадающий с каким-либо подсписком *x*. Поскольку вычисление *c* по *x* посредством базовых функций несложно, то задача сводится к синтезу функций риз. Алгоритм находит наиболее короткие подвыражения x_1 и x_2 списка *x*, такие, что в x_1 и x_2 содержатся только те атомы, которые содержатся в *py* и *sy* соответственно. Применяя рекурсивно этот же алгоритм к примерам $x_2 \to py$ и $x_2 \to sy$ можно найти траекторию вычисления *y* по *x*, которая именно в силу своего построения обладает регулярной структурой и легко преобразовывается в программу. Особенностью метода является способность синтезировать программы по очень малому числу примеров.

Из приведенных примеров видно, что методы синтеза программ мало связаны со спецификой Лиспа. Они используют просто рекурсивную структуру программы. Такая постановка задачи синтеза содержится в [Бардзинь, 1982], где синтаксис и семантика программы не фиксируются, за исключением того, что языкк программирования содержит итеративные DO-операторы вида [I=α TO β <тело цикла>]. Задача состоит в восстановлении программ по протоколам (траекториям) их работы для частных примеров, т. е. в восстановлении DO-операторов по их разверткам. Формулируется понятие регулярности в текстах примеров и вводятся правила манипулирования этим понятием. При очень слабых ограничениях доказывается, что при большом числе примеров будет в конце концов синтезирована корректная программа (в дальнейших работах была доказана также возможность синтеза WHILE-циклов). Примечательно, что часто можно восстановление программы сопровождать доказательством ее корректности. В этом смысле можно сказать, что здесь речь идет не только об индуктивном синтезе программ, но и о восстановлении индуктивных рассуждений вообще. Эта взаимосвязанность используется также в [Shapiro, 1981], где алгоритм индуктивного синтеза программ является составной частью интерактивной системы отладки Пролог-программ.

Большинство разработок по синтезу программ отличает высокая сложность вычислений. Одним из способов сокращения перебора является использование знаний. В [Biermann et al., 1978] знания, записанные в продукционной базе знаний, представляют собой некоторые схемы программ, которые после конкретизации с помощью вход-выходной информации приводят к спецификации искомой программы. Окончательная формулировка программы понимается как определение значений переменных в этой спецификации, которое также производится с помощью базы знаний. В [Jantke, 1986; Lange, 1986] в качестве знаний используются спецификации класса программ, в котором ищется синтезируемая программа. Спецификации задаются посредством сигнатуры аксиом в исчислении первого порядка, определяющих отношения между основными примитивами в конструкции программ данного класса.

Более подробно с методами синтеза Лисп-программ можно ознакомиться в [Smith, 1982].

Системы индуктивного синтеза программ являются пока экспериментальными. Рассмотрим одно исключение из этого правила [Nix, 1983]. Представим ситуацию, когда непрограммирующий пользователь хочет изменить формат записи списка телефонных номеров таким образом, чтобы вместо записей

А. А.	Иванов— 11—22—33 (сл.)		Иванов А.	(сл.): 112233
Б. Б.	Петров— 12—34—56 (д.)	иметь	Петров Б.	(д.): 123456
Б. Б.	Петров — 23—45—67(сл.)		Петров Б.	(сл.): 234567

Если формат записей в первом случае изобразить условно шаблоном «X1.2.X3—X4—X5—X6_(X7).» (например, для первой записи X1 = А, X2 = А, X3 = Иванов и т. д.), то шаблоном преобразованных данных будет «X3_X1._(X7) : X4X5X6.». Задача системы по заданным примерам синтезировать функцию, осуществляющую преобразование

X1.X2.X3_-_X4-X5-X6_(X7) $\to$ X3_X1. _(X7): X4X5X6

Эта задача решается посредством последовательного синтеза исходного и результирующего шаблонов. Теоретическое исследование возможностей синтеза шаблонов по позитивным примерам выражений из языка, описываемого этим шаблоном, было предпринято в [Angluin, 1980 b]. Под шаблоном понимается любая конечная последовательность константных символов и переменных. Язык $L(p)$, определяемый шаблоном p — это множество слов, получаемых подстановкой произвольных цепочек константных символов вместо переменных в p. Каждая строка из приведенной выше «телефонной книжки» дает пример слова из языка, определяемого своим шаблоном.

Было показано, что существует универсальный алгоритм, синтезирующий в пределе любой язык $L(p)$, причем по каждому конечному подмножеству $L \subseteq L(p)$ в качестве промежуточной гипотезы синтезируется шаблон являющийся наилучшим по критерию предпочтения «p' лучше p'' для L, если из $L \subseteq L(p')$ и $L \subseteq L(p'')$ следует $L(p') \subseteq L(p'')$. Кроме того, p_L является наименее общим шаблоном для L (в смысле подстановки шаблонов вместо переменных), К сожалению, задача определения p_L по меньшей мере NP-сложна. В то же время Д. Англюин приводит примеры различных типов шаблонов, для которых эта задача полиномиальна. Для шаблонов, рассматриваемых в [Nix, 1983] задача нахождения p_L является NP-полной. Более того, если искомой программой является преобразование $p \to p'$ (p и p'—шаблоны), то нахождение даже при известном p также является NP-полной задачей. Тем не менее, удачный подбор эвристик позволил автору реализовать систему синтеза, как правило, справляющуюся с заданием за приемлемое время и с использованием малого числа примеров.

Отметим еще одну работу по синтезу шаблонов [Shinohara, 1982, 1986]. Если вводимая в машину текстовая информация подчиняется некоторому шаблону, система, контролирующая ввод, может через некоторое число примеров обучиться этому шаблону и известить оператора о том, что он может вводить только переменную информацию. Например, если вводимой информацией является библиография статей, задаваемых шаблоном.

«Автор: X1 Название: X2 Журнал: X3 Год (том): X4(X5) Номер X6, то система может автоматически набирать константные части шаблона, остиавляя пользователю места для заполнения X1-X6. Используемый ’ алгоритм имеет полиномиальную сложность и рассчитан на синтез так называемых регулируемых шаблонов, т. е. шаблонов, в которых каждая из перечисленных частей встречается только один раз (языки, определяемые такими шаблонами, являются регулярными).

В заключение главы отметим, что основными направлениями исследований в области обучения в настоящее время являются: выбор языков для формулирования эмпирических и теоретических утверждений; нахождение правил рационального индуктивного вывода для связи эмпирических утверждений; нахождение методов обоснования и оценки интересности множества выведенных теоретических утверждений.

СПИСОК ЛИТЕРАТУРЫ

Адельсон-Вельский и др., 1978	Адельсон-Вельский Г.М.Программирование игр. - М.: Наука, 1978. - 255 с.
Адельсон-Вельский и др., 1983	Адельсон-Вельский Г.М., Арлазаров В.Л., Битман А.Р., Донской М.В. Машина играет в шахматы. - М.: Наука, 1983. - 207 с.
Айзерман и др., 1970	Айзерман М.А., Браверман Э.М., Розоноер Л.И. Метод потенциальных функций в теории обучения машин. - М.: Наука, 1970. - 240 с.
Бардзинь и др., 1972	Бардзинь Я.М., Фрейвалд Р.В. 0 прогнозировании общерекурсивных функций // ДАН СССР. 1972. - Т.206, N 3. - С. 521-524.
Бардзинь и др., 1974	Бардзинь Я.М., Кинвер В.Б., Подниекс К.М. Об ускорении синтеза и прогнозирования функций // Ученые записки Латв.ГУ. - Рига, 1974. - Т.210.

- С. 117-128.

Бардзинь, 1982 — Бардзинь Я.М. Некоторые правила индуктивного вывода и их применение // Семиотика и информатика. - 1982. - Вып. 19. - С.59-89.

Вапник и др., 1974 — Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. - М.: Наука, 1974. -.415 с.

Гаазе-Рапопорт 1987 — Гаазе-Рапопорт М.Г., Поспелов Д.А. От амебы до робота: модели поведения. - М.: Наука, 1987. - 285 с. 285

Гаек и др., 1984 — Гаек П., Гавранек Т. Автоматическое образование гипотез: Пер. с англ. - М.: Наука, 1984. - 277 с.

Гладун, 1970 — Гладун В.П. Формирование понятий путем обучения растущих сетей // Кибернетика.- 1970.- N 2. - С.

Гэри и др., 1982 — Гери М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: Пер. с англ. - М.: Мир, 1982. - 416 с.

Дружинин, 1973 — Дружинин Н.К. Логика оценки статистических гипотез. - М.: Статистика, 1973. - 212 с.

Журавлев, 1978 — Журавлев Ю.И. Об алгебраическом подходе к решению задач распознавания и классификации // Проблемы кибернетики. - М.: Наука, 1978. - Вып. 33. - С. 5-68.

Кендел и др., 1960 — Кендел М., Юл Дж. Э. Теория статистики.: Пер. с англ. - М.: Госстандарт. - 1960. - 401 с.

Кинбер, 1977 — Кинбер Е.В. Об ускорении предельной идентификации путем изменения порядка вопросов // Elekt.Information sverarb.Kibern. (EIK). -1977. V.17. - P. 345-349.

Ленат, 1975 — Ленат Д. БИИНГИ: знание как система взаимодействующих экспертов // Тр. 1Y междунар. конф, по искусственному интеллекту. - Тбилиси, 1975. - Т. 2. - С. 63-96.

Погосян, 1977 — Погосян Э.М. К теории автоматического синтеза понятий // Семиотика и информатика. - 1977. - Вып. 8. - С. 125-152.

Погосян, 1983 — Погосян Э.М. Адаптация комбинаторных алгоритмов. - Ереван: Изд-во АН АрмССР, 1983. - 288 с.

Подниекс, 1974 — Подниекс К.М. Сравнение различных типов предельного синтеза и прогнозирования функций // Учен, записки Латв. ГУ. - Рига, 1974. - Т. 210. - С. 68-81.

Подниекс, 1975 — Подниекс К.С,. Сравнение различных типов предельного синтеза и прогнозирования. / Учен, записки Латв.ГУ. - Рига. 1975. - Т. 233. - С. 35-44.

Попов Э.В. Общение с ЭВМ на естественном языке. - М.: Наука, 1982. - 360 с.

Поспелов Д., 1981 — Поспелов Д.А. Логико-лингвистические модели в системах управления. М.: Энергоиздат, 1981.- 231 с.

Поспелов Д., 1986 — Поспелов Д.А. Ситуационное управление: теория и практика. - М.: Наука. 1986. - 284 с.

Трантенброт и др. 1970 — Трахтенброт Б.А., Вардзинь Я.М. Конечные автоматы (поведение и синтез)- М.: Наука,1970.- 400 с.

Уинстон, 1978 — Уинстон П.Г. Построение структурных описаний по примерам // Психология машинного зрения: Пер. с англ. - М.: Мир, 1978. - С. 185-248.

Финн и др., 1981 — Финн В.К., Забежайло М.И. Логические проблемы автоматического формирования гипотез // Тез. IX Всесоюз. симп. по кибернетике. - М.: ВИНИТИ, 1981. - Т.2. - С. 54-60.

фрейвалд, 1974 — фрейвалд Р.В. Предельно вычислимые функции и функционалы //Учен. зап. Латв. ГУ. - Рига, 1974. - Т.210. - С.6-19.

Фрейвалд, 1975 Фрейвалд Р.В. 0 сложности и оптимальности предельных вычислений. Учен. зап. Латв. ГУ. - Рига, 1975. - Т. 233. - С. 155-173.

Фрейвалд и др., 1975 Фрейвалд Р.В., Вардзинь Я.М. Соотношения между прогнозируемостью и предельной оинтезируемоотью //Учен. зап. Латв. ГУ. - Рига, 1975. - Т. 233. - С. 26-34.

Фукунага, 1979 фукунага К. Введение в статистическую теорию распознавания. - М.: Наука, 1979. - 368.

Хант и др., 1970 Хант Э., Марин Дж., Стоун Ф. Моделирование процесса формирования понятий на вычислительной машине: Пер. с англ. - М.: Мир, 1970. - 301 с.

Цыпкин, 1968 Цыпкин Я.З. Адаптация и обучение в автоматических системах. - М.: Наука. 1968. - 399 с.

Цыпкин, 1970 Цыпкин Я.З. Основы теории обучающихся систем. - М.: Наука, 1970. - 252 с.

Angluin, 1980 a Angluin D. Inductive inference of formal languages from positive data // Information and Control.-1980. -N.45.-1980.-P.117-135.

Angluin, 1980 b Angluin D. Finding patterns common to a set o£ strings // J. Comput. System Science. -1980. -N.21. -P.46-62.

Angluin, 1982 Angluin D. Inference of reversible languages // J. ACM. -1982. -N.29. -P. 741-765.

Angluin et al., 1983 Angluin D., Smith C.H. Inductive inference: theory and methods // Comp. Surveys. -1983. -Vol.15, N.3. -P.237-269.

Biermann et al., 1972 Biermann A.W., Feldman J.A. On the synthesis of finite-state machines from samples of their' behavior// IEEE Trans.-1972. V.-SMC-21.-N. -P. 592-597.

Biermann, 1978 Biermann A.W. The inference of regular LISP Programs from Examples // IEEE Trane. -1978.- SMC-8. -P. 585-600.

Blum et al., 1975 Blum I>., Blum M. Toward a mathematical theory of inductive inference // Information and Control. -1975. -N.28. -P.125-155.

Brayer et al., 1977 Brayer J.M., Fu K.S.A Note on the K-tail method of free grammar inference // IEEE Tranв. -1977. -SMC -7. - P. 293-300.

Buchanan et al., 1978 Buchanan G.B., Feigenbaum E.A. DENDRAL and • META-DENDRAL, their applications* dimensions // Artif. Intell. - 1978. -N.11.-P.5-24.

Case et al., 1983 Case J., Smith C. Comparison of identifications criteria for machine inductive inference // Theor. Comput. Scien.-1983.-N.25.-P.193-220.

Cook et al., 1976 Cook C.M., Rosenfeld A., Aronson A.R. Grammatical inference by hill-climbing // Infonnation Science. -1976. -N.10.-P.59-80.

Daley, et al., 1977 Daley R. On the inference of optimal descriptions // Theor. Comput. Soien.-1977.-N.4.-P. 301-319.

Daley et al., 1983 Daley R., Smith. C. On the complexity of inductive inference (Tech. Rep. N.83-4).- Dept. Coniput. Soien., University of Pittsburgh, 1983.

Dietterich, 1979 Dietterioh T.G. The methodology of knowledge layers for inducing descriptions of sequentially orderd events. Master's thesis. - Urbana: Univ, of Illinois, 1979.

Dietterich, 1981 Dietterioh T.G., Michalski R.S. Inductive learning of structural descriptions: evaluation criteria and comparative review of selective methods // Artif. Intel. - 1981.- 16 P.257-294.

Gold, 1967 Gold E.M. Language identification in the limit// Information and Control.-1967.-N.10.-P.447-474.

Hayes-Roth et al., 1978 Hayes-Roth F., McDermot J. An interference matching technique for inducing abstractions // Communications of the ACM.-1978.-N.21.-401-411.

Horning, 1969	Horting J.J. A study of agrammatical inference: Ph. D. Dissertation, Computer Science Dept., Stanford Univ., Stanford, Calif.,1969.
Jantke, 1986	Jantke K.P. An algebraic framework for inductive program synthesis // Lecture Notes in Computer Science.-1986.-Vol.215.P.101-122.
Jantke at al., 1981	Jantke K.P., Beiok H.R. Combining postulates of naturalness in inductive inference // Electron. Informations-Verarb. Kybern..-1981.-N.17.-P. 465-484.
Jounaud at al., 1977	Jounaud J.R., Guiho G. Program synthesis from examples for a simple class of non-loop functions.//Technical rep.. Lab. de Resherohe en Infoimatique , Univ. Paris-Sud, Paris, 1977.
Klette et al., 1980	Klette R., Wiehagen R. Research in the theory of inductive -inference by GDR mathematicians - a survey. // Information Soienpe -1980.-N.22. -P.149-169.
Kodratoff, 1979	Kodratoff Y. A class of functioned synthesized from a finite umber of examples and a LISP pregram scheme // Int.J Comp, and Inf. Science. -1979. -N.8. -P.489-521.
Kugel, 1977	Kugel P. Induction, pure and simple // Information and Control. -1977.-N.35.-P.276-337.
Lange, 1986	Lange S. A program syntheses algorithm exemplified // Lecture Notes in Computer Science.-1986.-P.185-193.
Lenat, 1976	Lenat D. AM: an artificial intelligence approach to discovery in mathematics as heuristic search // Comp. Science Dept. Rep. STAN-CS-76-570. Stanford Univ. -Stanford, Calif.,1976.
Lenat et al., 1978	Lenat D..Harris G. Designing a rule system that searches for scientific discovery // Pattern Directed Inference Systems.-N.Y.: Academic Press,1978.-P.25-51.
Lenat, 1984	Lenat D. Automated theorem formation in mathematics // Contemporary math.-1984. - N.29.-P.287-314.
Levine, 1982	Levine B. The use of free derivatives and a sample support parameter for inferring tree systems // IEEE Trans. Pattern Anal. Maoh. Intel.-1982.-P.25-34.
Michalsky, 1983	Michalski R.A. A theory and methodology of inductive learning // Artif. Intell. -1983. -Vol.20., N-2.-P.11-161.
Minton, 1985	Minton S. Selectively generalizing plans for problem-solving // Proo. 9th IJCAY.-1985. Vol.1. -P.596-599.
Nix, 1983	Nix R. Editing by Examples.Ph.D.dissertation, Computer Science Dep.,Yale University.-New Haven, Connecticute, 1983.
Osherson et al., 1982	Osherson D., Wienstein S. Criteria of language learning // Information and Control.-1982.-N.52. -P.123-138.
Plotkin, 1970	Plotkin G.D. A note on inductive generalization. // Machine Intell.- 1970.-N.5.-P.153-163.
Plotkin, 1971	Plotkin G.D. A further note on inductive generalization // Machine Intell.- 1971.-Vol. 5.-P.101-124.
Pocorny, 1980	Pocorny D. Knowledge acquisition by the GUHA method // Int. J. Policy Anal.Infonn. Syst.-1980.-N.4. -P.379-399.
Reynolds, 1970	Reynolds J.C. Transformational systems and the algebraic structure of atomic formulas // Machine Intelligenoe-1970.-N.5.-P.135-151.
Salzberg, 1985	Salzberg S. Heuristics for inductive learning // Proo. 9th IJCAI.-1985.-Vol.1.-P.603-609.
Shapiro, 1981	Shapiro E. Inductive inference of theories from facts // Tech. Rep. 192 Dept. Comput. Soien., Yale Univ. -New Haven, Conn. -1981.
Shinohara, 1982	Shinohara T. Polinomial time inference of extended regular pattern languages // Lecture Notes in Computer Science.-1982.-N.147.-P.115-127.
Shinohara, 1986	Shinohara T. Some' problems on inductive inference from positive data // Lecture Notes in Computer Soi. -1986.-N.215.-P.41-58.

Science, 1986	Skience C. An overview of machine learning in computer chess // ICCA J.-1986.-N.1.-P.20-28.
Smith, 1982	Smith D.R. A survey of the synthesis of LISP programs from examples // Automatic Program Construction Techniques.N.Y.:MacMillan, 1982.
Solomonoff, 1964	Solomonoff R.J. A formal theory of inductive inference // Information and Control.-1964.-N.7. -P.1-22, 224-254.
Soloway et al., 1977	Soloway E.M., Riseman E.M.Levels of pattern descriptions in learning // Papers 5th IJCAI. -Cambridge,Mass.-1977.-P•801-811.
Summers, 1977	Summers P.D. A methodology for LISP program construction from examples //J.ACM.-1977. -N.24.-P.161-175.
Vere, 1980	Vere S.A. Multilevel counter-faotuals for generalizations of relational concepts and productions // Artif. Intell. 1980. N.14. -P. 139-164.
Weihagen, 1978	Weihagen R. Characterization problems in the theory of inductive inference // Lecture Notes in Comp.Sci.-1978.-N.62.-P.494-508.
Wilkins, 1982	Wilkins D. Using knowledge to control tree searching // Artif. Intell. 1982.-N.18. -P.1-15.

Appendix 9: On the Background of Tournment Enpower of Cognizers

Pogossian E. Systems of incomparable sets with the smallest number of subsets[1]

PhD, Computing Center of Academy of Sciences, Moscow, 1970. Math. Problems of Cybernetics and Comp. Science, Yerevan, 1986, v. XVI,pp 148-161.

Introduction

In this work a combinatorial apparatus for investigation of features of subsets of finite sets is described, with using of which a solution of an extreme combinatorial problem on constructing systems of incomparable sets with the smallest number of subsets.

The problem is stated as follows.

For a given finite set *M* and an integer set $l_1,\ldots, l_k, m_1\ldots,m_k,$ determine *m* subsets of the set *M* with the power l_j, $1 \leq j \leq k$, such that no sets will be pairwise contained in each other and the number of different subsets of power *i,* where *i* is an arbitrary natural number, obtained from the sets will be the smallest.

The systems of sets of this kind under condition $l_1 = \ldots = l_k = l$ were first built by Macaulay in 1927 [1], who also proved their extremality for $i = l-1$. A generalization of Macaulay theorem for a non-binary case and an arbitrary i was obtained by Clements and Lindstrôm in 1969.

In 1970 the author independently derived a complete solution of the mentioned problem [3].

In the work non-enumerative algorithms for the construction of the mentioned extreme system of subsets of the set *M,* called a compacted set here, computation of the number of subsets of some *compacted set* of given power, as well as proof of the fact that the system is the one being sought are expounded.

As it has been demonstrated in the work, the actual construction of the compacted set requires sequential counting of $m_1 + m_2 + \ldots + m_k$ lexicographically ordered subsets of the set *M.* In addition it reveals that the compacted set is built in the most saving mode, meaning that it uses only a minimal number of elements of the set *M*, from which the construction of such kind of systems is possible in general.

The combinatorial apparatus under consideration made basis for construction of systems of arbitrary sets with the smallest number of subsets [5]. The corresponding problem is formulated in the following way:

for the given finite set *M* of power *n* and a number $m < 2^n$ build a system from m subsets of *M,* such that the number of different subsets of power i^* , obtained from the elements of the system will be the smallest, where i^* is a minimal number i in the range 0 to *n* such that there exists at least one system of the above-mentioned type with the number of subsets less than the number of all subsets of *M* with the power *i.*

Based on solution of above-noted problems non-enumerative algorithms for construction of lower bounds of minimal test lengths of specific tables were suggested in [3, 4]. Estimates of such kind obtained earlier are asymptotically dependent on parameters of tables [5, 6].

The paper consists from four sections.

[1] PhD, Computing Center of Academy of Sciences, Moscow, 1970. Math. Problems of Cybernetics and Comp. Science, Yerevan, 1986, v. XVI,pp 148-161.

In section 1 there are given main definitions and the main result of the paper, namely the theorem on the minimal number of subsets, is formulated. The complete proof of the main theorem is set out in the subsequent sections 2-4.

In section 2 we give the algorithm for producing suites of sequences from segments of which the compacted sets will be arranged. In essence the method provides obtaining a lexicographically ordered sequence composed from all possible distinct subsets of the fixed set.

An analysis of a special class of graphs- Δ - graphs, is conducted. For a set of Δ - graphs a lemma on isomorphism is proved and a number of quantitative regularities reflecting the properties of vertex sets of Δ - graphs are rendered. It is proved that there exists an inductive method of producing of Δ graphs. This fact makes pivotal use in the proof of the main theorem.

In section 4 the concluding part of proof of main theorem is set out. It is conducted by induction on Δ -graphs and is significantly based on results obtained in 1-2.

1. Main definitions. Statement of the theorem.

In further exposition we will call any finite set of elements considered with some fixed relation of linear ordering a bundle. An empty bundle is an empty set by definition.

An arbitrary finite set composed from elements of the bundle A with assigned ordering induced by ordering of A we will call a sub-bundle of bundle A.

Any sub-bundle of bundle A distinct from A we will call a proper sub-bundle of A.
We will call the length of A the number of elements in the bundle A.

Let $I_{n,p}$ where n, p are arbitrary integer positive numbers or zeros be a bundle $\{p+1,\dots,p+n\}$ and $R_{n,p}$ be the set of all subsets of $I_{n,p}$. If $p=0$ then the index p will be omitted in corresponding places of denotations. If $n=0$ then $R_{o,p}=I_{o,p}$ is an empty bundle by definition.

The elements of the set $R_{n,p}$we will call $\Delta_{n.p}$ *bundles*, we refer as $\Delta^k_{n,p}$to $\Delta_{n.p}$ bundles of length k, $0\le k\le n$, (speaking of $\Delta^k_{n,p}$ sub-bundles and proper sub-bundles $\Delta^k_{n,p}$ respectively).

Denote the set of all $\Delta^k_{n,p}$ bundles through $R^k_{n,p}$.

It is obvious that $\left|R^k_{n,p}\right| = C^k_n$ with $0\le k\le n$, and

$$|R_{n,p}| = \left|\bigcup_{k=0}^{n} R^k_{n,p}\right| = \sum_{k=0}^{n} C^k_n = 2^n.$$

For $\Delta_{n,p}$ bundles we introduce operations of union $\cup$, intersection $\cap$, subtraction /, defined as follows: if A and B are $\Delta_{n,p}$bundles, P and Q are sets of their elements then $A\cup B$(respectively $A\cap B$, $A\setminus B$)we define as sets $P\cup Q$ ($P\cap Q$, $P\setminus Q$ respectively), treated simultaneously with relation of ordering induced in bundle $I_{n,p}$.

Define a relation of linear ordering $\prec$ in $R_{n,p}$.

For arbitrary distinct $\Delta_{n,p}$ bundles $B=\{b_1,\dots,b_{k_2}\}, A=\{a_1,\dots,a_{k_1}\}$ we assume:

1. If $k_1\ne k_2$, then: *a*) A is smaller than B ($A\prec B$), if $k_1<k_2$: *b*) B is smaller than A ($B\prec A$), if $k_2<k_1$.
2. Assume $k\ne k_2$. Then there exists a $0\le j\le k_1-1$ such that $a_i=b_i$for any $i\in I_j$ and $a_{j+1}\ne b_{j+1}$. We suppose : *a)* $A\prec B$ if $a_{j+1}<b_{j+1}$; *b*) $B\prec A$, if $b_{j+1}<a_{j+1}$.

We denote $\succ$ (greater) the relation inverse to $\prec$. If $\Delta_{n,p}$ bundle A is smaller or equal to $\Delta_{n,p}$ bundle of B (A is greater or equal to B accordingly) we will write $A \preccurlyeq B$ ($A \succcurlyeq B$ respectively).

It can be easily seen that a lexicographically ordered set $R_{n,p}$ is isomorphically embedded in the set of all lexicographically ordered words in an n-letter alphabet. Then the relation of ordering $\prec$ introduced in $R_{n,p}$ we will call lexicographical.

The sequence of $\Delta_{n,p}$- bundles $A_1, \dots, A_m$ we call $+$*compacted*, if

$\forall_j (j \in I_{m-1} \to A_j \prec A_{j+1} \& \rceil\exists A(A \in R_{n,p} \& A_j \prec A \prec A_{j+1}))$,

and $-$compacted, if

$$\forall_j (j \in I_{m-1} \to A_j \succ A_{j+1} \& \rceil\exists A(A \in R_{n,p} \& A_j \succ A \succ A_{j+1})).$$

$\Delta_{n,p}$-bundle $B = \{b_1, \dots, b_{k_2}\}$ we define as k_2-*extension* of $\Delta_{n,p}$-bundle $A = \{a_1, \dots, a^{k_1}\}$ if $a_i = b_i$ for any integer i from 1 to $k_1$and $k_2 \geq k_1$.

An arbitrary k-extension B of the $\Delta_{n,p}$ bundle A we will call the *extension* of A, while the relation «bundle B is k-extension of bundle A» we will call the *relation* of k-extension.

A lexicographically ordered sequence of $\Delta_{n,p}^{k_2}$-bundles, composed from all distinct k_2-extensions of all bundles of some sequence $\Delta_{n,p}^{k_1}$-bundles $A_1, \dots, A_m$ or any set $\Delta_{n,p}^{k_1}$-bundles S, where $0 \leq k_1 \leq k_2 \leq n$, we will call a k_2-*extension of the sequence* $A_1, \dots, A_m$ and k_2-*extension of the set* S, respectively.

It is clear that k_2-extension of a arbitrary set $\Delta_{n,p}^{k_1}$-bundles S coincides with the k_2-extension of an arbitrary sequence consisting from all bundles of set S, k_2-extension of an arbitrary sequence of $\Delta_{n,p}^{k_1}$-bundles coincides with the k_2-extension of the set of all bundles of the same sequence.

We shall note that some sets or sequences $\Delta_{n,p}^{k_1}$-bundles can have no k_2-extensions.

For an arbitrary $\Delta_{n,p}$-bundle A any bundle $B = A \cup C$, where C is an arbitrary $\Delta_{n,p}$-bundle, we will call a $\Delta_{n,p}$ - a subbundle of bundle A. $\Delta_{n,p}$-sub-bundle B of bundle A we call a proper $\Delta_{n,p}$-*sub-bundle* A if B is distinct from A.

In particular any k-extension of an arbitrary $\Delta_{n,p}$-bundle A is simultaneously a $\Delta_{n,p}$-subbundle A.

Let's denote$(A)_i$,$i \in \{1,2, \dots, |A|\}$, the i-th element of an arbitrary bundle A. Hence $A = \{(A)_1, \dots, (A)_{|A|}\}$.

We assume defined $(A)_o = (\emptyset)_k = 0$, where A is an arbitrary $\Delta_{n,p}$-bundle, $0 \leq k \leq n$.

It should be noted also that the number of k-extensions of an empty bundle is defined with parameter n. Therefore we adopt terms "empty $\Delta_{n,p}$-bundle" or $\Delta_{n,p}^0$-bundle".

Lemma 0 following beneath implies that all reasoning regarding lexicographical ordering and k-extension can be held for the set R_n, while derived results can be transferred to an arbitrary set $R_{n,p}$without any modification.

L e m m a 0. *An arbitrary lexicographically ordered set $R_{n,p}$ is isomorphic to lexicographically ordered set R_n with respect to order and k-extension relations for any k, $0 \leq k \leq n$. При этом:*

3. The bijective mapping θ of set $R_{n,p}$to set R_n in this isomorphism is defined uniquely;

2. $\forall BC (B \in R_{p,n} \& C \in R_n \to ((B = \emptyset \to \theta(B) = \emptyset) \&$
$\& (C = \emptyset \to \theta^{-1}(C) = \emptyset) \& (B \neq \emptyset \to \theta(B) = \{(B)_1 - p, \dots,$
$(B)_{|B|} - p\} \in R_{n,p}) \& (C \neq \emptyset \to \theta^{-1}(C) = \{(C)_1 + a, \dots, (C)_{|C|} + p\} \in R_n))).$

4. If $B_1, \dots, B_m$ and $C_1, \dots, C_r$ $-$ $+$-compacted sequences from$R_{n,p}$ и R_n respectively, then sequences $\theta(B_1), \dots, \theta(B_m)$ and $\theta^{-1}(C_1), \dots, \theta^{-1}(C_r)$ are –compacted sequences of bundles fromR_n and $R_{n,p}$

Hereinafter the set R_n is assumed lexicographically ordered.

We say that Δ_n-bundles A и B *are incomparable with respect to inclusion if they are not contained in each other in pairs.*

The set of bundles pairwise incomparable with respect to inclusion (s. p. i. b) we specify with the following system of numbers.

Let $S = \{S_1, \dots, S_m\}$ be an arbitrary s. p. i. b.

Let's divide the set S to non-empty subsets $P_1, \dots, P_k$, where $k \geq 1$, each P_t consists from bundles of some fixed length l_t and lengths of bundles included in distinct sets $P_1, \dots, P_k$ are different.

Without loss of generality we can take $l_1 > l_2 > \cdots > l_k$ *and denote numbers* $|P_1|, \dots, |P_k|$ as $m_1, \dots, m_k$.

The sequence of numbers $\chi = (n,\ l_1, \dots, l_k,\ m_1, \dots, m_k)$ obtained in the above indicated way we will call the characteristic sequence (c, s) of s. p. i. b. S.

An arbitrary integer sequence of type $\chi = (n,\ l_1, \dots, l_k,\ m_1, \dots, m_k)$ we will call the characteristic sequence (c, s) if it satisfies the following conditions:

$$1.\, n \geq l_1 > l_2 > \cdots > l_k \geq 0;$$
$$2.\, m_i \neq 0, \text{при } i = 1, \dots, k.$$

With an arbitrary k, $0 \leq k \leq n$, denote $\Gamma_n^k(1), \Gamma_n^k(2), \dots, \Gamma_n^k(C_n^k - 1), \Gamma_n^k(C_n^k)$ as a $+$-compacted sequence consisting from all Δ_n-bundles of length k, a через $L_n^{n-k}(j)$, with $j \in \{1, 2, \dots, C_n^k\}$, denote the difference $I_n \backslash \Gamma_n^k(j)$.

With any c.s. $\chi = (n,\ l_1, \dots, l_k,\ m_1, \dots, m_k)$ we will associate a set L_χ called a compacted set with c.s. χ defined as a set of Δ_n-bundles of the following type:

$$L_\chi = \{L_n^{l_1}(1),\ L_n^{l_1}(2),\ \dots,\ L_n^{l_1}(m_1),\ L_n^{l_2}(\tilde{m}_1 + 1),\ \dots,\ L_n^{l_2}(\tilde{m}_1 + m_2),$$
$$\dots,\ L_n^{l_k}(\tilde{m}_{k-1} + 1),\ \dots,\ L_n^{l_k}(\tilde{m}_{k-1} + m_k)\},$$

where $L_n^{l_j+1}(\widetilde{m}_j) = I_n \backslash \Gamma_n^{n-1_{j+1}}(\widetilde{m}_j)$ и $\Gamma_n^{n-1_{j+1}}(\widetilde{m}_j)$ is the greatest in lexicographical ordering of $(n - l_{j+1})$-extension of the last bundle of the sequence $\Gamma_n^{n-l_j}(1), \Gamma_n^{n-l_j}(2), \dots, \Gamma_n^{n-l_j}(\widetilde{m}_{j-} + m_j)$ having a non-empty $(n - l_{j+1})$-continuation, where $j = 1, 2, \dots, k-1$, $\widetilde{m}_0 = 0$.

From this definition it can be seen that for producing of a compacted set with given c.s. $\chi = (n,$ $l_1, \dots, l_k,\ m_1, \dots, m_k)$ it is necessary to define $\sum_{j=1}^{k} m_j$ of Δ_n-bundles.

We will state the main theorem.

T h e o r e m 1(on the smallest number of subsets). *For any* χ *let* K_x *be the class of all* s. p. i. b. *with c.s.* χ, L_x *– compacted set with c.s.* χ, *then:*

5. *Class* K_x *is non-empty if and only iff it is possible to produce a compacted set with c.s..* χ.
6. *If class* K_x *is non-empty then for arbitrary* i *и* S, *where* $0 \leq i \leq n$, $S \in K_x$, *the number of distinct subsets of the power* i, *obtainable from bundles of the set* L_x *is not greater than the number of distinct subsets of the power* i *obtained from bundles of the set* S.

C o r a l l a r y 1. *For any c.s.* χ *the number of different elements required for producing the compacted set with c.s.* χ *does not exceed the number of elements necessary for producing an arbitrary* s. p. i. b *with the same c.s.*

The corollary is immediately implied from theorem 1 if we observe that the number of elements used for producing s. p. i. b we are interested in coincides with the number of one-element subsets of the bundles.

The property of non-redundancy indicated in Corollary 1 or the property of efficiency of compacted sets with respect to the number of elements used in producing can serve as some enlightenment to assertion of Theorem 2.

For an arbitrary set of bundles S and natural number i we identify the number of all subbundles of length i from bundles of S and denote it $N_i\ (S)$.

Corallary 2. *Let χ be an arbitrary c.s. $\chi = (n,\ l_1, \dots, l_k,\ m_1, \dots, m_k)$, if K_χ is a non-empty class s. p. i. b. with c.s. χ and L_χ is a compacted set with c.s. χ, then:*

$$N_i\,(L_x) = \min_{S \in K_x} \left(\sum_{t=1}^{m} (-1)^{t-1} \sum_{\{S_{p_1}, \dots, S_{p_t}\} \subseteq S} C^{i}_{|S_{p_1} \cap S_{p_2} \cap \dots \cap S_{p_t}|} \right),$$

where $0 \le i \le n,\ \ m = \sum_{j=1}^{k} m_j$. *?*

Theorem 1 and corollary 2 apart from independent interest in combinatorics can be useful in the theory of control systems, coding theory and economics. One of such application of theorem 1 in the theory of tests is indicated in [3, 4].

In further sections 2–4 the complete proof of theorem 1 and corollary 2 will be given. Furthermore in sections 2 and 3 we have identified the connected parts which, from our point of view, present more general interest than the referred proof.

2. THE ALGORITHM FOR COMPACTED SEQUENCES PRODUCING

The algorithm Γ_n^k for sequential generation of a + -compacted sequence of Δ_n^k-bundles with $0 \le k \le n$ is defined in the following way.

7. If $k = 0$, then $\Gamma_n^o(1)$ is an empty bundle by definition and the algorithm accomplishes the work.

8. Let $0 < k \le n$. Then:

1. Assign the value $\Gamma_n^k(1) = \mathrm{I}_k$.

2. Let $\Gamma_n^k(i)$ *be* Δ_n^k-bundle that has already been produced on the i step by the algorithm Γ_n^k, then $\Gamma_n^k(i+1)$ is produced in such a way that:

a) if $\Gamma_n^k(i) \prec \{1 + n - k, \dots, n\}$ then:

$$\exists q \forall xy\,(1 \leqslant q \leqslant k \,\&\, 1 \leqslant x \leqslant q \,\&\, q < y \leqslant k \rightarrow$$
$$(\Gamma_n^k(i))_x < x + n - k \,\&\, (\Gamma_n^k(i))_y = y + n - k).$$

The statement follows from lemma 2, corollary 2.2 and definition of Δ_n-bundles.

We identify $\Gamma_n^k(i+1) = \{(\Gamma_n^k(i))_{1_1}, \dots, (\Gamma_n^k(i))_{q-1}, (\Gamma_n^k(i))_q + 1,\ (\Gamma_n^k(i))_q + 2, \dots, (\Gamma_n^k(i))_q + k - q + 1\}$;

б) if $\Gamma_n^k(i) = \{1 + n - k, \dots, n\}$, the algorithm completes the work and assigns $i = C_n^k$.

Further in lemma 2 a necessary and sufficient condition for + compactness of a given sequence of Δ_n^k-bundles, which provides in essence the justification of the algorithm Γ_n^k.

Applying the algorithm Γ_n^k and lemma 4 following below we obtain the algorithm L_n^{n-k} of sequential producing of –compacted sequence of Δ_n^{n-k}-bundles, taking $L_n^{n-k}(i) = I_n \setminus \Gamma_n^k(i)$, where $1 \le i \le C_n^k$.

Assuming some $k,\ 0 \le k \le n$, we will identify the sequences $\Gamma_n^k(1), \Gamma_n^k(2), \dots, \Gamma_n^k(C_n^k)$ and $L_n^k(1), L_n^k(2), \dots, L_n^k(C_n^k)$ as Γ_n^k *sequence* and L_n^k.sequence respectively.

We proceed to justification of algorithms Γ_n^k and L_n^k. Having this goal for later use we formulate without proof the following simple lemmas.

Lemma 1.

$$\forall jk\,(1 \leqslant j \leqslant k \,\&\, 1 \leqslant k \leqslant n \,\&\, A \in R_n^k \rightarrow j \leqslant (A)_j \leqslant j + n - k).$$

Corollary 1.1.

$$\forall kA\,(1 \leqslant k \leqslant n \,\&\, A \in R_n^k \rightarrow \{1, \ldots, k\} \leqslant A \leqslant \{1 + n - k, \ldots, n\}$$

C o r o l l a r y 2.1.

$$\forall xjkA\,(1 \leqslant j \leqslant x \leqslant k \,\&\, A \in R_n^k \& (A)_j = j + n - k \rightarrow (A)_x = x + n - k).$$

C o r o l l a r y 3.1.

$$\forall xjkA\,(1 \leqslant x \leqslant j \leqslant k \,\&\, A \in R_n^k \& (A)_j < j + n - k \rightarrow (A)_x < x + n - k).$$

L e m m a 2. *The sequence of Δ_n^k-bundles $A_1, \ldots, A_m$, where $0 < k < n$, $A_1 < A_2 < \cdots < A_{m+}$, is +compacted iff the following conditions hold:*

$$\forall p \exists b \forall yzu\,(1 \leqslant p \leqslant m-1 \,\&\, q < y \leqslant k \,\&\, 1 \leqslant z < q \leqslant u \leqslant k \rightarrow (A_p)_y =$$
$$= y + n - k \,\&\, (A_p)_z = (A_{p+1})_z \,\&\, (A_{p+1})_u = (A_p)_q + u - q + 1). \qquad (1)$$

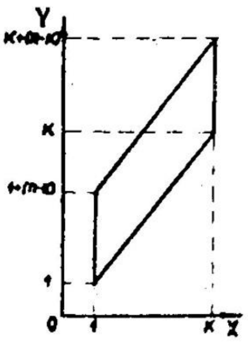
L e m m a 3. *The set R_n is inverse isomorphic to itself in relation to lexicographical ordering under such mapping R_n to itself that complement of any Δ_n-bundle to I_n corresponds to an arbitrary Δ_n-bundle.*

R e m a r k.

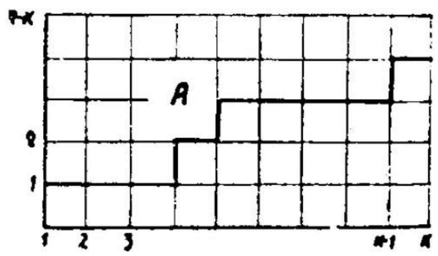
While studying the quantitative characteristics of Δ_n^k-bundles it is effective to use the following geometric interpretation.

Let's depict some Δ_n^k bundle A with fixed k, $0 \leq k \leq n$ either as a complex of points with coordinates $(1,\ (A)_1, (2,\ (A)_2, \ldots, (k,\ (A_k))$, on a two-dimensional plane or as segments of a piecewise linear curve connecting the same points.

Thus the set of all Δ_n^k-bundles will be enclosed in a rectangle with the vertices $(1,1), (1,1+n-k)$, $(k, k+n-k)$, (k,k) (pic. 1).



Pic. 1.



Pic. 2.

The latter can be considered independently in the form of a rectangular grid (pic.2), where some Δ_n^k-bundle A is depicted by means of points with coordinates $(1, (A)_1 - 1), (2, (A)_2 - 2), \ldots, (k, (A)_k - k)$or the corresponding piecewise linear curve.

Under this presentation the counting Δ_n^k-bundles with definite features is replaced with counting of the number of non-decreasing polygonal chains with corresponding properties.

It should be noted that tasks of similar type are described in [7, 8].

3. Δ-GRAPHS

This paragraph has a supplementary significance. A definition of special class of graphs, Δ-graphs and a number of its properties is investigated.

In particular, it is demonstrated that Δ-graphs can be produced inductively. This fact is pivotal in the proving of the main theorem.

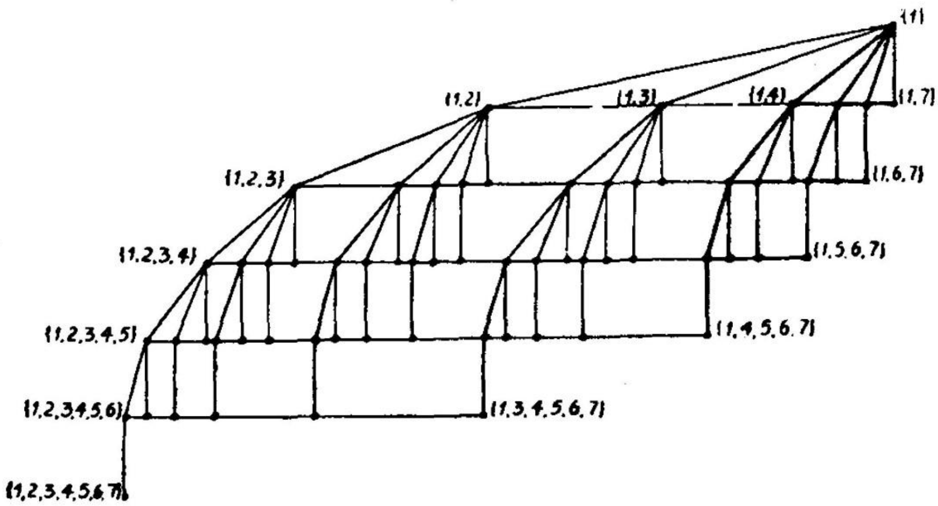
Several quantitative regularities for definite complexes of vertices of Δ-graphs which will be significantly used in further constructions are referred.

Let A be an arbitrary Δ_n-bundle, P_A be the set of all extensions of A.

We will identify Δ_n*-graph with the root* A a graph G defined in the following way.

1. Each Δ_n-bundle B, entering the set P_A, is a vertex marked B in graph G.
2. Vertices B и C of the graph G are connected an edge only in case one of following conditions holds.
3. a)$|B| = |C|$ *and* B, C *or* C, B *are* $+$ -compacted sequences of Δ_n-bundles;

б) C is $|C|$-extension of B and $|C| = |B| + 1$ or B is $|B|$-extension of C and $|B| = |C| + 1$.



Pic. 3.

Picture 3 depicts one of such graphs, Δ_7-graph with the root $\{1\}$.

It is easy to prove the following lemma.

L e m m a 4. *If* B *is any vertex of* Δ_n*-graph* G *with the root* A*, where* A *– any* Δ_n*-bundle, then the* Δ_n*-graph with the root* B *is a subgraph of* G.

Let G *be* Δ_n-graph with the root A, where A is some Δ_n-bundle, $(A)_{|A|} = a$. Then we will call $\delta\Delta_n^{(p)}$*-graph with the root* A the subgraph G_p of the graph G such that $G_p = G \setminus \bigcup_{i=1}^{p} G_i$ where G_i is a Δ_n-graph with the root $A \cup \{a+i\}$, $i = 1, \dots, p$, $0 \le p \le n - a$.

Let G be an arbitrary $\delta\Delta_n^{(p)}$-graph with the root A, we denote $\{G\}$ the set of all vertices of G.

Observe that $\delta\Delta_n^{(0)}$-graph with the root A coincides with Δ_n-graph with the root A, while $\delta\Delta_n^{(n-a)}$-graph with the root A is a graph with a single vertex A.

The $\delta\Delta_7^{(2)}$-graph with the root $\{1\}$ is marked with bold lines on pic. 3. The set of all $\delta\Delta_n^{(p)}$-graphs with isomorphism among graphs is partitioned to non-trivial equivalence classes.

Before proving this statement we indicate the following lemma.

L e m m a 5. *For an arbitrary* $\Delta_n^{k_1}$*-bundle* A*, if* B *is* k_2 *–extension fo* A*, then* $B = A \cup C$*, where* $0 \le k_1 \le k_2 < n$ *and* C *is some* $\Delta_n^{k_2-k_1}$*-sub-bundle of bundle* $\{(A)_{k_1}+1,\ (A)_{k_1}+2, \dots, n\}$ *and conversely, if* D *is an arbitrary* $\Delta_n^{k_2-k_1}$*-sub-bundle of bundle* $\{(A)_{k_1}+1, \dots, n\}$*, then* $A \cup D$ *is a* k_2*-extension of* A.

C o r o l l a r y 1.5. *An arbitrary* $\Delta_n^{k_1}$*-bundle has exactly* $C_{n-(A)_{k_1}}^{k_2-k_1}$ k_2*-extensions, where* $0 \le k_1 \le k_2 \le n$.

C o r o l l a r y 2.5. k_2*-extension of the set of* $\Delta_n^{k_1}$*-bundles* $\{A_1, A_2, \dots, A_m\}$ *has exactly* $\sum_{j=1}^{m} C_{n-(A_j)_{k_1}}^{k_2-k_1}$ $\Delta_n^{k_2}$*-bundles, where* $0 \le k_1 \le k_2 \le n$.

L e m m a 6.* *For an arbitrary* Δ_n*-bundle of* A, *if* G *is some* $\delta\Delta_n^{(p)}$*-graph with the root* A, $(A)_{|A|} = a$, *then:*

1. *Graph* G *is isomorphic to* Δ_{n-a-p}*-graph* F, *the root of which is an empty* Δ_{n-a-p}*-bundle: the isomorphism provides a bijective mapping* θ *between vertices of* G *and* F *which is unique:*

$$2.\ \forall BC\,(B \in \{G\}\ \&\ C \in \{F\} \rightarrow ((B = A \rightarrow \theta(B) = \varnothing)\ \&$$

$$\&\ (C = \varnothing \rightarrow \theta^{-1}(C) = A)\ \&\ (B \neq A \rightarrow \theta(B) = \{(B)_{|A|+1} - a - p,$$

$$\dots, (B)_{|B|} - a - p\} \in \{F\})\ \&\ (C \neq \varnothing \rightarrow \theta^{-1}(C) =$$

$$= A \cup \{(C)_1 + a + p, \dots (C)_{|C|} + a + p\} \in \{G\})));$$

$$3.\ \forall BC\,(B \in \{G\}\ \&\ C \in \{F\} \rightarrow |\theta(B)| = |B| - |A|\ \&\ |\theta^{-1}(C)| =$$

$$= |C| + |A|;$$

1. Assume $\forall B_1, B_2, C_1, C_2$, *if* $B_1, B_2 \in \{G\}$ and $C_1, C_2 \in \{F\}$, B_2 *is a* $|B_2|$*-extension of* B_1 *and* C_2 *is a* $|C_2|$*-extension of* C_1, *then* θB_2 *will be* $(|B_2| - |A|)$*-extension* of θB_1 *and* $\theta^{-1}(C_2) - (|C_2| + |A|$ *-extension of* $\theta^{-1}(C_1)$;
2. *If* $B_1, \dots, B_m$ *and* $C_1, \dots, C_r$ are $+$ *-compacted sequences of bundles from* $\{G\}$ *and* $\{F\}$ *respectively, then* $\theta(B_1), \dots, \theta(B_m)$ *and* $\theta^{-1}(C_1), \dots, \theta^{-1}(C_r)$ *are* $+$ *-compacted sequences from* $\{F\}$ *and* $\{G\}$ respectively.

P r o o f.

If $\{G\} = A$, then the assertion of lemma is evident.

Assume $|\{G\}| > 1$.

Let's explore the graph G', obtained from the graph G with substitution of its each vertex to a new vertex defined by $B' = B \cup \{a + p\} \setminus \{a\}$ and assume x be the corresponding bijective mapping between vertices of G and G'.

The assertions on relations of lexicographical ordering, extension, $+$ -compactness and numerical estimates corresponding to such relations which are true for arbitrary vertices of G' match to analogous assertions for corresponding vertices from G under mapping x.

Hence it is sufficient to prove the assertions of lemma 6* for the graph G'.

Graph G' is a Δ_n-graph with the root $A' = A \cup \{a + p\} \setminus \{a\}$, the conclusion follows from the definition of Δ_n-graphs, preceding remark and observation that the set of vertices of the graph G' matches with the set of all extensions of A'.

Note also that $||C'|| = 2^{n-a-p}$.

Consider a lexicographically ordered set of $\Delta_{n-a-p,\ a+p}$-bundles $R_{n-a-p,\ a+p}$.

It is recognized that $|R_{n-a-p,\ a+p}| = 2^{n-a-p}$ and the set $R_{n-a-p,\ a+p}$ coincides with the set of all sub-bundles $I_{n-a-p,\ a+p} = \{a + p + 1, \dots, n\}$. Define a mapping θ_1 of the set $\{G'\}$ to $R_{n-a-p,\ a+p}$ in the following way: $\forall B(B \in \{G'\} \rightarrow \theta_1(B) = B \setminus A')$. It follows from lemma 5 that $\forall B\big(B \in \{G'\} \rightarrow \theta_1(B) \in R_{n-a-p,\ a+p}\big)$.

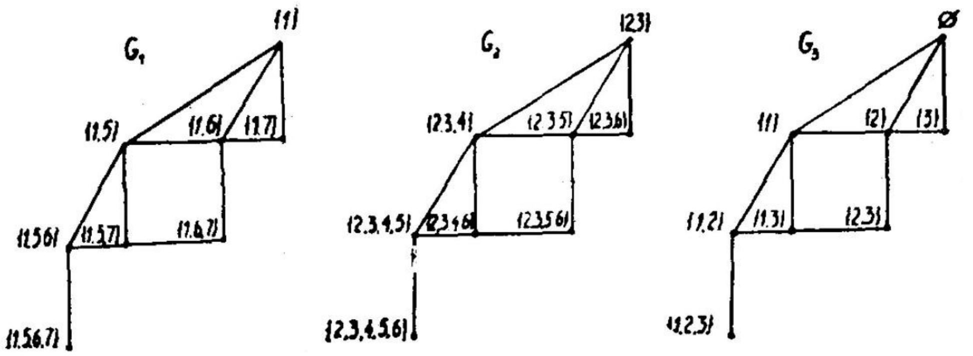
The mapping θ_1 will be bijective since images of different elements from $\{G'\}$ are different under the mapping θ_1 and $|G'| = R_{n-a-p,\ a+p}$.

It can be easily seen that under the mapping θ_1 sets $\{G'\}$ and $R_{n-a-p,\ a+p}$ are isomorphic in relation to lexicographical ordering and extension.

Since each of the sets $\{G'\}$ and $R_{n-a-p,\ a+p}$ is lexicographically ordered in a unique way and the ordering is linear the mapping of $\{G'\}$ to $R_{n-a-p,\ a+p}$ preserving the ordering is also unique.

It is not difficult to prove also that the indicated mapping of $\{G'\}$ to $R_{n-a-p,\ a+p}$ matches the $+$ -compacted sequence from $\{G'\}$ with $+$compacted sequences from $R_{n-a-p,\ a+p}$, and the contrary.

Taking into consideration the relation between sets $R_{n-a-p,\ a+p}$ and R_{n-a-p}, established in Lemma 0, and above-stated reasoning we come to the statement of Lemma 6*.



Pic. 4.

Picture 4 depicts three isomorphic three isomorphic each other graphs: $\delta\Delta_7^{(3)}$-graph G_1 with the root {1}, Δ_6-graph G_2 with the root {2,3} and $\Delta_3^{(0)}$-graph G_3, the root of which is an empty Δ_3-bundle.

Assume $0 \le a \le n$. Denote the set of Δ_n-bundles A, such that $(A)_{|A|=a}$ as P_a and the set of all Δ_n-graphs the roots of which belong to P_a as $\tilde{E}_{(a)}$.

An arbitrary element of the set $\tilde{E}_{(a)}$, $0 \le a \le n$, we identify as $\square_n^a$ *-graph.*.

The elements of sets $\tilde{E}_n = \bigcup_{a\in\{0,1,\dots,n\}} \tilde{E}_a$? and $\tilde{E} = \widetilde{\bigcup_{n=0}^{\infty}} \tilde{E}_n$? we will call Δ_n*-graphs* and Δ-*graphs*, respectively.

Thus, assume also $\widetilde{H}_{a,p}$ be the set of all possible $\delta\Delta_n^{(p)}$-graphs the roots of which belong to P_a, $\widetilde{H}_a = \bigcup_{p=0}^{n-a} \widetilde{H}_{a,p}$, $\widetilde{H}_n = \bigcup_{a\in\{0,1,\dots,n\}} \widetilde{H}_{(a)}$, $\widetilde{H} = \bigcup_{n=0}^{\infty} \widetilde{H}_n$. ?

Elements of sets $\tilde{E}_n = \bigcup_{a\in\{0,1,\dots,n\}} \tilde{E}_a$? and $\tilde{E} = \bigcup_{n=0}^{\infty} \tilde{E}_n$? we will call Δ_n*-graphs* and Δ*-graphs.*

As it follows from lemma 6* the set of all $\delta\Delta$-graphs H in relation to isomorphism between graphs can be partitioned to equivalence classes such that class H_c consists from such $\delta\Delta_n^{a,p}$ -graphs and only such $\delta\Delta_n^{a,p}$ -graphs, for which $n - a - p = c$, где $p \in \{0, 1, \dots, n-a\}$, $a \in \{0, 1, \dots, n\}$, $n = 0, 1, 2, \dots$, $c = 0, 1, 2, \dots$.

Lets introduce several notions for $\delta\Delta$-graph$G$.

Assume G *be* $\delta\Delta_n^{(p)}$*-graph* with the root $A$and $a = (A)_{|A|}$. Then:

1. Δ_n-graph with the root $A \cup \{a + p + 1\}$ we will call the *left sub-graph* of G, while $\delta\Delta_n^{(p+1)}$-graph with the root A —*right sub-graph*G;

9. Vertices of the left and right sub-graphs of $\delta\Delta_n$-graph$G$we will call the *left* and *right* vertices G, resepectively*;*

10. The set of all $(a + r)$-extensions of Δ_n-bundle$A$, belonging G, where $0 \le r \le n - a$, we will call an$r$*-tier*G.

In an arbitrary $\delta\Delta$*-graph*G, having only one vertex we identify the left and right sub-graphs Gequal to G.

In an arbitrary graph F, isomorphic to $\delta\Delta$*-graph* G, we will call the left, right sub-graphs, left, right vertices, r-tier the parts of F, that are in one to one correspondence with the parts of G having the same label.

The following corollary immediately follows from point 1, lemma 6*.

Corollary 1.6*. In an arbitrary $\delta\Delta$*-graph G the left sub-graph$G$ is isomorphic to the right sub-graph* G.

We will conditionally depict an arbitrary $\delta\Delta$-graph G in the shape of an ellipse (pic. 5), at which point A indicates the root G, horizontal lines are drawn for tiers G.

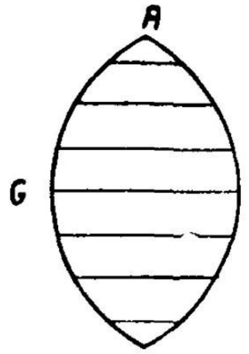
It follows from corollary 1.6* that provided G is a $\delta\Delta_n^{a,p}$-graph it can be drawn by means of two $\delta\Delta_{n-1}$-graphs G_1 and G_2 with roots A_1 and A_2, respectively, connected with an edge (A_1, A_2). Observe that $A_1$is the vertex of 1-tier G (pic. 6). $G_1$and G_2 are also $\delta\Delta$-graphs, the indicated process of $\delta\Delta$-graphs partition can be continued. At the final stage of partition we obtain $\delta\Delta$-graphs having only one vertex which is the root.

It is clear that in $\delta\Delta_n^{a,p}$-graph G the partition allows differentiating $2^{n-a-p-1}$ $\Delta-$ sub-graphs isomorphic to $\Delta_i^{(0)}$ –graph, where $i = 0, 1, \dots, n-a-p$.

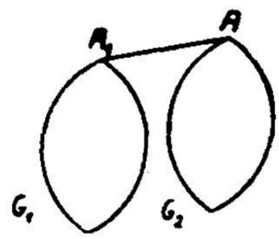
We define several quantitative regularities specifying the number of extensions of some sets of vertices of Δ-graphs.

Let E_n^k be the set of all subsets R_n^k and $E_n = \bigcup_{h=0}^{n} E_n^k$. ?

On the set E_n we define a function $\Psi_k(x)$ such that fir each $S \in E_n$, $\Psi_k(S)$ is equal to the number of all Δ_n^k-bundles in k-continuation of S.



Pic. 5.



Pic. 6.

It is implied from distinction of k-extension of two arbitrary distinct Δ_n-bundles that the function $\Psi_k(S)$ is additive, consequently the following lemma is true.

L e m m a 7. *For an arbitrary set of* $\Delta_n^{k_1}$*-bundles* S*, if* $S \in S_1 \cup S_2$ *and* $S_1 \cap S_2 = \emptyset$, *then for each* k_2, *where* $0 \le k_1 \le k_2 \le n$, *we have:*

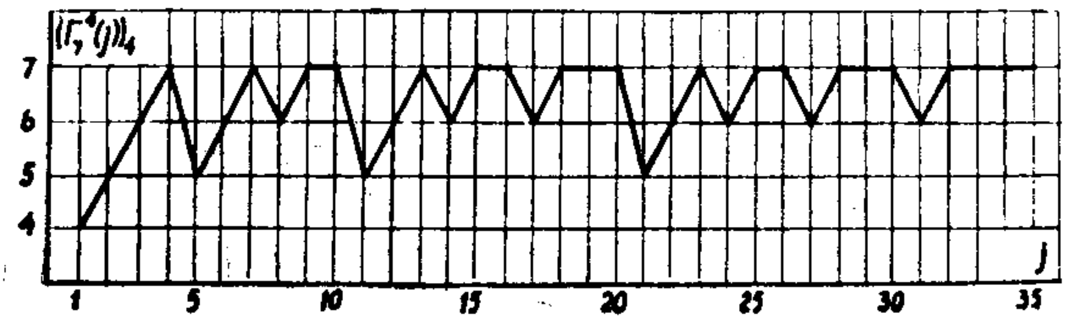
$$\Psi_{k_2}(S) = \Psi_{k_2}(S_1) + \Psi_{k_2}(S_2).$$

In lemma 8* referred below we will prove two inequalities concerning the number of elements k_2-extensions of +-compacted sequences. The following features are peculiar to those inequalities.

11. The indicated inequalities rather often become equalities and hence discarding any different from zero elements in the process of the proof is unacceptable.

12. As it has been stated in corollary 2.5, the number of elements in k_2-continuation of some given sequence of $\Delta_n^{k_1}$-bundles $A = A_1, \dots, A_m$ is defined by means of numbers $(A_1)_{k_1}, \dots, (A_m)_{k_1}$.

However even in case A is a +-compacted sequence the nature of change in the numbers $(A_1)_{k_1}, \dots, (A_m)_{k_1}$ is significantly non-linear which leads to essential difficulties in computations, resulting from the indicated regularities.



Pic. 7.

The peculiarities stated above provided significant impact on determining of the method of proof of the lemma, namely the method of induction on Δ-graphs, meanwhile the set of all possible cases that require examination is partitioned to distinct classes each of which assumes application of methods common for the elements of the picked class, but different for different classes.

We introduce the following notation. Let **A** be any sequence of bundles, we denote {**A**} the set of all bundles of **A**.

L e m m a 8*[2].

$$\forall k_1 k_2 m r\,(0 \leqslant k_1 < k_2 \leqslant n \;\&\; 1 \leqslant m \leqslant C_n^{k_1} \;\&\; 1 \leqslant r \leqslant C_n^{k_1} - m + 1 \;\&\; \mathbf{A} =$$
$$= \Gamma_n^{k_1}(r), \;\ldots, \Gamma_n^{k_1}(r+m-1) \;\&\; \mathbf{A}^{*} = \Gamma_n^{k_1}(1), \ldots, \Gamma_n^{k_1}(m) \;\&\; \mathbf{A}^{**} =$$
$$= \Gamma_n^{k_1}(C_n^{k_1} - m + 1), \ldots, \Gamma_n^{k_1}(C_n^{k_1}) \rightarrow \psi_{k_2}(\{\mathbf{A}^{*}\}) \geqslant \psi_{k_2}(\{\mathbf{A}\}) \geqslant \psi_{k_2}(\{\mathbf{A}^{**}\}))^{1}.$$

First we will prove the following lemmas 9–11[3].

L e m m a 9. *k_2-extension $\boldsymbol{B} = B_1, \ldots, B_m$ of an arbitrary $\Delta_n^{k_1}$-bundle A, where* $0 \leq k_1 \leq k_2 \leq n$, *is +compacted. At the same time if $A = \Gamma_n^{k_2}(I)$, then $B_j = \Gamma_n^{k_2}(j)$ under j=1,…,m.*

P r o o f.

If either $m = 0$ or $k_2 = k_1$ the lemma is evident.

Suppose $m > 0$ and $k_1 < k_2$. Since B *is* k_2-extension of A, then proceeding from the definition k_2-extension, we obtain $B_1 \prec B_2 \prec \cdots \prec B_m$.

Assume **B** is not a $+$-compacted sequence. This means that at some $p \in \{1, \ldots, m-1\}$ the condition for $+$-compactness of **B** is violated, then there exists a bundle $B \in R_n^{k_2}$ such that

$$B_p \prec B \prec B_{p+1}. \qquad (1)$$

Hence B is k_2-extension of A, since it follows from (1) that:

$$A = \{(B_p)_1, \ldots, (B_p)_{k_1}\} = \{(B_{p+1})_1, \ldots, (B_{p+1})_{k_1}\} = \{(B)_1, \ldots, (B)_{k_1}\},$$

and B does not coincide with any from $\Delta_n^{k_1}$-bundles **B**, since an arbitrary bundle from **B** is either less than B_p or greater than B_{p+1}. This implies that not all k_2- extensions of A are included in **B** which is false since **B** is k_2- extensions of A. Consequently our assumption is not true and **B** is + -compacted sequence.

If $A = \Gamma_n^{k_1}(1)$ then the bundle $\Gamma_n^{k_2}(1)$ is k_2-extension of A. Since $\Gamma_n^{k_2}(1)$ is the least bundle in the sequence $\Gamma_n^{k_2}$ it will be the least in the sequence **B** as well, that is $\mathrm{B}_1 = \Gamma_n^{k_2}(1)$. Besides **B** is a +-compacted sequence we conclude that **B** coincides with the beginning segment of $\Gamma_n^{k_2}$.

Lemma 9 is proved.

L e m m a 10. *For an arbitrary $\delta\Delta$-graph G, if B is some vertex of G distinct from the root G then Δ-graph with the root B is a sub-graph of G.*

P r o o f.

I f $Gis\ A\ a$ Δ-graph, then the statement of lemma 9 reduces to lemma 4.

Let G be a $\delta\Delta_n^{(p)}$-graph with the root A, where $0 < p < n - a,\ a = (A)_{|A|}$ and F be Δ_n-graph with the root A.

According the definition of $\delta\Delta_n^{(p)}$-graph with the root A,

---

[2] As it is shown in [9], the assertion of lemma 8* can be generalized to the case of the words in n-letter alphabet with the number of each letter entry not exceeding a predetermined integer (individual for each letter). In Δ_n-bundles considered in lemma 8*, each letter (integer from 1 to n) has only one entry.

$$G = F \setminus \bigcup_{i=1}^{p} G_i,$$

Where G_i is a Δ_n-graph with the root $A \cup \{a+i\}$, $i = 1, \dots, p$.

Obviously the Δ_n-graphs G_q and G_r with roots $A \cup \{a+q\}$ and $A \cup \{a+r\}$, respectively, where $r, q \in \{1, \dots, n-a-1\}$, and $r \neq q$ do not have common vertices. This is the ground that deletion of one of the graphs for example G_q from F does not change connections between vertices of G_r. This implies that provided $B \in \{A \cup \{a+p+1\}, \dots, A \cup \{a+n-a-1\}$ Δ_n-graph with the root B will be according lemma 4, a sub-graph of F, hence it will be a sub-graph of G.

If B is a vertex of Δ_n-graph G_j with the $A \cup \{a+j\}$, where $j \in \{p+1, \dots, n-a\}$, then Δ_n-graph with the root B will be according lemma 4, a sub-graph G_j, hence a sub-graph of G.

The lemma is proved.

Corollary 1.10.

If F and G are arbitrary $\delta\Delta$-graphs such that G is a sub-graph of F, S is an arbitrary set of bundles of the equal lengths fromв G, different from the root G, then for any natural number k all bundles of k-extension S will be simultaneously vertices of G and F.

Remark. Corollary 1.10 is true for the root G, if G is Δ-graph.

Thus, if $\delta\Delta$-graph G is a sub-graph of $\delta\Delta$-graph of F and S is a set of bundles of the identical length the same for $\{F\}$ and $\{G\}$, then examinations concerning extensions of bundles from S can be conducted in one of these graphs, but the derived statements can be transferred to the other graph without any modifications.

The following statement is implied from point 4 lemma 6.

Corolllary 2.6*. *Assume G_1 and G_2 be arbitrary mutually isomorphic $\delta\Delta$-graphs, A_1 and A_2 be respectively roots of G_1 and G_2, S_1 – arbitrary set of bundle of fixed length from $\{G_1\}$, distinct from A_1, S_2 – set of corresponding bundles from $\{G_2\}$, then the following equalities hold for any natural nuтber k:*

$$\Psi_{k+|A_1|}(S_1) = \Psi_{k+|A_2|}(S_2).$$

Remark. Corollary 2.6* is true for the roots of A_1 and A_2 provided we consider not all $(k+|A_1|)$-extensions of A_1 and respectively $(k+|A_2|)$-extension of A_2, but only those that belong $\{G_1\}$ and respectively $\{G_2\}$.

The assertions of lemma 6*for the case of Δ_n^0-graphs are essentially used in the proof of lemma 8*, as well as in further inferences. For later use we will state the results of lemma 6* for the special case of Δ_n^0-graphs in independent lemma supplemented with some properties of Δ_n^0-graphs.

Let's introduce a function $v(x, y, z)$, that takes value $v(S, G_1, G_2)$ equal to the set of all vertices G_2 corresponding of vertices S under isomorphism G_1 and G_2.

It is immediately implied from assertion 1 of lemma 6* that function $v(x, y, z)$ is uniquely defined.

If S is a single element set for conciseness we will write simply «bundle $v(S, G_1, G_2)$» instead of expression «bundle from the set $v(S, G_1, G_2)$».

Introduce the following denotations.

If G_n is $\Delta_n^{(0)}$-graph then we will denote G_n^L and G_n^R the left and right sub-graphs G_n respectively, while the left and right sub-graphs of G_n^L (G_n^R) we identify as G_n^{LL} and G_n^{LR} (G_n^{LR} and G_n^{RR}).

Лемма 11. *If G_n is $\Delta_n^{(0)}$-graph then:*

1. *Graphs G_n^L and G_n^R are isomorphic to $\Delta_{n-1}^{(0)}$-graph G_{n-1}, graphs G_n^{ll}, G_n^{lr}, G_n^{rl}, G_n^{rr} to Δ_{n-2}^0 graph G_{n-2};*

under isomorphism G_n^L and G_{n-1}(*respectively* G_n^R *and* G_{n-1}) *to sub-graphs* G_n^{ll} *and* G_n^{lr} *(to subgraphs* G_n^{rl} , G_n^{rr} *, respectively), to subgraphs* G_n^{ll} *and* G_n^{lr} *(to subgraphs* G_n^{rl} , G_n^{rr} *, respectively), in the graph* G_{n-1}*, correspond subgraphs* G_{n-1}^{l} *and* G_{n-1}^{r} *, respectively.*

2. а) $\{G_n^л\} = \bigcup\limits_{k=1}^{n} \{\Gamma_n^k(1), \ldots, \Gamma_n^k(C_{n-1}^{k-1})\}$,

б) $\{G_n^п\} = \bigcup\limits_{k=0}^{n-1} \{\Gamma_n^k(C_{n-1}^{k-1}+1), \ldots, \Gamma_n^k(C_n^k)\}$,

в) $\{G_n^{лл}\} = \bigcup\limits_{k=2}^{n} \{\Gamma_n^k(1), \ldots, \Gamma_n^k(C_{n-2}^{k-2})\}$,

г) $\{G_n^{лп}\} = \bigcup\limits_{k=1}^{n-1} \{\Gamma_n^k(C_{n-2}^{k-2}+1), \ldots, \Gamma_n^k(C_{n-1}^{k-1})\}$,

д) $\{G_n^{пл}\} = \bigcup\limits_{k=1}^{n-1} \{\Gamma_n^k(C_{n-1}^{k-1}+1), \ldots, \Gamma_n^k(C_{n-1}^{k-1}+C_{n-2}^{k-1})\}$,

е) $\{G_n^{пп}\} = \bigcup\limits_{k=0}^{n-2} \{\Gamma_n^k(C_{n-1}^{k-1}+C_{n-2}^{k-1}+1), \ldots, \Gamma_n^k(C_n^k)\}$;

13.a) *for arbitrary* k, f, s, r, *if* $0 \le s \le n-1$ *и* $1 \le r \le C_{n-1}^{s}$, *then:*

$$1)\ 1 \le k \le n \,\&\, 1 \le t \le C_{n-1}^{k-1} \to \nu(\{\Gamma_n^k(t)\}, G_n^л, G_{n-1}) = \\ = \{\Gamma_{n-1}^{k-1}(t)\} \,\&\, \nu(\{\Gamma_{n-1}^s(r)\}, G_{n-1}, G_n^л) = \{\Gamma_n^{s+1}(r)\};$$

$$2)\ 0 \le k \le n-1 \,\&\, C_{n-1}^{k-1}+1 \le t \le C_n^k \to \nu(\{\Gamma_n^k(t)\}, G_n^п, G_{n-1}) = \\ = \{\Gamma_{n-1}^k(t-C_{n-1}^{k-1})\} \,\&\, \nu(\{\Gamma_{n-1}^s(r)\}, G_{n-1}, G_n^п) = \{\Gamma_n^s(r+C_{n-1}^{s-1})\};$$

б) for arbitrary $k, t, s,$ г, *if* $0 \le s \le n-2$ and $1 \le r \le C_{n-2}^{s}$, then*:*

$$1)\ 2 < k \le n \,\&\, 1 \le t \le C_{n-2}^{k-2} \to \nu(\{\Gamma_n^k(t)\}, G_n^{лл}, G_{n-2}) = \\ = \{\Gamma_{n-2}^{k-2}(t)\} \,\&\, \nu(\{\Gamma_{n-2}^s(r)\}, G_{n-2}, G_n^{лл}) = \{\Gamma_n^{s+2}(r)\};$$

$$2)\ 1 \le k \le n-1 \,\&\, C_{n-2}^{k-2}+1 \le t \le C_{n-1}^{k-1} \to \nu(\{\Gamma_n^k(t)\}, C_n^{лп}, G_{n-2}) = \\ = \{\Gamma_{n-2}^{k-1}(t-C_{n-2}^{k-2})\} \,\&\, \nu(\{\Gamma_{n-2}^s(r)\}, G_{n-2}, G_n^{лп}) = \{\Gamma_n^{s+1}(r+C_{n-2}^{s-1})\};$$

$$3)\ 1 \le k \le n-1 \,\&\, C_{n-1}^{k-1}+1 \le t \le C_{n-1}^{k-1}+C_{n-2}^{k-2} \to \nu(\{\Gamma_n^k(t)\}, G_n^{пл}, \\ G_{n-2}) = \{\Gamma_{n-2}^{k-1}(t-C_{n-1}^{k-1})\} \,\&\, \nu(\{\Gamma_{n-2}^s(r)\}, G_{n-2}, G_n^{пл}) = \{\Gamma_n^{s+1}(r+C_{n-1}^{s})\};$$

$$4)\ 0 \le k \le n-2 \,\&\, C_{n-1}^{k-1}+C_{n-2}^{k-1}+1 \le t \le C_n^k \to \nu(\{\Gamma_n^k(t)\}, G_n^{пп}, \\ G_{n-2}) = \{\Gamma_{n-2}^k(t-C_{n-1}^{k-1}-C_{n-2}^{k-2})\} \,\&\, \nu(\{\Gamma_{n-2}^s(r)\}, G_{n-2}, G_n^{пп}) = \\ = \{\Gamma_n^s(r+C_{n-1}^{s-1}+C_{n-2}^{s-1})\};$$

14.*For arbitrary* k, l, s, r:

а) $1 \leqslant k \leqslant n \,\&\, 1 \leqslant t \leqslant C_{n-1}^{k-1} \,\&\, 0 \leqslant s \leqslant n-1 \,\&\, C_{n-1}^{k-1}+1 \leqslant r \leqslant C_n^k \rightarrow$

$$\rightarrow \nu(\{\Gamma_n^k(t)\},\ G_n^{\text{л}},\ G_n^{\text{п}}) = \{\Gamma_n^{k-1}(t + C_{n-1}^{k-2})\} \,\&\, \nu(\{\Gamma_n^s(r)\},\ G_n^{\text{п}},\ G_n^{\text{л}}) =$$

$$= \{\Gamma_n^{s+1}(r - C_{n-1}^{s-1})\};$$

б) $2 \leqslant k \leqslant n \,\&\, 1 \leqslant t \leqslant C_{n-2}^{k-2} \,\&\, 1 \leqslant s \leqslant n-1 \,\&\, C_{n-2}^{s-2}+1 \leqslant r \leqslant C_{n-1}^{s-1} \rightarrow$

$$\rightarrow \nu(\{\Gamma_n^k(t)\},\ G_n^{\text{лл}},\ G_n^{\text{лп}}) = \{\Gamma_n^{k-1}(t + C_{n-2}^{k-3})\} \,\&\, \nu(\{\Gamma_n^s(r)\},\ G_n^{\text{лп}},\ G_n^{\text{лл}}) =$$

$$= \{\Gamma_n^{s+1}(r - C_{n-2}^{s-2})\};$$

в) $1 \leqslant k \leqslant n-1 \,\&\, C_{n-1}^{k-1}+1 \leqslant t \leqslant C_{n-1}^{k-1} + C_{n-2}^{k-1} \,\&\, 0 \leqslant s \leqslant n-2 \,\&$

$$\& \, C_{n-1}^{s-1} + C_{n-2}^{s-2} + 1 \leqslant r \leqslant C_n^s \rightarrow \nu(\{\Gamma_n^k(t)\},\ G_n^{\text{пл}},\ G_n^{\text{пп}}) =$$

$$= \{\Gamma_n^{k-1}(t - C_{n-1}^{k-1} + C_{n-1}^{k-2} + C_{n-2}^{k-2})\} \,\&\, \nu(\{\Gamma_n^s(r)\},\ G_n^{\text{пп}},\ G_n^{\text{пл}}) =$$

$$= \{\Gamma_n^{s+1}(r - C_{n-1}^{s-1} - C_{n-2}^{s-1} + C_{n-1}^{s})\}.$$

P r o o f.

I. The isomorphism of graphs stated in point 1 immediately follows from assertion 1 lemma 6*.

We will prove that sub-graphs $G_{n-1}^l$and G_{n-1}^r correspond to sub-graphs G_n^{ll} and G_n^{lr}under isomorphism G_1^l and G_{n-1} .

Indeed only two sub-graphs, namely sub-graphs G_n^{ll} and G_n^{lr}, isomorphic to $\Delta_{n-1}^{(0)}$-graph G_{n-2} can be identified in Δ-graph графе G_n^l. This follows from the fact that for an arbitrary bundle B from $\{G_n^l\}$distinct from {1, 2}({1, 2}- the root of G_n^{ll}) and {1} ({1} –the root of G_n^{lr}) and $\Delta_n^{a,p}$sub-graph $G_B$with the root B the difference $n-a-p$ is less than $n-2$, thus proceeding from point 1 in lemma 6* we conclude that the graph $G_B$will not be isomorphic to G_{n-2}.

Consequently G_n^{ll} corresponds either to G_{n-1}^l or to G_{n-1}^r under isomorphism G_n^l and G_{n-1}.

Since G_{n-1}^l and G_{n-1}^r as well as G_n^{ll} and G_n^{lr} do not have common vertices while the vertex $\{1\} \in G_{n-1}^l$ corresponds to $\{1,2\} \in G_n^{ll}$ (lemma 6*, point 2) under isomorphism G_n^l and G_{n-1}, then G_n^l corresponds to $G_{n-1}^l$and G_n^{lr} to G_{n-1}^r under indicated isomorphism.

The corresponding assertion for graphs G_n^r and G_{n-1} is proved in similar way. Obviously it is sufficient to prove the statement in point 2 for sub-parts stated in a,c,e.

a. The vertex {I} and {1} = $\Gamma_n^1(1)$ is the roots of Δ_n^1-graph of G_n^l.

It is clear that $\{G_n^l$vertex r} coincides with the set of all extensions of $\Gamma_n^1(1)$, with $k=1,\dots,n$.

The sequence $A_k = \Gamma_n^k(1),\dots,\Gamma_n^k(p)$ will be the k-extension for each k, $1 \leq k \leq n$.

The assertion follows from lemma 9.

Provided G_n^l is isomorphic to G_{n-1} we derive from point 2 of lemma 6* that sequences A_k will correspond to the sequence $\Gamma_{n-1}^{k-1}(1),\dots,\Gamma_{n-1}^{k-1}(C_{n-1}^{k-1})$. It is implied then that $p = C_{n-1}^{k-1}$.

c. This assertion is proved in analogy to the previous with the difference in taking in consideration the $\Delta_{n-2}^{(0)}$-graph G_{n-2} and graph G_n^{ll}.

e. In the sequence $\Gamma_n^k(C_{n-1}^{k-1}+1),\dots,\Gamma_n^k(C_n^k)$, which consists of bundles $\{G_n^l\}$ with $0 \leq k \leq n-1$, bundles from $\{G_n^{rl}\}$precede to bundles from $\{G_n^{rr}\}$. The assertion follows from the fact that the root $\{G_n^{rl}\}$, which is the bundle {2}, is the least from bundles of length I belonging to $\{G_n^r\}$. Thus, if , $1 \leq k \leq n-1$, $\Gamma_n^k(C_{n-1}^{k-1}+1) \in \{G_n^{\text{пл}}\}$. Using this fact through reasoning similar to that given above we got convinced in validity of the assertion in clause "c".

The assertions in 3.a are easily obtained using the isomorphism $G_n^l(G_n^r)$ and G_{n-1}, in 2.a, 2.b of lemma 11 and 2.5, lemma 6*.

Similarly the assertions in 3.b are obtained from isomorphism $G_n^{ll}(G_n^{lr}, G_n^{rl}, G_n^{rr})$ and G_{n-2} defined in clauses 2.c., 2.d, 2.e, 2.f of lemma 11 and 2 to 5 lemma 6*.

2. Vertices corresponding isomorphism G_n^l and G_{n-1}^r, G_n^{rl} and G_n^{rr} respectively, are indicated in this case.

They are easily derived by sequential application of statements in clause 3, lemma 11.

Thus the statement 4.a леммы 11 follows from isomorphism G_n^l, G_{n-1}, G_{n-1}^r and assertions in clauses 3.a, lemma 11.

Lemma 11 is proved.

It follows from 1, lemma 11 that $\Delta_n^{(0)}$-graph can be depicted as it is shown on picture 8.

P r o o f o f l e m m a 8*.

If $k_1 = k_2$, $k_1 = 0$ or $k_2 = n$ the lemma is evident.

Assume $0 < k_1 < k_2 < 0$.

We prove the lemma by induction on parameter n.

If $n = 0,1$ the lemma is valid, it is proved with immediate check.

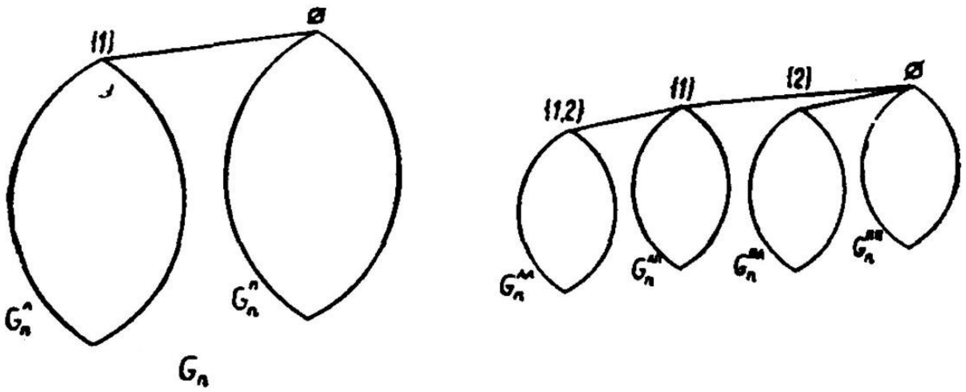
Assume lemma is valid at n equal $t - 1$ with $t \geq 1$. Let prove that the lemma will be valid at $n = t$.

Examine $\Delta_n^{(0)}$-graphs G_n provided $n = t - 1, t$. Our inductive assumption means that the lemma is true if bundles from sequences $\mathbf{A}$, $\mathbf{A}^*$, $\mathbf{A}^{**}$ belong to the set $\{G_{t-1}\}$. We aim to prove the validity of lemma for the case $\mathbf{A}$, $\mathbf{A}^*$, $\mathbf{A}^{**}$ are composed from bundles belonging to the set $\{G_t\}$.

Our assumptions on $\Psi_{k_2}(\{\mathbf{A}^*\}) \geq \Psi_{k_2}(\{\mathbf{A}\})$ and $\Psi_{k_2}(\{\mathbf{A}\}) \geq \Psi_{k_2}(\{\mathbf{A}^{**}\})$ when $n = t - 1$, we will call the first and second inductive assumptions.

1. Prove at $n = t$ the following inequality holds:

$$\Psi_{k_2}(\{\mathbf{A}^*\}) \geq \Psi_{k_2}(\{\mathbf{A}\}). \tag{0}$$



Pic. 8.

Let's partition the set of all possible cases determined with parameters m and r to subsets defined in the following way:

I. 1a.	$1 \leq m \leq C_{t-1}^{k_1-1}$	and	$1 \leq r \leq C_{t-1}^{k_1-1} - \mathrm{m} + 1$
I. 1b.	" "	and	$C_{t-1}^{k_1-1} + 1 \leq r \leq C_t^{k_1} - \mathrm{m} + 1$
I. 1c.	" "	and	$r \leq C_{t-1}^{k_1-1}$, $r + \mathrm{m} - 1 > C_{t-1}^{k_1-1}$
I. 2b.	$C_{t-1}^{k_1-1} < m \leq C_t^{k_1}$	and	$C_{t-1}^{k_1-1} + 1 \leq r \leq C_t^{k_1} - \mathrm{m} + 1$
I. 2c.	" "	and	$r \leq C_{t-1}^{k_1-1}$, $r + \mathrm{m} - 1 > C_{t-1}^{k_1-1}$

We proceed to examining each of these cases (pic.. 9)[4].

[4] For technical reasons gothic letters on pictures 9, 10 are changed **A** и **B**.

1.1a. It follows from 2a, lemma 11, that sequences **A**, **A*** are produced only from vertices of the left sub-graph G_t.

The left sub-graph G_t is isomorphic to $\Delta_{t-1}^{(0)}$-graph G_{t-1} (1, lemma 11), then using 3a1 of lemma 11, corollaries 1.10, 1.6* and the first inductive assumption we got convinced in validity of (0) for the case 1.1a.

Note also that in the process of the proof we will not specially indicate the cases where corollaries 1.10 or 1.6* are used and the roots of graphs $G_t^l, G_t^r, G_t^{ll}, G_t^{lr}, G_t^{rl}, G_t^{rr}$ are among the considered bundles due to triviality of their parsing.

1.1b. In this case the bundles of the sequence **A*** belong to $\{G_t^l\}$, while bundles from $\mathbf{A} - \{G_t^r\}$. The assertion follows from 2, lemma 11.

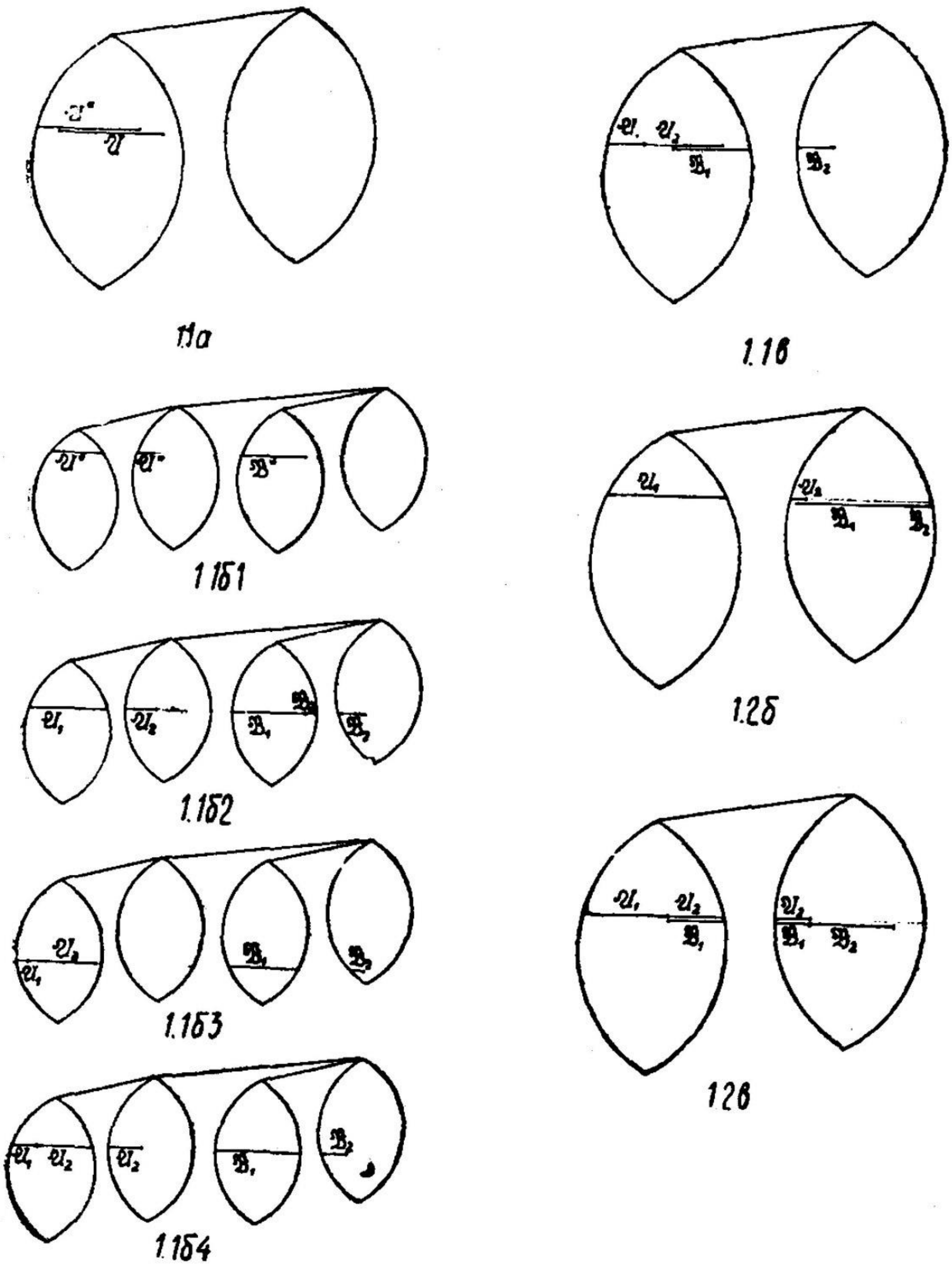
Since $G_t^{\text{п}}$ is isomorphic to $\Delta_{t-1}^{(0)}$-graph G_{t-1} (1, lemma 11), from the statements 3a 2, lemma11, corollaries I. 10, I. 6* and the first inductive assumption we conclude that

$$\Psi_{k_2}(\{\mathbf{B}^*\}) \geq \Psi_{k_2}(\{\mathbf{A}\}), \tag{1}$$

where

$\mathbf{B}^* = \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + \mathrm{m});$

$\mathbf{A} = \Gamma_t^{k_1}(\mathrm{r}), \dots, \Gamma_t^{k_1}(\mathrm{r} + \mathrm{m} - 1), \qquad C_{t-1}^{k_1} + 1 \leq \mathrm{r} \leq C_t^{k_1} - \mathrm{m} + 1.$



Pic.. 9.

Hence in case 1.1.b it is sufficient to prove that

$$\Psi_{k_2}(\{\mathbf{A}^*\}) \geq \Psi_{k_2}(\{\mathbf{B}^*\}). \tag{2}$$

We divide the case 1.1.b with respect to value m to the following subcases:

I.1b1. $1 \leq m \leq C_{t-2}^{k_1-1}$,

I.1b2. $C_{t-2}^{k_1-2} \leq C_{t-2}^{k_1-1} < m$,

I.1b3. $C_{t-2}^{k_1-1} < m \leq C_{t-2}^{k_1-2}$,

I.1b4. $C_{t-2}^{k_1-1} \leq C_{t-2}^{k_1-2} < m$.

I. 1b1. In this case all bundles $\mathbf{B}^*$ belong to $\{G_t^{rl}\}$ (lemma 11 clause 2d).

Based on statements of clause 1, lemma 11 and corollaries 1.10 and 1.6* we conclude that the sequence $\mathbf{B}'$ in G_t^{lr}can be considered instead of sequence $\mathbf{B}^*$**,** which corresponds to already studied case I. 1a.

I. 1б2. Let's represent sequences $\mathbf{A}^*$ and $\mathbf{B}^*$ in the following way:
$\mathbf{A}^* = \mathbf{A}_1, \mathbf{A}_2$; $\mathbf{B}^* = \mathbf{B}_1, \mathbf{B}_2$,
where

$$\begin{gathered} \mathbf{A}_1 = \Gamma_t^{k_1}(1), \dots, \Gamma_t^{k_1}\left(C_{t-2}^{k_1-2}\right); \\ \mathbf{A}_2 = \Gamma_t^{k_1}\left(C_{t-2}^{k_1-2} + 1\right), \dots, \Gamma_t^{k_1}(\mathrm{m}); \end{gathered} \tag{3}$$

$$\begin{gathered} \mathbf{B}_1 = \Gamma_t^{k_1}\left(C_{t-1}^{k_1-1} + 1\right), \dots, \Gamma_t^{k_1}\left(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-2}\right); \\ \mathbf{B}_2 = \Gamma_t^{k_1}\left(C_{t-1}^{k_2-1} + C_{t-2}^{k_1-2} + 1\right), \dots, \Gamma_t^{k_2}\left(C_{t-2}^{k_1-2} + \mathrm{m}\right); \end{gathered} \tag{4}$$

Since all bundles $\mathbf{B_1}$ belong to $\{G_t^{rl}\}$ (lemma 11, 2c), based on already parsed case I. 1b1, we conclude that

$$\Psi_{k_2}(\{\mathbf{A}_1\}) \geq \Psi_{k_2}(\{\mathbf{B}_1\}). \tag{5}$$

We imply from assertions in clause 1, lemma 11, corollaries 1.10 and 1.6*, that sequence $\mathbf{A}_2$ in G_t^{rl} can be considered instead of $\mathbf{A}_2$**.** From lemma 11, clauses 3b2, 3b3 we derive that:

$$\mathbf{A}_2' = \Gamma_t^{k_1}\left(C_{t-1}^{k-1} + 1\right), \dots, \Gamma_t^{k_1}\left(\mathrm{m} + C_{t-2}^{k_1-1}\right).$$

Since lengths of $\mathbf{A}_2'$ and $\mathbf{B}_2$ are equal and both sequences are in G_t^r (2b, lemma11), based on (1) we obtain:

$$\Psi_{k_2}(\{\mathbf{A}_2'\}) \geq \Psi_{k_2}(\{\mathbf{B}_2\}). \tag{6}$$

From (5) and (6), taking into account lemma 7, we obtain (2).

I. 1b3. Present sequences $\mathbf{A}^*$ and $\mathbf{B}^*$ in the following way:

$$\mathbf{A}^* = \mathbf{A}_1, \mathbf{A}_2; \quad \mathbf{B}^* = \mathbf{B}_1, \mathbf{B}_2,$$

where

$$\begin{gathered} \mathbf{A}_1 = \Gamma_t^{k_1}(1), \dots, \Gamma_t^{k_1}\left(m - C_{t-2}^{k_1-1}\right); \\ \mathbf{A}_2 = \Gamma_t^{k_1}\left(m - C_{t-2}^{k_1-1} + 1\right), \dots, \Gamma_t^{k_1}(m); \end{gathered} \tag{7}$$

$$\begin{gathered} \mathbf{B}_1 = \Gamma_t^{k_1}\left(m - C_{t-1}^{k_1-1} + 1\right), \dots, \Gamma_t^{k_1}\left(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1}\right); \\ \mathbf{B}_2 = \Gamma_t^{k_1}\left(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1} + 1\right), \dots, \Gamma_t^{k_1}\left(C_{t-1}^{k_1-1} + m\right). \end{gathered} \tag{8}$$

It can be observed that the length of $\mathbf{A}_1$ is equal to length $\mathbf{B}_2$, the length of $\mathbf{A}_2$ is equal to the length of $\mathbf{B}_1$.

From assertions 1, lemma 11 and corollaries 1.10, 1.6* follows that the sequence $\mathbf{B}_1'$ can be considered instead of sequence $\mathbf{B}_1$ all bundles of which are in the graph G_t^{lr}.

Proceeding from 3b2, 3b3, lemma 11 we obtain that:

$$\mathbf{B}_1' = \Gamma_t^{k_1}\left(C_{t-2}^{k_1-2} + 1\right), \dots, \Gamma_t^{k_1}\left(C_{t-1}^{k_1-1}\right).$$

Since $\mathbf{A}_2$ и $\mathbf{B}_1'$ are composed from bundles onle the left sub-graph G_t, then using corollaries 1. 10, 1. 6* and 1, lemma 11, we can consider $\mathbf{A}_2'$ and $\mathbf{B}_1''$ in the $\Delta_{t-1}^{(0)}$graph G_{t-1} instead of $\mathbf{A}_2$ and $\mathbf{B}_1'$ in G_t^lrespectively.

Having proved that $\mathbf{B}_1''$ enters the bundle $\Gamma_{t-1}^{k_1-1}\left(C_{t-1}^{k_1-1}\right)$, which is implied from 3aI леммы 11, on the base of inductive assumption we have:

$$\Psi_{k_{2-1}}(\{\mathbf{A}'_2\}) \geq \Psi_{k_{2-1}}(\{\mathbf{B}''_1\}),$$

or (9)

$$\Psi_{k_2}(\{\mathbf{A}_2\}) \geq \Psi_{k_2}(\{\mathbf{B}_1\}).$$

Prove that

$$\Psi_{k_2}(\{\mathbf{A}_1\}) \geq \Psi_{k_2}(\{\mathbf{B}_1\}). \tag{10}$$

Assume

$$\mathbf{A}'_1 = \Gamma_t^{k_1}(1 + C_{t-2}^{k_1-2}), \ldots, \Gamma_t^{k_1}(m - C_{t-2}^{k_1-1} + C_{t-2}^{k_1-2});$$

$$\mathbf{A}''_1 = \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \ldots, \Gamma_t^{k_1}(m + C_{t-2}^{k_1-2}).$$

Lengths $\mathbf{A}'_1$ and $\mathbf{A}''_1$ are equal to lengths $\mathbf{A}$, $\mathbf{A}'_1$ and $\mathbf{A}''_1$, are positioned in G_t^{lr} and G_t^{rl} (2.d, lemma 11), bundles from $\mathbf{A}'_1$ and $\mathbf{A}''_1$ meet each other under isomorphism G_t^{lr} and G_t^{rl} (3b2, 3b3, lemma 11).

On the base of already parsed case I. 1a we have:

$$\Psi_{k_2}(\{\mathbf{A}_1\}) \geq \Psi_{k_2}(\{\mathbf{A}'_1\}), \tag{11}$$

And from (I) we get :

$$\Psi_{k_2}(\{\mathbf{A}''_1\}) \geq \Psi_{k_2}(\{\mathbf{B}_2\}). \tag{12}$$

Since $\Psi_{k_2}(\{\mathbf{A}'_1\}) = \Psi_{k_2}(\{\mathbf{A}''_1\})$, we obtain (10) from (11) and (12). Taking into account lemma 7, from (9) and (10) we derive (2).

1.1b4. In this case we prove validity of (2) in the same way as it was done in case 1.1b3.

Thus we have parsed the case 1.1b.

I. Ic. We will divide sequences $\mathbf{A}^*$ and $\mathbf{A}$ to the following sub sequences:

$$\mathbf{A}^* = \mathbf{A}_1, \mathbf{A}_2; \qquad \mathbf{A} = \mathbf{B}_1, \mathbf{B}_2, \tag{13}$$

where

$$\mathbf{A}_1 = \Gamma_t^{k_1}(1), \ldots, \Gamma_t^{k_1}(r + m - 1 - C_{t-1}^{k_1-1}),$$

$$\mathbf{A}_2 = \Gamma_t^{k_1}(r + m - C_{t-1}^{k_1-1}), \ldots, \Gamma_t^{k_1}(m),$$

$$\mathbf{B}_1 = \Gamma_t^{k_1}(r), \ldots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1}),$$

$$\mathbf{B}_2 = \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \ldots, \Gamma_t^{k_1}(r + m - 1).$$

The lengths of $\mathbf{A}_1$ and $\mathbf{B}_2$, $\mathbf{A}_2$ and $\mathbf{B}_1$, as it can be easily seen, are respectively equal.

Since $\mathbf{A}_2$ and $\mathbf{B}_1$ are in G_t^l(2b, lemma 11), $\mathbf{B}_1$ contains the bundle $\Gamma_t^{k_1}(C_{t-1}^{k_1-1})$, with justification of 1, lemma 11, corollaries 1. 10, 1.6* and the second inductive assumption we obtain that:

$$\Psi_{k_2}(\{\mathbf{A}_2\}) \geq \Psi_{k_2}(\{\mathbf{B}_1\}). \tag{14}$$

Besides provided that examination of the sequences $\mathbf{A}_1$ and $\mathbf{B}_2$ concerns to already parsed case 1. 1b, we have:

$$\Psi_{k_2}(\{\mathbf{A}_1\}) \geq \Psi_{k_2}(\{\mathbf{B}_2\}). \tag{15}$$

From (14) and (15) by virtue of lemma 7 we obtain (2).

1. 2b. Partition the sequences $\mathbf{A}^*$ and $\mathbf{A}$ to the following sub-sequences:

$$\mathbf{A}^* = \mathbf{A}_1, \mathbf{A}_2, \qquad \mathbf{A} = \mathbf{B}_1, \mathbf{B}_2, \tag{16}$$

where

$\mathbf{A}_1 = \Gamma_t^{k_1}(1), \ldots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1}),$

$\mathbf{A}_2 = \Gamma_t^{k_1}(C_{t-1}^{k_1-1}+1), \ldots, \Gamma_t^{k_1}(m),$

$\mathbf{B}_1 = \Gamma_t^{k_1}(r), \ldots, \Gamma_t^{k_1}(r + C_{t-1}^{k_1-1} - 1),$

$\mathbf{B}_2 = \Gamma_t^{k_1}(r + C_{t-1}^{k_1-1}), \ldots, \Gamma_t^{k_1}(m + r - 1).$

The lengths of $\mathbf{A}_1$ and $\mathbf{B}_1$, $\mathbf{A}_2$ and $\mathbf{B}_2$ are respectively equal:

Arguing similar to the case 1.1b, for the sequences $\mathbf{A}_1$ and $\mathbf{B}_1$, while using (1) for sequences $\mathbf{A}_2$ and $\mathbf{B}_2$, we derive:

$$\Psi_{k_2}(\{\mathbf{A}_1\}) \geq \Psi_{k_2}(\{\mathbf{B}_1\}), \tag{17}$$
$$\Psi_{k_2}(\{\mathbf{A}_1\}) \geq \Psi_{k_2}(\{\mathbf{B}_2\}). \tag{18}$$

Application of (17), (18) and lemma 7 yields (2).

1. 2c. We write sequences $\mathbf{A}^*$ and $\mathbf{A}$ as follows:

$$\mathbf{A}^* = \mathbf{A}_1, \mathbf{A}_2, \qquad \mathbf{A} = \mathbf{B}_1, \mathbf{B}_2,$$

where

$$\begin{aligned} &\mathbf{A}_1 = \Gamma_t^{k_1}(1), \ldots, \Gamma_t^{k_1}(r-1); \\ &\mathbf{A}_2 = \Gamma_t^{k_1}(r), \ldots, \Gamma_t^{k_1}(m); \\ &\mathbf{B}_1 = \mathbf{A}_2; \\ &\mathbf{B}_2 = \Gamma_t^{k_1}(m+1), \ldots, \Gamma_t^{k_1}(r+m-1). \end{aligned} \tag{19}$$

Consideration of sequences $\mathbf{A}_1$ and $\mathbf{B}_2$ coincides the case 1.1b. Therefore

$$\Psi_{k_2}(\{\mathbf{A}_1\}) \geq \Psi_{k_2}(\{\mathbf{B}_2\}). \tag{20}$$

From (19), (20) and lemma 7 we obtain (2).

Thus, assuming that the lemma is true at $n = t - 1$, we verified the correctness of the first part of the inequality being proved at $n = t$.

With similar methods one can prove that the correctness of the second part of the inequality being proved $n = t$ follows from justification of lemma at $n = t - 1$.

The induction step has been accomplished.

Lemma 8* is proved.

It was demonstrated in corollary 2.5 how to calculate the value of function $\Psi_k(\{\mathbf{A}\})$ for any given sequence of Δ_n-bundles $\mathbf{A}$.

From the proved formula it is seen that in general the calculation is rather cumbersome. However in case $\mathbf{A}$ is an arbitrary $+$-compacted sequence of Δ_n-bundles the function $\Psi_k(\{\mathbf{A}\})$ takes relatively simple form.

L e m m a 12*. *For an arbitrary $+$-compacted set of sequences $\Delta_n^{k_1}$-наборов:* $\mathbf{A} = \Gamma_n^{k_1}(m_1), \Gamma_n^{k_1}(m_1+1), \ldots, \Gamma_n^{k_1}(m_2)$, *where* $0 \leq k_1 \leq k_2 \leq n$, $1 \leq m_1 \leq m_2 \leq C_n^{k_1}$, *the following equality holds:*

$$\Psi_{k_2}(\mathbf{A}) = \sum_{j=0}^{k_1-1} (C_{n-a_{j+1}}^{k_2-j} - C_{n-b_{j+1}}^{k_2-j}),$$

where

$a_j = \left(\Gamma_n^{k_1}(m_1 - 1)\right)_j$, $b_j = \left(\Gamma_n^{k_1}(m_2)\right)_j$, at $j = 1, \ldots, k_1$;

and

$$\left(\Gamma_n^{k_1}(0)\right)_j = \begin{cases} j & \text{at} \quad 1 \le j \le k_1 - 1, \\ k_1 - 1 & \text{at} \quad j = k_1. \end{cases}$$

Lemma 12* immediately follows from lemma 18, which is preceded by the following lemmas.

Лемма 13. *For an arbitrary $\Delta_n^{k_1}$-bundle A has at least one k_2-extension, with $0 \le k_1 \le k_2 \le n$, it is necessary and sufficient that the inequality meets:*

$$(A)_{k_1} \le k_1 + n - k_2.$$

Proof.

The necessity is evident when $k_1 = 0$.

The necessity in case $k_1 \ne 0$ is conditioned by lemma 2. Indeed, if B is a k_2-extension of A, then

$$(A)_{k_1} = (B)_{k_1} \le k_1 + n - k_2.$$

Sufficiency. If A has at least one k_2-extension then $C_{n-(A)_{k_1}}^{k_2-k_1} \ge 1$ in the ground of corollary 1.5, which is true when the inequality holds $0 \le k_2 - k_1 \le n - (A)_{k_1}$. The required condition follows from the last inequality since in virtue of condition of lemma $0 \le k_1 \le k_2$.

Lemma 14. *If $\mathbf{A} = A_1, \dots, A_m$ is an arbitrary $+$ -compacted sequence of $\Delta_n^{k_1}$-bundles, where $0 \le k_1 \le n$, and $\mathbf{B}_j = B_{j_1}, \dots, B_{jrj}$ is a k_2-extension A_j with $j = 1, \dots, m, k_1 \le k_2$ then the sequence $B_1, B_2, \dots, B_m = \mathbf{B}$ is $+$-compacted and there exists a k_2-extension* $\mathbf{A}$. *In addition if $A_1 = \Gamma_n^{k_1}(1)$, then* $\mathbf{B}$ *matches to starting segment of the sequence $\Gamma_n^{k_2}$.*

Proof.

If $\{\mathbf{B}\} = \emptyset$ or $k_2 = k_1$ the lemma is evident.

Assume $\{\mathbf{B}\} = \emptyset$ and $k_1 \ne k_2$.

It follows from corollary 1. 5, that at some $j \in \mathrm{I}_m$ the set $\{\mathbf{B}_j\}$ can be empty. Therefore it is sufficient to prove the lemma for the sequence $\mathbf{B}' = B_{j1}, \dots, B_{jt}$ produced from $\mathbf{B}$ after removing from $\mathbf{B}$ all sub-sequences$\mathbf{B}_j$ such that $\{\mathbf{B}_j\} = \emptyset,\ j = 1, \dots, m$.

For $p \in \mathrm{I}_{t-1}$, $\mathbf{B}_{j_p}$ and $\mathbf{B}_{j_{p+1}}$ are k_2-extensions of A_{j_p} and $A_{j_{p+1}}$respectively, besides $A_{j_p} < A_{j_{p+1}}$. Therefore

$$\forall B'B''(B' \in \{\mathbf{B}_{j_p}\} \,\&\, B'' \in \{\mathbf{B}_{j_{p+1}}\} \rightarrow B' \prec B''). \qquad (1)$$

Since each of sequences $\mathbf{B}_{j_p},\ p \in \mathrm{I}_t$ is $+$ -compacted (lemma 9), from (1) it follows that bundles in $\mathbf{B}'$ are positioned in lexicographical order. But $\mathbf{B}'$ coincides with the set of all extensions of bundles from $\mathbf{A}$, then $\mathbf{B}'$ is a k_2-extension of $\mathbf{A}$.

We will show that $\mathbf{B}'$ is a $+$ -compacted sequence. To prove that it is sufficient to demonstrate that the sequence $\mathbf{B}_{j_p}, \mathbf{B}_{j_{p+1}}$**is** $+$ -compacted for an arbitrary $p \in \mathrm{I}_{t-1}$.

The following cases are possible:

a. Bundles A_{j_p} and $A_{j_{p+1}}$ are adjacent in $\mathbf{A}$. Since $\mathbf{A}$ is a $+$ -compacted sequence of $\Delta_n^{k_1}$-bundle then

$$\neg\exists A\,(A \in R_n^{k_1} \,\&\, A_{j_p} \prec A \prec A_{j_{p+1}}). \qquad (2)$$

Let B' and B'' be respectively the first and last bundles $\mathbf{B}_{j_p}$ and $\mathbf{B}_{j_{p+1}}$.

Assume that $\mathbf{B}_{j_p}$, $\mathbf{B}_{j_{p+1}}$ are not $+$ -compacted. Then:

$$\exists B\,(B \in R_n^{k_2} \,\&\, B' \prec B \prec B'') \qquad (3)$$

and for $A = \{(B)_1, (B)_2, \dots, (B)_{k_1}\}$ the following relation is met:

$$(A_{j_p} \preceq A \preceq A_{j_{p+1}}) \,\&\, (A \in R_n^{k_1}). \qquad (4)$$

Provided B' and $B''$are respectively the last and the first bundles of $\mathbf{B}_{j_p}$ and $\mathbf{B}_{j_{p+1}}$, it is implied from (3) that $B \overline{\in} \{\mathbf{B}_{j_p}\}$ and $B \overline{\in} \{\mathbf{B}_{j_{p+1}}\}$, therefore B can be k_2-extension neither for A_{j_p} nor for $A_{j_{p+1}}$. Hence from (4) we obtain

$$(A_{j_p} \prec A \prec A_{j_{p+1}}) \,\&\, (A \in R_n^{k_1}), \qquad (5)$$

which contradicts to (2).

b. Bundles A_{j_p} and $A_{j_{p+1}}$ are not adjacent in $\mathbf{A}$, i.e. $\mathbf{A}$ is of type $A_1, A_2, \dots, A_{j_p}, A_{j_p+i}, \dots, A_{j_p+s}$, $A_{j_{p+1}}, \dots, A_m$, where for k_2-extension of bundles A_{j_p+i}, $i = 1, \dots, s$ the following equality holds: $\{\mathbf{B}_{j_p+i}\} = \emptyset$.

After repeating reasoning for case "a" we derive that:

$$\exists A\,(A \in R_n^{k_1} \,\&\, A_{j_p} \prec A \prec A_{j_{p+1}}) \qquad (6)$$

Thus A has k_2-extension B.

Since $\mathbf{A}$ is a + -compacted sequence then A coincides with one of bundles $A_{j_p+1}, \dots, A_{j_p+S}$. But neither of them has k_2-extensions thus proving that (6) is wrong.

We conclude that $\mathbf{B}' = \mathbf{B}$ is a + -compacted sequence. The last assertion we get arguing in the way as during proof similar to proof of comparable assertion of lemma 9.

Lemma 14 is proved.

L e m m a 15. *If* $\mathbf{B} = B_1, \dots, B_r$ k_2*-extension of an arbitrary* $\Delta_n^{k_1}$*-bundle* A, *где* $0 \le k_1 < k_2 \le n$, *то при* $r \neq 0$,

$$B_1 = A \cup \{(A)_{k_1} + 1, \dots, (A)_{k_1} + k_2 - k_1\},$$

$$B_r = A \cup \{k_1 + n - k_2 + 1, \dots, n\}.$$

The assertion of lemma 15 лis easily derived from lemma 6*.

Lemma 16. *If* $\mathbf{B} = B_1, \dots, B_r$ k_2 *is a prolongation of an arbitrary +-compacte sequence of* $\Delta_n^{k_1}$*-bundles* $\mathbf{A} = A_1, \dots, A_m$, *where* $0 \le k_1 < k_2 \le n$, *then at* $r \neq 0$, *a number* s *can be determined such that*

$$\forall x\,(0 \leqslant x \leqslant s \leqslant k_1 \to (A_m)_x \leqslant x + n - k_2 \,\&\, (A_m)_{s+1} > s + 1 + n - k_2),$$

Where $(A_m)_{k_1+1} = \infty$, *and*

$$B_r = \{(A_m)_1, \dots, (A_m)_s, s + 1 + n - k_2, s + 2 + n - k_2, \dots, n\},$$

P r o o f.

Let A_j be the first form the right bundle in the sequence $\mathbf{A}$, having at least one k_2-extension, $1 \le j \le m$.

Such a bundle A_j can be indicated provided $r \neq 0$.

Let $\mathbf{B}_j = B_{j_1}, \dots, B_{jv}$ be a k_2-extension of A_j.

Since $\mathbf{A}$ is a + -compacted sequence, applying lemma 14 produces that the last members of sequences $\mathbf{B}$ and $\mathbf{B}_j$ are the same, i.e. $B_r = B_{jv}$.

Therefore with consideration of lemma 15 we obtain that:

$$B_r = B_{jv} = A_j \cup \{k_1 + 1 + n - k_2, k_1 + 2 + n - k_2, \dots, n\}. \qquad (1)$$

Let us express components of bundle A_j through components of bundle A_m.

From lemma 13 we have that

$$(A_j)_{k_1} \leqslant k_1 + n - k_2, \tag{2}$$

$$(A_{j+t})_{k_1} > k_1 + n - k_2, \quad (t = 1, \ldots, m - j). \tag{3}$$

From (2) and (3) we obtain for bundles A_j and A_m that

$$\exists s_1 \forall x (0 \leqslant x \leqslant s_1 \leqslant k_1 \to (A_j)_x < x + n - k_2 \,\&\, (A_j)_{s_1+1} = $$
$$= s_1 + 1 + n - k_2), \tag{4}$$

$$\exists s_2 \forall x (0 \leqslant x \leqslant s_2 \leqslant k_1 \to (A_m)_x \leqslant x + n - k_2 \,\&\, (A_m)_{s_2+1} >$$
$$> s_2 + 1 + n - k_2). \tag{5}$$

In (3) and (4) we assume that $(A)_{|A|+i} = \infty$ and $(A_j)_e = -\infty$, where $A \in \{A_j, A_m\}$. Here if $s_1 = k_1$, then $A_j = A_m$ and only relation (4) is considered..

For sub-bundles $A_j,\ A_{j+1}, \ldots, A_m$ the following equalities are true as well:

$$(A_j)_x = (A_t)_x, \quad \text{with} \qquad 0 \leq x \leq s_1, \ \ t = j+1, \ldots, m. \tag{6}$$

Indeed the sequence $A_j,\ A_{j+1}, \ldots, A_m$ is +-compacted and thus can be sequentially produced with the algorithm $\Gamma_n^{k_2}$. Note also that proceeding from analysis of the procedure $\Gamma_n^{k_1}$, we conclude that first bundles for which condition (6) is met will be produced, then after all bundles of such type have been depleted, we obtain some bundle

$$\Gamma_n^{k_1}(p) = \{(A_j)_1, \ldots, (A_j)_{S_1-1}, (A_j)_{S_1} + 1, (A_j)_{S_1} + 2, \ldots, (A_j)_{S_1} + k_1 - s_1 + 1).$$

Bundle $\Gamma_n^{k_1}(p)$ cannot belong to the sequence $A_j, A_{j+1}, \ldots, A_m$ since as it follows from (4) $(\Gamma_n^{k_1}(p))_{k_1} \leq k_1 + n - k_2$, i.e. $\Gamma_n^{k_1}(p)$ has k_2-extensions, while all bundles $A_{j+1}, \ldots, A_m$ do not have k_2-extensions according the assumption.

It follows from (4), (5) and (6) that:

$$A_j = \{(A_m)_1, \ldots, (A_m)_{s_2}, \quad s_2 + 1 + n - k_2, \ldots, k_1 + n - k_2\}. \tag{7}$$

From (1) and (7) we derive that:

$$B_r = \{(A_m), \ldots, (A_m)_{s_2}, \ s_2 + 1 + n - k_2, \ s_2 + 2 + n - k_2, \ldots, \ n\},$$

Q.E.D.

Let's note that components of the first element of the sequence **B** can be expressed in terms of components of first term in the sequence **A**.

Let **B** be a k_2-extension of the + -compacted sequence

$$\mathbf{A} = \Gamma_n^{k_1}(m_1), \Gamma_n^{k_1}(m_1 + 1), \ldots, \Gamma_n^{k_1}(m'_2),$$

where

$$0 \leqslant k_1 \leqslant k_2 \leqslant n, \qquad 1 \leqslant m_1 \leqslant m'_2 \leqslant C_n^{k_1}.$$

Based on lemma 14 we conclude that the sequence **B** will also be + -compacted. Sequences **A** и **B** are uniquely defined with parameters соответственно k_1, m_1, m'_2 and k_2, k_1, m_1, m'_2 respectively. Thus the number of bundles in k_2-extension of some **A** in particular case **A** is +compacted sequence can be defined through a function $\theta(k_2, k_1, m_1, m'_1)$.

L e m m a 17. $\quad m = \theta(\boldsymbol{k}, k, 1, m) = C_n^k - \displaystyle\sum_{j=0}^{k-1} C_{n-a_{j+1}}^{k-j}$, *where* $0 \leq k \leq n$,

$1 \leq m \leq C_n^k, \quad a_j = (\Gamma_n^k(m))_j, \ \text{ with } \ j = 1, \ldots, k.$

Proof.

Consider the sequence $A = \Gamma_n^k(1), \dots, \Gamma_n^k(m)$. Since it is $+$ -compacted and starts from $\Gamma_n^k(1)$, it includes all Δ_n^k-bundles which do not exceed the bundle $\Gamma_n^k(m)$. under lexicographical ordering.

It follows from the algorithm Γ_n^k that the set $\{\mathbf{A}\}$ can be partitioned to subsets $Q_1, \dots, Q_k$ where Q_j, $j = 1, \dots, k$, is the set of all Δ_n^k-bundles at which the first $j-1$components coincide are equal to $a_1, a_2, \dots, a_{j-1}$ respectively, the j component does not exceed a_j and other components take all admissible values.

It can be easily seen that:

$$
\begin{aligned}
|Q_1| &= \sum_{x=1}^{a_1-1} C_{n-x}^{k-1},\\
|Q_2| &= \sum_{x=a_1+1}^{a_2-1} C_{n-x}^{k-2},\\
&\dots\dots\dots\dots \qquad (1)\\
|Q_{k-1}| &= \sum_{x=a_{k-1}+1}^{a_{k-1}-1} C_{n-x}^{1},\\
|Q_k| &= \sum_{x=a_{k-1}+1}^{a_k} C_{n-x}^{0}.
\end{aligned}
$$

Provided $C_n^k = C_{n-1}^{k-1} + C_{n-2}^{k-1} + \dots + C_{n-p-1}^{k-1} + C_{n-p-1}^{k}$, we derive from (1):

$$
\begin{aligned}
|Q_1| &= C_n^k - C_{n-a_1-1}^{k},\\
|Q_2| &= C_{n-a_1}^{k-1} - C_{n-a_2+1}^{k-1},\\
&\dots\dots\dots\dots\\
|Q_{k-1}| &= C_{n-a_{k-2}}^{2} - C_{n-a_{k-1}+1}^{2},\\
|Q_k| &= C_{n-a_{k-1}}^{1} - C_{n-a_k}^{1},
\end{aligned}
$$

From these equations we derive:

$$m = \sum_{j=1}^{k} |Q_j| + 1 = C_n^k - C_{n-a_1}^{k} - C_{n-a_2}^{k-1} - \dots - C_{n-a_{k-2}}^{3} - C_{n-a_{k-1}}^{2} -$$

$- C_{n-a_k}^{1} = C_n^k - \sum_{j=0}^{k_1-j} C_{n-a_{j+1}}^{k-j}$, Q.E.D.

Lemma 18.

$$\theta(k_2, k_1, l, m) = C_n^{k_2} - \sum_{n}^{k_1-1} C_{n-a_{j+1}}^{k_2-j} \theta(k_2, k_1, 1, m) = C_n^{k_2} - \sum_{j=0}^{k_1-1} C_{n-a_{j+1}}^{k_2-j}, \text{ where}$$

$0 \le k_1 \le k_2 \le n$, $1 \le m \le C_n^{k_1}$, $a_j = (\Gamma_n^{k_1}(m))_j$, $\quad j = 1, \dots, k$.

Consider the sequence $A_2 = \Gamma_n^{k_2}(1), \dots, \Gamma_n^{k_2}(\theta(k_2, k_1, 1, m))$. This sequence is $+$compacted and presents k_2-extension of the sequence $A = \Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)$ (refer to lemma 14 and definition of the function $\theta(x, y, z, u)$).

Lemma 16 provides that

$$\Gamma_n^{k_2}(\theta(k_2, k_1, 1, m)) = \{a_1, \dots, a_s, s+1+n-k_2, \dots, n\}, \qquad (1)$$

where s is obtained from the following condition:

$$\forall x\,(0 \leqslant x \leqslant s \leqslant k_1 \rightarrow a_x \leqslant x + n - k_2 \;\&\; a_{s+1} > s + 1 + n - k_2). \quad (2)$$

From lemma 17 and (1) we infer:

$$\theta(k_2, k_1, 1, m) = C_n^{k_2} - \sum_{j=0}^{s-1} C_{n-a_{j+1}}^{k_2-j} - \sum_{j=s}^{k_1-1} C_{k_2-j-1}^{k_2-j} = C_n^{k_2} - \sum_{j=0}^{s-1} C_{n-a_{j+1}}^{k_2-j}. \quad (3)$$

Meanwhile having given (2) we obtain:

$$C_n^{k_2} - \sum_{j=0}^{k_1-1} C_{n-a_{j+1}}^{k_2-j} = C_n^{k_2} - \sum_{j=0}^{s-1} C_{n-a_{j+1}}^{k_2-j} - \sum_{j=s}^{k_1-1} C_{n-a_{j+1}}^{k_2-j} =$$

$$= C_n^{k_2} - \sum_{j=0}^{s-1} C_{n-a_{j+1}}^{k_2-j}. \quad (4)$$

The required equality follows from (3) and (4).

Lemma 18 is proved.

§ 4. PROOF OF THE THEOREM ON SYSTEMS OF INCOMPATIBLE SETS WITH THE SMALLEST NUMBER OF SUB-SETS
(COMPLETION)

In the proof of the theorem we need some facts obtained in previous sections as well as a number of lemma the main from which is lemma 19*. ??а где ссылка.[5]в статье *

L e m m a 19*. *The number of the distinct Δ_n^l-sub-bundles produced from Δ_n^l-bundles of the set $\{L_n^l(1), \dots, L_n^l(m)\}$, where $0 \le i \le l \le n$, $1 \le m \le C_n^l$, is not greater than the number of all distinct Δ_n^l-sub-bundles produced from Δ_n^l-bundles of an arbitrary set consisting from m different Δ_n^l-bundles.*

We infer the next lemma directly from the definition of Δ_n-sub-bundles and Δ_n-super-bundles.

L e m m a 20. *For arbitrary Δ_n-bundles A and B, if A is a (proper) $\Delta_n^{|A|}$-sub-bundle B, then $I_n \backslash A$ is a (proper) $\Delta_n^{n-|A|}$-super-bundle $I_n \backslash B$, and, the contrary, if A is a (proper) $\Delta_n^{|A|}$-super-bundle B, then $I_n \backslash A$ is a (proper) $\Delta_n^{n-|A|}$-sub-bundle $I_n \backslash B$.*

C o r o l l a r y 1. 20. *If $S = \{A_1, \dots, A_m\}$ is an arbitrary set of Δ_n-bundles and $S' = \{I_n \backslash A_1, \dots, I_n \backslash A_m\}$, then for any k, $0 \le k \le n$, the number of all distinct Δ_n^k-sub-bundles of bundles from S is equal to the number of all distinct Δ_n^{n-k}-super-bundles of bundles from S'.*

It follows from the corollary 1.20 that to prove lemma 19* it is sufficient to prove the following lemma.

L e m m a 19. (on minimal number of super-bundles). *The number of distinct $\Delta_n^{k_2}$-super-bundles produced from $\Delta_n^{k_1}$-bundles of the set $\{\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)\}$, where $0 \le k_1 \le k \le n$, $1 \le m \le C_n^{k_1}$, is not greater the number of all distinct $\Delta_n^{k_2}$-super-bundles produced from $\Delta_n^{k_1}$-bundles of an arbitrary set consisting from m different $\Delta_n^{k_1}$-bundles.*

We proceed to the proof of lemma 19. First we will prove a number of assertions concerning relations between aggregations of Δ_n-bundles resulting from the feature of some bundles to be super-bundles of others.

Let's introduce the following notations.

Assume S be an arbitrary set of $\Delta_{n,p}$-bundles, we denote a lexicographically ordered sequence composed from all bundles of S as S^0. For an arbitrary set of Δ_n-bundles of -S we denote the set of distinct Δ_n-supers-bundles of bundles the length k from S, where $0 \le k \le n$, through $V(k, S)$, while $N_k^o(S)$ is the number $|V(k, S)|$.

---

[5] Lemma 19*substantially coincides with the Macaulay theorem [1]. It was obtained by Kraskal (I950) and Katona as well (I960). The proof of the lemma given here was independently derived by the author [10] and is substantially different from [2].

L e m m a 21. *If the sequence* $B_1, \dots, B_r$ *is a* k_2*-extension of* $\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)$, *where* $0 \leq k_1 \leq k_2 \leq n$, *then:*

$$\forall jB \left(1 \leqslant j \leqslant m \,\&\, B \in R_n^{k_1} \setminus \bigcup_{j=1}^{r} \{B_1, \dots, B_r\} \rightarrow \Gamma_n^{k_1}(j) \sqsubset B\right).$$

P r o o f.

Let B be an arbitrary element from $R_n^{k_1} \setminus \bigcup_{j=1}^{r}\{B_1, \dots, B_r\}$. ?

It is clear that the bundle B is a k_2-extension of $\Delta_n^{k_1}$-bundle.

$$A = \{(B)_1, \dots, (B)_{k_1}\}.$$

As all k_2-extensions of each bundle $\Gamma_n^{k_1}(j)$, $j = 1, \dots, m$, enter the set $\bigcup_{j=1}^{r}\{B_j\}$? and $B \neq B_j$, $j = 1, \dots, m$, then

$$A \in R_n^{k_1} \setminus \bigcup_{j=1}^{m} \{\Gamma_n^{k_1}(j)\}. \qquad (1)$$

It follows from (1) that

$$\Gamma_n^{k_1}(j) \prec A \quad j = 1, \dots, m, \tag{2}$$

or

$$\forall j \exists q_j \forall x (1 \leqslant j \leqslant m \,\&\, 0 \leqslant x \leqslant q_j < k_1 \rightarrow (A)_x =$$
$$= (\Gamma_n^{k_1}(j))_x \,\&\, (A)_{q_j+1} > (\Gamma_n^{k_1}(j))_{q_j+1}). \qquad (3)$$

Since

$$(A)_x = (B)_x \quad x = 1, \dots, q_j. \tag{4}$$

and

$$\forall jy (1 \leqslant j \leqslant m \,\&\, q_j + 1 < y \leqslant k_2 \rightarrow (B)_y > (B)_{q_j+1}). \qquad (5)$$

then from (3), (4), (5) we obtain that $(\Gamma_n^{k_1}(j))_{q_j+1} \overline{\in}+ B$, at $j = 1, \dots, m$, it is implied from the latter that $\Gamma_n^{k_1}(j)\urcorner \in B$, ? which in turn and etc.

C o r o l l a r y 1.21. *The set of all* $\Delta_n^{k_2}$ *-super-bundles of bundles of the sequence* $\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)$, *where* $0 \leq k_1 \leq k_2 \leq n$, $1 \leq m \leq C_n^{k_1}$, *coincides with the set of all bundles of* k_2*-prolongation* $\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)$, *which in turn implies that:*

$$N_{k_2}^{0}(\{\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)\}) = \psi_{k_2}(\{\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)\}).$$

C o r o l l a r y 2.21. k_2*-extension of an arbitrary bundle* $\Gamma_n^{k_1}(m)$, *where* $0 \leq k_1 \leq k_2 \leq n$, $1 \leq m \leq C_n^{k_1}$, *coincides with the set of all* $\Delta_n^{k_2}$*-super-bundles* $\Gamma_n^{k_1}(m)$ *such that no one of them is a* $\Delta_n^{k_2}$*-super-bundles for any of bundles* $\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m-1)$, *which in particular implies that:*

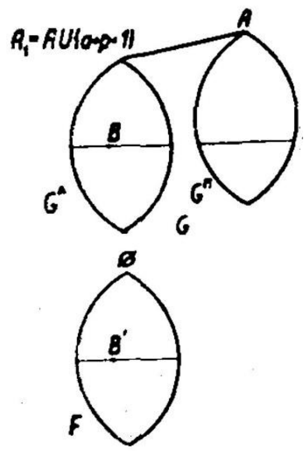
$$N_{k_2}^{0}(\{\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)\}) = N_{k_2}^{0}(\{\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m-1)\} + \psi_{k_2}(\{\Gamma_n^{k_1}(m)\}).$$

L e m m a 22. *For an arbitrary* $\delta\Delta$*-graph* G, *if* G *is some* $\delta\Delta_n^{a,p}$*-graph,* G^l *and* G^r *– the left and right sub-graphs of the graph* G *respectively,* B *is an arbitrary left vertex of* G, *then*

$$\nu\{(B), G^l, G^r\} = \{B \setminus \{a + p+\}\}.$$

P r o o f.

Let A be the root of G and $\Delta_{n-a-p-1}^{(0)}$-graph (pic. 10).



Pic. 10.

According the definition, G^l is a Δ_n- graph with the root $A \cup \{a+p+1\}$ while $G^r$is a $\delta\Delta_n^{a,p+1}$-graph with the root A.

It follows from lemma 6* that G^l and G^r are isomorphic the graph F. Therefore if

$$\{B\}' = \nu(\{B\},\ G^l, F),$$

then

$\{B''\} = \nu(\{B'\}, F, G^r) = \nu(\{B\},\ G^l, G^r)$.

Since B is a vertex from $G^l$and G^l is a Δ-графом, then, on the ground of definition of Δ-graphs, B is an extension of the root G, i.e. the vertex $A_1 = A \cup \{a+p+1\}$. Hence B with justification of lemma 5 can be presented as $A_1 \cup C$, where C is some super-bundle of the bundles $\{a+p+2, \dots, n\}$.

By virtue of lemma 6* point 2, we obtain:

1. If $B = A_1$, then $B' = \emptyset$ and $B'' = A = A_1 \setminus \{a+p+1\} = B \setminus \{a+p+1\}$;

2. If $B \neq A_1$, then $B' = \{(B)_{|A_1|+1} - a - p - 1, \dots, (B)_{|B|} - a - p\} = \{(C)_1 - a - p - 1, \dots, (C)_{|C|} - a - p - 1\}$ and

$B'' = A \cup \{(C)_1, \dots, (C)_{|C|}\} = (A_1 \setminus \{a+p+1\}) \cup C = B \setminus \{a+p+1\}$.

Lemma 22 is proved.

Corollary 1.22. *If S is an arbitrary set of bundles and G is some $\delta\Delta$-graph, then for every natural number k,*

$$V^r\ (k, S) \neq \emptyset \to V^l(k, S) \neq \emptyset,$$

where $V^l\ (k, S)$ and $V^r\ (k, S)$ are equal to sets of supebundles of bundles from S, belonging respectively to the left and right sub-graphs G.

Corollary 2.22. *For an arbitrary $\delta\Delta$-graph G, if G is some $\delta\Delta_n^{a,p}$-graph, G^l and G^r are the left and right sub-graphs G respectively, B is an arbitrary right vertex of G, then $\nu(\{B\}, G^r, G^l)$ is a $\Delta_n^{|B|+1}$-super-bundle B, containing the number $\{a+p+1\}$.*

It's worth to note that lemma 22 allows to formulate a simple algorithm of synthesis an arbitrary Δ-sub-graph. Actually assume we need to produce a $\Delta_n^{a,p}$-graph with the root A.

We implement the following productions.

1. Produce a Δ_n-graph G_{n-a-p} with the root $A \cup \{a+p+1,\ a+p+2, \dots, n\}$.

It is clear that G_{n-a-p} coincides with its root.

2. Let Δ_n be a graph G_i with the root $A \cup \{a+p+1,\ \dots, a+p+i\}$, where $1 \leq i \leq n-a-p$, has been already produced. Then Δ_n -graph G_{i-1} with the root c $A \cup \{a+p+1,\ \dots, a+p+i-1\}$ is the graph the left sub-graph of which coincides with G_i, while the right one is produced from G_i by substituting each vertex B of the graph G_i with the left vertex $C = B \setminus \{a+p+i\}$.

The graph G_0, obtained in this way will be the sought $\delta\Delta_n^{a,p}$-graph with the root A.

L e m m a 23. *If Если* G_r is a $\Delta_r^{(0)}$*-graph, where* $r \in \{n-1,\ n\}$, B *and* C *are arbitrary vertices of* G_n^l *and* G_{n-1}, B_1 *and* C_1 *are arbitrary super-bundles of* B *and* C, *respectively, then:*

1. $B_1 \in G_n^l, \quad C_1 \in G_{n-1}$;

2. *Bundle* $B_1' = \nu(\{\mathrm{B}_1\}, G_n^l, G_{n-1}$ *will be a* $\Delta_{n-1}^{|B_1|-1}$*-super-bundle of the bundle* $B' = \nu(\{B\}, G_n^l, G_{n-1})$, *and the bundle* $C_1' = (\{C_1\}, G_{n-1}, G_n^l) - \Delta_n^{|C_1|+1}$*-super-bundle of the bundle* $C' = \nu(\{B\}, G_{n-1}, G_n^l)$.

P r o o f.

Assume $|B| = k_1,\ |B_1| = k_2,\ |C| = l_1,\ |C_1| = l_2$.

1. Since $B \in G_n^l$, from 2a, lemma 19 we conclude that $B = \Gamma_n^{k_1}(t)$, where $1 \le t \le C_{n-1}^{k_1-1}$.

From corollary 1.21 we derive that B_1 is a k_2-extension of one of the bundles $\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(t)$. Since each of these bundles enters $\{G_n^l\}$ and G_n^l is a Δ_n^1-graph, then it is implied from lemma 4 that B_1 also enters $\{G_n^l\}$.

One can get convinced in the correctness of the statement $C_1 \in \{G_{n-1}\}$ in similar way.

2. We will prove the first part of the statement of lemma 23, namely the fact that B_1' is $\Delta_{n-1}^{k_2-1}$-super-bundles of B'.

The second part of the statement is proved by analogy.

Provided G_n and G_{n-1} are Δ-graphs we derive that $n \ge 1$, which in turn implies that $B \ne \emptyset$.

Since $B,\ B_1 \in \{G_n^l\}$ and G_n^l is a Δ_n-graph with the root $A = \{1\}$ then B and B_1 are extensions of A. Therefore $B = A \cup \tilde{B}$, $B_1 = A \cup \tilde{B}_1$ where $\tilde{B}$ and $\tilde{B}_1$ are sub-bundles of the bundle $\{2, \dots, n\}$ with justification of lemma 5.

As B_1 is a super-bundle of B, it is clear that B is a sub-bundle of B_1, consequently, $\tilde{B}$ is a sub-bundle of B_1.

From 2, lemma 6*, we obtain that at $B \ne \emptyset$, we have:

$$B' = \{(\tilde{B})_1 - 1, \dots, (\tilde{B})_{|B|} - 1\}, \tag{1}$$

$$B_1' = \{(\tilde{B}_1)_1 - 1, \dots, (\tilde{B}_1)_{|B_1|} - 1\}. \tag{2}$$

As $\tilde{B}$ is a sub-bundle of $\tilde{B}_1$, then on the ground of (1) and (2) we conclude that B' is a sub-bundle of B_1', hence B_1' is a super-bundle of B'.

Lemma 23 is proved.

L e m m a 24. *If* G_r *is a* Δ_r^0*-graph, where* $r \in \{n-1,\ n\}$, B *and* C *are arbitrary vertices of* G_n^r *and* G_{n-1}, B_1 *is an arbitrary super-bundle of* B, *excluding numbers* 1, C_1 *– arbitrary super-bundle* C, *then:*

1. $B_1 \in \{G_n^r\}, \quad C_1 \in \{G_{n-1}\}$;
2. *The bundle* $B_1' = \nu(\{B_1\},\ G_n^r,\ G_{n-1})$ *will be a* $\Delta_{n-1}^{|B_1'|}$*-super-bundle* $B' = \nu$ *and the bundle* $C_1' = \nu(\{C_1\},\ G_{n-1},\ G_n^r)$ *will be* $\Delta_{n-1}^{|C_1'|}$*-super-bundle of the bundle* $C' = \nu(\{G\},\ G_{n-1},\ G_n^r)$.

P r o o f.

15.As B_1 does not contain the number 1, on the ground of lemma 22 we have that $B_1 \in \{\ G_n^r\}$.

Assertion: $C_1 \in \{G_{n-1}\}$. The case is repetition of 1, lemma 23.

16.Prove that B_1' is a $\Delta_{n-1}^{|B_1'|}$-super-bundle B'.

Actually if $B = \emptyset$, then the assertion being proved obviously follows from 2, lemma 6*.

Assume $B = \emptyset$. As B_1 is a super-bundle of B, by virtue of the definition B is a sub-bundle B_1.

From 2, lemma 6* we have that under $B \ne \emptyset$, the following is met:

$$B' = \{(B)_1 - 1, \dots, (B)_{|B|} - 1\}, \tag{1}$$

$$B_1' = \{(B_1)_1 - 1, \dots, (B_1)_{|B_1|} - 1\}. \tag{2}$$

Since B is a sub-bundle of B_1, then on the ground of (1) and (2) it follows that B' will be a sub-bundle of B_1' and, consequently, B' is super-bundle of **B**.

The second part of 2, lemma 24 is proved by analogy.

Lemma 24 is proved.

Let's introduce the following notations.

If $G_n$is a $\Delta_n^{(0)}$-graph and S – an arbitrary subset $\{G_n\}$, then $V_n^p(k,S)$ is a denotation for the set of all Δ_n^k-super-bundles of bundles from S entering to $\{G_n^p\}$, where k is an arbitrary natural number $p \in (\emptyset, l, r, ll, lr, rl, rr, G_n^\emptyset = G_n, V_n^\emptyset(k,S) = V_n(k,S)$.

L e m m a 25. *If* $G_r$is a $\Delta_t^{(0)}$*-graph, where* $r \in \{n-2,\ n-1,\ n\}, S$ is an arbitrary subset of bundles of equal length then*:*

1. *If* $S \subseteq G_n^l$ *and* $S_1 = \nu(S,\ G_n^l,\ G_{n-1})$, *then:*
   a. $V_n(k,S) = V_n^l(k,S)$;
   b. $|V_n(k,S)| = |V_{n-1}(k-1,\ S_1)|$;
2. *If* $S \subseteq G_n^r$ *and* $S_2 = \nu(S,\ G_n^r,\ G_{n-1})$, *then:*
   a. $V_n(k,S) = V_n^l(k,S) \cup V_n^r(k,S)$;
   b. $V_n^l(k,S) = V_n\big(k,\ \nu(S,\ G_n^r, G_n^l)\big)$;
   c. $|V_n^r(k,S)| = |V_{n-1}(k,\ S_2)|$;
   d. $\nu(\nu(S,\ G_n^r, G_n^l), G_n^l, G_{n-1}) = S_2$;
3. *If* $S \subseteq \{G_n^{ll}\}$ *and* $S_2 = \nu(S,\ G_n^{ll},\ G_{n-1})$, *then*:
   a. $V_n(k,S) = V_n^{ll}(k,S)$;
   b. $|V_n(k,S)| = |V_{n-2}(k-2,\ S_2)|$;
4. *If* $S \subseteq \{G_n^{lr}\}$, $S_4 = \nu(S,\ G_n^{lr},\ G_{n-2})$, $P = \nu(S,\ G_n^{lr}, G_n^{rl})$, $P_1 = \nu(P,\ G_n^{rl},\ G_{n-2})$, *then:*
   a. $V_n(k,S) = V_n^{ll}(k,S) \cup V_n^{lr}(k,S)$;
   b. $V_n^{ll}(k,S) = V_n\big(k,\ \nu(S,\ G_n^{lr}, G_n^{ll})\big)$;
   c. $\big|V_n^{lr}(k,S)\big| = |V_{n-2}(k-1,\ S_4)|$;
   d. $V_n(k,P) = V_n^{ll}(k,P) \cup V_n^{rl}(k,P)$;
   e. $V_n^{ll}(k,P) = V_n\big(k,\ \nu(P,\ G_n^r, G_n^l)\big)$;
   f. $|V_n^{rl}(k,P)| = |V_{n-2}(k-1,\ P_1)|$;
   g. $\nu(P,\ G_n^r, G_n^l = \nu(S,\ G_n^{lr}, G_n^{ll})$;
   h. $P_1 = S_4$;
5. *If* $S \subseteq \{G_n^{rr}\}$ *and* $S_5 = \nu(S,\ G_n^{rr},\ G_{n-2})$, *then:*
   a. $V_n(k,S) = V_n^l(k,S) \cup V_n^{rl}(k,S) \cup V_n^{rr}(k,S)$;
   b. $V_n^l(k,S) = V_n\big(k,\ \nu(S,\ G_n^r, G_n^l)\big)$;
   c. $V_n^{rl}(k,S) = V_n^{rl}\big(k,\ \nu(S,\ G_n^{rr}, G_n^{rl})\big)$;
   d. $|V_n^{rr}(k,S)| = |V_{n-2}(k,\ S_5)|$.

P r o o f.

1a, 1b. The assertions la and 1b of the lemma directly follow from lemma 23.

2a.The assertion 2a follows from the corollary 1.22 and the fact that all Δ_n-bundles belong to $\{G_n\}$.

2b.As it follows from lemma 22, each bundle from G_n^l contains 1 and none of bundles from G_n^r contains the number 1.

At the same time each super-bundle of an arbitrary bundle from S, containing 1 will obviously be a super-bundle for one from $\Delta_n^{k_1+1}$-super-bundles from bundles of S, containing 1, where $k_1$is the length of bundles from S.

Therefore with justification of the corollary 2.22 we obtain:

$$V_n^l(k,S) = V_n^l(k,\ \nu(S, G_n^r, G_n^l)). \tag{1}$$

The required assertion follows from point 1a, lemma 25 and (1).

2c. Since no super-bundle of bundles from S, belonging $V_n^r(k,S)$, contains the number 1 the required assertion follows from statement 2 lemma 24.

2d. The assertion immediately follows from isomorphism among G_n^l, G_n^r and G_{n-1} (lemma 11, item 1).

3a. 3b. Let $S_1 = \nu\big(S, G_n^l, G_{n-1}\big)$.

Since $S \subseteq \{G_n^{ll}\}$ then $S_1 \subseteq \{G_{n-1}^l\}$. It immediately follows from the fact that graph G_n^{ll} corresponds to the graph G_{n-1}^{ll} under isomorphism G_n^l and G_{n-1}(lemma11, 1). From isomorphism G_n^{ll}, G_{n-1}^l and G_{n-2} (lemma 11, point 1) it follows that $\nu(S_1, G_{n-1}^l, G_{n-2} = \nu\big(S, G_n^{ll}, G_{n-2}\big) = S_2$.

From item 1 of the lemma being proved we have:

$$V_{n-1}(l,\ S_1) = V_{n-1}^l(l,\ S_1), \tag{1}$$

and also

$$|V_{n-1}(l,\ S_1)| = |V_{n-2}(l-1,\ S_2)|, \tag{2}$$

$$|V_n(k,S)| = |V_{n-1}(k-1,\ S_1)|. \tag{3}$$

Assuming $l = k-1$ from (2) and (3), we infer:

$$|V_n(k,S)| = |V_{n-2}(k-2,\ S_2)|,$$

which is the assertion 3b of lemma 25.

Whereas G_n^{ll} and G_{n-1}^l correspond each other under isomorphism G_n and G_{n-1}, from 2, lemma 23 and (1) it follows that

$$V_n(k,S) = V_n^{ll}(k,S),$$

which is assertion 3a of lemma 25.

4a. $S \subseteq \{G_n^l\}$ since $S \subseteq \{G_n^{lr}\}$, and thus on the ground of 1a lemma 25 we have:

$$V_n(k,S) = V_n^l(k,S). \tag{1}$$

Applying the corollary 1.22 to the graph G_n^l, we get:

$$V_n^l(k,S) = V_n^{ll}(k,S) \cup V_n^{lr}(k,S). \tag{2}$$

The statement 4a follows from (1) and (2). .

4b. It follows from lemma 22, each bundle from G_n^{ll} contains the number 2 while no bundle from G_n^{lr} contains it.

Arguing by analogy with the case 2b, we get that:

$$V_n^{ll}(k,S) = V_n^{ll}(k,\ \nu(S, G_n^{lr}, G_n^{ll})). \tag{1}$$

The statement 4b follows from (1) and the point 3a, lemma 25.

4c. The proof of the statement 4c of lemma 25 is similar to the proof of 3b, lemma 25.

4d; 4e; 4f. On the ground of 2a of lemma 25 we have:

$$V_n(k,P) = V_n^l(k,P) \cup V_n^r(k,P). \tag{1}$$

$S \subseteq \{G_n^r\}$ since $S \subseteq \{G_n^{rl}\}$, then from. 2b of lemma 25 we obtain:

$$V_n^l(k,P) = V_n(k,\ \nu(P, G_n^r, G_n^l)), \tag{2}$$

While according we conclude:

$$V_n(k,\ \nu(P, G_n^r, G_n^l)) = V_n^l(k,\ \nu(P, G_n^r, G_n^l)). \tag{2'}$$

All bundles from $\nu(P, G_n^r, G_n^l)$ belong to $\{G_n^{ll}\}$, the assertion follows from 4a and 2b of lemma 11. Therefore from 3, lemma 25, we obtain:

$$V_n(k,\ \nu(P, G_n^r, G_n^l)) = V_n^{ll}(k,\ \nu(P, G_n^r, G_n^l)). \tag{3}$$

It follows from (2) and (2′) that to examine super-bundles of bundles from P, entering G_n^l, we can consider the set $\nu(P, G_n^r, G_n^l)$ instead of P.

Thus

$$V_n^{ll}(k,P) = V_n^{ll}(k,\ \nu(P, G_n^r, G_n^l)). \tag{4}$$

From (2), (3) and (4) we infer that:

$$V_n^l(k,P) = V_n^{ll}(k,P). \tag{5}$$

$$V_n^{ll}(k,P) = V_n(k,\ \nu(P, G_n^r, G_n^l)). \tag{6}$$

Let $P_2 = \nu(P, G_n^r, G_{n-1})$. Then on the ground of 2b, lemma 25, we have:

$$|V_n^r(k,P)| = |V_n(k,P_2)|. \tag{7}$$

With justification of point 2a of lemma 11 we see that all bundles from P_1 are vertices of G_{n-1}. By virtue of point 2, lemma 25 we can write

$$V_{n-1}(k,P_2) = V_{n-1}^l(k,P_2), \tag{8}$$

$$|V_{n-1}(k,P_2)| = |V_{n-2}(k-1,P_1)|. \tag{9}$$

Here we used the fact that:

$$\nu\big(P_2, G_{n-1}^l, G_{n-2}\big) = P_1.$$

As graphs G_n^{rl} and G_{n+1}^l correspond each other under isomorphism of graphs G_n^r and G_{n-1} (lemma 11, point. 1), then from (8) and point 2 of lemma 24 we conclude that:

$$V_n^r(k,P) = V_n^{rl}(k,P). \tag{10}$$

From (7), (9) and (10) the point 4d of lemma 25 follows, from (1), (5) and (10) –point 4f of lemma 25, while (6) is the assertion 4d of lemma 25.

4g, 4i. Let B be arbitrary bundles from S, $|B| = k_1$,

$$\nu\big(\{B\}, G_n^{lr}, G_n^{rl}\big) = B_1.$$

As $S \subseteq \{G_n^{lr}\}$, then $B \in \{G_n^{lr}\}$ and by virtue of point 2d of lemma 11, it can be represented in the following form: $B = \Gamma_n^{k_1}(t)$, where:

$$G_{n-2}^{k_1-2} + 1 \le t \le C_{n-1}^{k_1-1}.$$

We obtain the assertion of the point 4i of the lemma 11 as an immediate consequence of isomorphism of graphs G_n^{lr}, G_n^{rl}, G_{n-2} and the definition of set P, then with recalling the 3b2 and 3b3 of lemma 11 we derive:

$$B_1 = \nu(\{B\}, G_n^{lr}, G_n^{rl}) = \Gamma_n^{k_1}\big(t + C_{n-2}^{k_1-1}\big). \tag{1}$$

Then from (1), points 4b and 4a of lemma 14 it is implied that:

$$\nu\big(\{B\}, G_n^{lr}, G_n^{rl}\big) = \big\{\Gamma_n^{k_1+1}\big(t - C_{n-2}^{k_1-2}\big)\big\} = \nu\big(\{B_1\}, G_n^r, G_n^l\big).$$

Considering the arbitrariness of choosing the element B and the definition of the set P, proves the assertion 4g of lemma 25.

5a. As $S \subseteq \{G_n^{rr}\}$, then $S \subseteq \{G_n^r\}$, therefore proceeding from 2a lemma 25 we have:

$$V_n\big(k,\ S\big) = V_n^l(k,S) \cup V_n^r\big(k,S\big),$$

while taking onto account corollary 1.22 we infer:

$$V_n\big(k,\ S\big) = V_n^l(k,S) \cup V_n^{rl}\big(k,S\big) \cup V_n^{rr}(k,S).$$

5b. The assertion 5b of lemma 25 is a particular case of 2b lemma 25.

5c; 5d. Let $S_2 = \nu(S, G_n^r, G_{n-1})$. As $S \subseteq \{G_n^{rr}\}$,.by virtue of 1 lemma 11 we infer that $S_2 \subseteq \{G_{n-1}^r\}$, hence on the ground of 2b of lemma 25 for the graph $G_{n-1}$and the equality

$$S_5 = \nu(S, G_n^{rr}, G_{n-2}) = \nu(S_2, G_{n-1}^r, G_{n-2}).$$

Thus we obtain:

$$V_{n-1}^l(k,S_2) = V_{n-1}\Big(k, \nu\big(S_2, G_{n-1}^r, G_{n-1}^l\big)\Big), \tag{1}$$

$$|V_{n-1}^r(k,S_2)| = |V_{n-2}(k,S_5)|. \tag{2}$$

As having defined isomorphism between G_n^r and G_{n-1} we get that graphs G_n^{rl} and G_n^{rr} correspond to graphs G_{n-1}^l and G_{n-1}^r, on the ground of 2 lemma 24, (1) and (2) we get that:

$$V_n^{rl}(k,S) = V_n^{rl}(k, \nu\big(S, G_n^{rr}, G_n^{rl}\big)),$$

$$|V_{n-1}^{rr}(k,S)| = |V_{n-2}(k,S_5)|,$$

which is just what we need to prove.

Lemma 25 is proved.

L e m m a 26. *If S is an arbitrary set of the right vertices of the k_1-tier of Δ_n^0-graph G_n, $T^* = \{\Gamma_n^{k_1}(C_{n-1}^{k_1-1}+1), \dots, \Gamma_n^{k_1}(C_{n-1}^{k_1-1}+m)\}$, where $0 \le k_1 \le n$, $m = |S|$, besides for the vertices of $\Delta_{n-1}^{(0)}$-graph G_{n-1} lemma 19 is true, then:*

$$N_{k_2}^0(T^*) \le N_{k_2}^0(S), \tag{1}$$

где $k_1 \le k_2 \le n$.

P r o o f.

From 2b, lemma 11 it follows that$T^* \subseteq \{G_n^r\}$. Therefore applying 2a of lemma 25 to sets S and T^*, we have proved that

$$N_{k_2}^0(T^*) = |V_n(k_2, T^*)| = |V_n^l(k_2, T^*)| + |V_n^r(k_2, T^*)|, \tag{2}$$

$$N_{k_2}^0(S) = |V_n(k_2, S)| = |V_n^l(k_2, S)| + |V_n^r(k_2, S)|. \tag{3}$$

Let $S' = \nu(S, G_n^r, G_{n-1})$ and $T' = \nu(T^*, G_n^r, G_{n-1})$.

From 2b lemma 25 we obtain:

$$V_n^l(k, S) = V_n(k, \nu(S, G_n^r, G_n^l)), \tag{4}$$

then from(4) and the point 1b, lemma 25 we derive:

$$|V_n^l(k, S)| = \left|V_{n-1}\left(k-1, \nu\left(\nu(S, G_n^r, G_n^l)\right), G_n^l, G_{n-1}\right)\right|. \tag{5}$$

Finally, from (5) and 2d, lemma 25, we obtain:

$$|V_n^l(k, S)| = |V_{n-1}(k-1, S')|. \tag{6}$$

According point 2c of lemma 25 the following equality takes place:

$$|V_n^r(k, S)| = |V_{n-1}(k, S')|. \tag{7}$$

Similarly, for the set T^* we get that:

$$|V_n^l(k, T^*)| = |V_{n-1}(k-1, T')|, \tag{8}$$

$$|V_n^r(k, T^*)| = |V_{n-1}(k, T')|. \tag{9}$$

As $\tilde{T}^*$is a +compacted and $\Gamma_n^{k_1}(C_{n-1}^{k_1-1}+1) \in T^*$, applying 5, lemma 6* and 3a2 of lemma 11 we infer that $\tilde{T}'$ is a + compacted sequence of Δ_{n-1}^k-bundles and $\Gamma_{n-1}^k(1) \in T'$.

It is clear also that $|T'|= |S'|$. Therefore on the ground of our assumption on correctness of lemma 19 for Δ_{n-1}^0-graphs we obtain that:

$$|V_{n-1}(k, T')| \le |V_{n-1}(k, S')|. \tag{10}$$

Assuming (2), (3), (6), (8), (9) and (10) proves (1).

Lemma 26 is proved.

L e m m a 27. *If P and R are arbitrary sets of the left and right vertices correspondingly of the k_1-tier of Δ_n^0-graph G_n,*
$P^* = \{\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m_1)\}$, $\quad R^* = \{\Gamma_n^{k_1}(C_{n-1}^{k_1-1}+1), \dots, \Gamma_n^{k_1}(C_{n-1}^{k_1-1}+m_2)\}$,
where $0 \le k \le n$, $m_1 = |P|$, $m_2 = |R|$, and for vertices of $\Delta_{n-1}^{(0)}$-graph G_{n-1} lemma19 is true, then

$$N_{k_2}^0(P^* \cup R^*) \le N_{k_2}^0(P \cup R), \tag{11}$$

where $k_1 \le k_2 \le n$.

P r o o f.

Introduce notations.

Assume

$$P_1 = \nu\big(P, G_n^l, G_{n-1}\big), \quad P_1^* = \nu\big(P^*, G_n^l, G_{n-1}\big), \quad R_1 = \nu(R, G_n^r, G_{n-1}),$$
$$R_1^* = \nu(R^*, G_n^r, G_{n-1}), \quad R_2 = \nu\big(R, G_n^r, G_n^l\big), \quad R_2^* = \nu\big(R^*, G_n^r, G_n^l\big).$$

On the virtue of «inclusion-exclusion» formula (see the proof of the corollary 2, theorem 1) we derive for sets P and R that

$$N_{k_2}^0(P \cup R) = |V_n(k_2, P)| + |V_n(k_2, R)| - |V_n(k_2, P) \cap V_n(k_2, R)|. \tag{12}$$

Having proved that $P \subseteq \{G_n^l\}$, $R \subseteq \{G_n^r\}$ and taking in consideration 1a, 2a, 2b of lemma 25 we bring (12) to the following :

$$N_{k_2}^0(P \cup R) = |V_n^l(k_2, P)| + |V_n^l(k_2, R_2)| + |V_n^r(k_2, R)| -$$
$$-|V_n^l(k_2, P) \cap V_n^l(k_2, R_2)|. \tag{13}$$

Applying «inclusion-exclusion» to formula (13) yields:

$$N_{k_2}^0(P \cup R) = |V_n^l(k_2, P) \cup V_n^l(k_2, R_2)| + |V_n^r(k_2, R)|. \tag{14}$$

Consider sets P* and R_2^*.

As $R^{\circ *}$? is a + -compacted set of bundles and $\Gamma_n^{k_1}\big(C_{n-1}^{k_1-1} + 1\big) \in \{R^*\}$, then $R_2^{\circ *}$? will also be + compacted and $\Gamma_n^{k_1+1}(1) \in \{R_2^*\}$. The latter is implication from 5, lemma 6* and 4a, lemma 11.

Therefore from the corollary 1.21 it follows that sets $V_n^r(k_2, P^*)$ and $V_n^l(k_2, R_2^*)$ will coincide with the set of bundles of their k_2- extensions. Furthermore, as it is implied from lemma 14, k_2-extensions of P^* and R_2^* will be + -compacted and will include the bundle $\Gamma_n^{k_2}(1)$.

We further see that at each k_2 either

$$V_n^l(k_2, P^*) \subseteq V_n^l(k_2, R_2^*), \tag{15}$$

or

$$V_n^l(k_2, R_2^*) \subseteq V_n^l(k_2, P^*). \tag{16}$$

Having fixed k_2 we will consider each of these cases:

Assume (15) is satisfied. Then

$$|V_n^l(k_2, P^*) \cup V_n^l(k_2, R_2^*)| = |V_n^l(k_2, R_2^*)|. \tag{17}$$

From (14) we have that:

$$N_{k_2}^0(P \cup R) \geq |V_n^l(k_2, R_2)| + |V_n^r(k_2, R)|. \tag{18}$$

Using 2c, la and 2b of lemma 25, we derive from (18):

$$N_{k_2}^0(P \cup R) \geq |V_{n-1}(k_2 - 1, R_1)| + |V_{n-1}(k_2, R_1)|.$$

From 5, lemma 6* and 3a1, lemma 11 it follows that $R_1^{\circ *}$? + compacted and includes the bundle $\Gamma_{n-1}^{k_1-1}(1)$.

Therefore on the ground of our assumption on correctness of lemma 19 for Δ_{n-1}^0 –graphs we justify that:

$$|V_{n-1}(k_2, R_1^*)| \leq |V_{n-1}(k_2, R_1)|. \tag{19}$$

From (18) and (19) we obtain

$$N_{k_2}^0(P \cup R) \geq |V_{n-1}(k_2 - 1, R_1^*)| + |V_{n-1}(k_2, R_1^*)|$$

or, making an inverse transition to the sets R^* and R_2^*, we conclude:

$$N_{k_2}^0(P \cup R) \geq |V_n^l(k_2, R_2^*)| + |V_n^r(k_2, R^*)|. \tag{20}$$

From (20), using (17), we get:

$$N_{k_2}^0(P \cup R) \geq \big|V_n^l(k_2, P^*) \cup V_n^l(k_2, R_2^*)\big| + |V_n^r(k_2, R^*)| =$$
$$= N_{k_2}^0(P^* \cup R^*),$$

And it is the required inequality (11).

Now assume that (16) is met. Then

$$|V_n^l(k_2, P^*) \cup V_n^l(k_2, R_2^*)| = |V_n^l(k_2, P^*)|. \tag{21}$$

With justification by (14) we have that:

$$N^0_{k_2}(P \cup R) \geq |V^l_n(k_2, P)| + |V^r_n(k_2, R)|. \tag{22}$$

Arguing by analogy with the previous case, we obtain from (22):

$$N^0_{k_2}(P \cup R) \geq \left|V^l_n(k_2, P^*)\right| + |V^r_n(k_2, R^*)|,$$

From this inequality using (21), we come back to (11). Lemma 27 is proved.

P r o o f o f l e m m a 19.

We will express the statement of lemma 19 in the following way: for natural numbers n, k_1, k_2, m, $j_1, \dots, j_m$, if $0 \leq k_1 \leq k_2 \leq n$,

$$0 \leq m \leq C_n^{k_1},\ j_1, \dots,\quad j_m \in \{1, \dots, C_n^{k_1}\},\quad j_1 \neq j_2 \neq \dots \neq j_m,$$
$$S^* = \{\Gamma_n^{k_1}(1), \dots, \Gamma_n^{k_1}(m)\},\qquad S = \{\Gamma_n^{k_1}(j_1), \dots, \Gamma_n^{k_1}(j_m)\},$$

then

$$N^0_{k_2}(S^*) \leq N^0_{k_2}(S). \tag{1}$$

Under $k_1 = 0$, $k_1 = k_2$ or $k_2 = n$ the lemma is evident.

Assume $0 < k_1 < k_2 < n$.

The proof is conducted by induction on parameter n.

We verify the lemma under $n = 0, 1, 2$ with immediate substitution.

Assume lemma is true under n equal to $t-2$ and $t-1$, where $t \geq 3$.

Let's prove that the lemma is correct under $n = t$.

The introduced inductive assumption means, that the lemma is true for cases when the sequences S° ? and $S^{\circ *}$? enter to the k_1-tier of the $\Delta_t^{(0)}$-graph G_n under $n \in \{t-2, t-1\}$. We need to prove the correctness of the lemma in case S and S^* belong to k_1-tier of $\Delta_t^{(0)}$-graph G_t. To prove the lemma for $n = t$ we will partition the set of all possible cases, determined by position of the set S in k_1-tier G_t and the value of parameter m, to subsets given as follows:

1a. $S \subseteq \{G_t^l\}$, $\quad 1 \leq m \leq C_{t-1}^{k_1-1}$;

1b. $S \subseteq \{G_t^r\}$, $\quad 1 \leq m \leq C_{t-1}^{k_1-1}$;

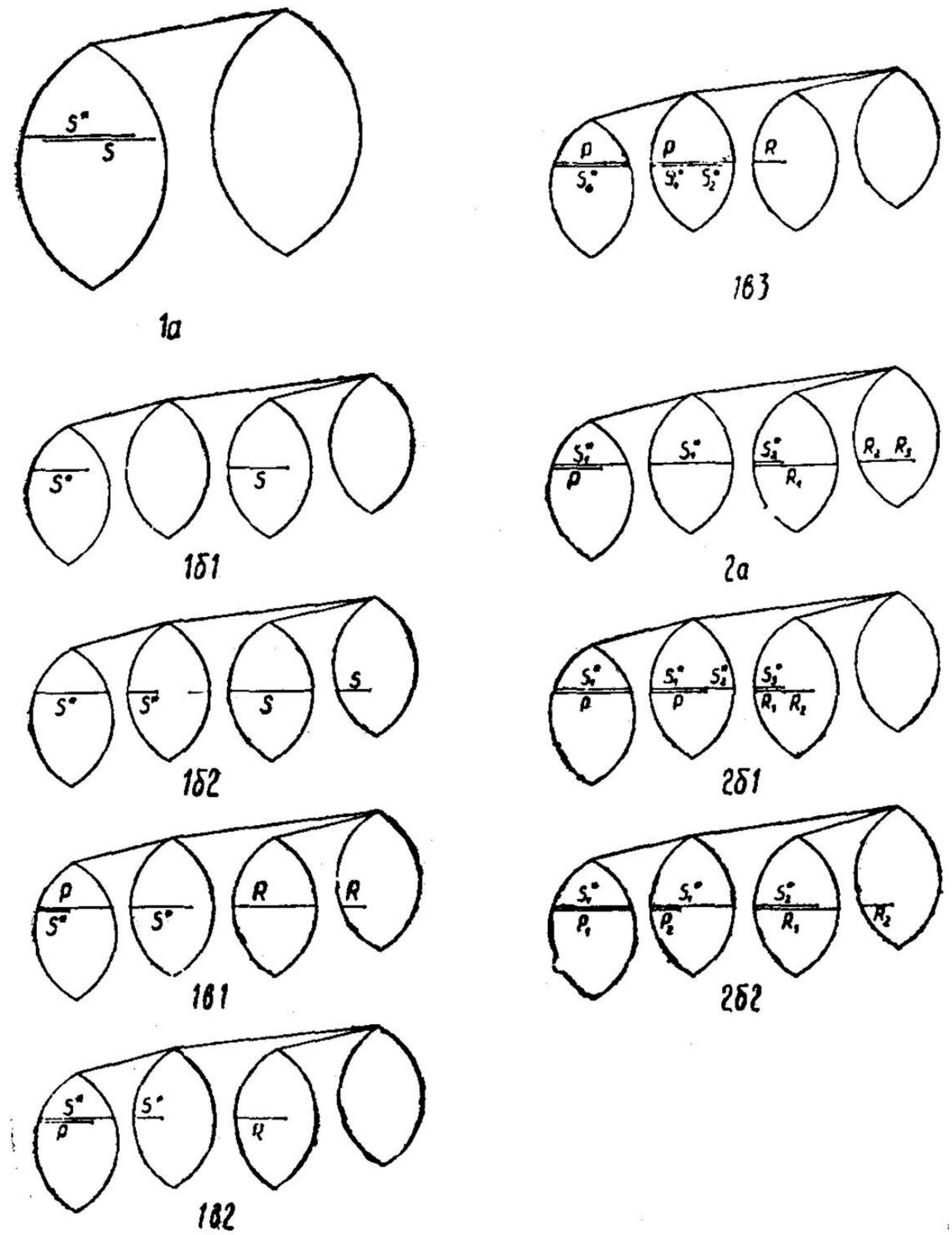
1c. $S = P \cup R$, $\quad P \subseteq \{G_t^l\}$, $\quad R \subseteq \{G_t^r\}$, $\quad |P| = m_1$,
$|R| = m_2$, $\quad m_1 + m_2 = m$, $\quad 1 \leq m \leq C_{t-1}^{k_1-1}$;

2. $C_{t-1}^{k_1-1} + 1 \leq m \leq C_t^{k_1}$.

It can be easily seen that the partition is exhaustive.

Let's consider each case separately (pic. 11).

1a. $S_1 = \nu(S, G_t^l, G_{t-1})$ и $S_1^* = \nu(S^*, G_t^l, G_{t-1})$.



Pic. 11.

As $S^{\circ *}$? is a +compacted sequence and $\Gamma_t^{k_1}(1) \in \{S^*\}$, from 5, lemma 6* and 3a1, lemma 11 we obtain that $S_2^{\circ *}$? will be a + compacted sequence and $\Gamma_{t-1}^{k_1-1}(1) \in \{S_1^*\}$.

Therefore on the ground of the inductive assumption we can write:

$$|V_{t-1}(k_2-1,\ S_1^*)| \leq |V_{t-1}(k_2-1,\ S_1)|,$$

From which using 1b of lemma 25, we come to (1).

1b. From lemma 2b it follows that it is sufficient to study the case 1b for $S = \{\Gamma_t^{k_1}\left(C_{t-1}^{k_1-1} + 1\right), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + m)\}$.

We will divide case 1b to the following sub-cases:

1b1. $1 \leq m \leq C_{t-2}^{k_1-1}$;

1b2. $C_{t-2}^{k_1-1} \leq m \leq C_{t-1}^{k_1-1}$,

1b1. CBy virtue of. 2d, lemma 11, $S \subseteq \{G_t^{rl}\}$ in this case.

Assume $S_1 = \nu\left(S, G_t^{rl}, G_t^{lr}\right)$.

From 4 of lemma 25 it follows that

$$|V_t(k_2, S)| \leq |V(k_2, S_1)|. \tag{2}$$

But S_1 meets conditions of the case 1a..

Therefore on the base of the proof we have:

$$|V_t(k_2, S^*)| \leq |V_t(k_2, S_1)|. \tag{3}$$

We obtain (1) from (3) and (2).

1b2. Assume $T = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1})\}$.

It is evident that:

$$N_{k_2}^0(T) \leq N_{k_2}^0(S). \tag{4}$$

From 2d, lemma 11, it follows that $T \subseteq \{G_t^{rl}\}$.

Assume $S_1^* = \nu(S^*, G_t^l, G_{t-1})$, $T_1 = \nu(T, G_t^{rl}, G_{t-2})$.

With justification of 1b of lemma 25 we conclude that

$$|V_t(k_2, S^*)| \leq |V_{t-1}(k_2 - 1, S_1^*)|. \tag{5}$$

It is clear, that:

$$|V_{t-1}(k_2 - 1, S_1^*)| \leq C_{t-1}^{k_2-1}. \tag{6}$$

For the set T, using assertions 3 and 4 of lemma 25, we obtain:

$$|V_t(k_2, T)| = |V_{t-2}(k_2 - 1, T_1)| + |V_{t-2}(k_2 - 2, T_1)|. \tag{7}$$

Apparently the following equality holds:

$$|T_1| = |T| = C_{t-2}^{k_2-1},$$

That is T_1 contains all bundles of the $(k_1 - 1)$-tier of the Δ_{t-2}^0-graph G_{t-2}.

Consequently

$$|V_{t-2}(k_2 - 1, T_1)| = C_{t-2}^{k_2-1}. \tag{8}$$

From (7) and (8) we obtain that:

$$|V_t(k_2, T)| = C_{t-2}^{k_2-1} + C_{t-2}^{k_2-2} = C_{t-1}^{k_2-1}. \tag{9}$$

It is implied from (5), (6) and (9) that:

$$|V_t(k_2, S^*)| \leq |V_t(k_2, T)|.$$

The last inequality and the inequality (4) provide (1).

1c. It is implied from lemma 27 that it is sufficient to consider the case 1c for the sets P and R, which have the following form:

$$P = \{\Gamma_t^{k_1}(1), \dots, \Gamma_t^{k_1}(m_1); \quad R = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + m_2)\}.$$

We will divide the case 1c to the following sub-cases:

1c1. $m_2 \geq C_{t-2}^{k_1-1}$;

1c2. $m_2 < C_{t-2}^{k_1-1}$ и $m_1 \leq C_{t-2}^{k_1-2}$;

1c3. $m_2 < C_{t-2}^{k_1-1}$ и $m_1 > C_{t-2}^{k_1-2}$.

Study these cases.

1c1. Assume $T = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1})\}$.

Arguing in the way similar to that for 1b2, we sequentially obtain:

$$N_{k_2}^0(S) = N_{k_2}^0(P \cup R) \geq N_{k_2}^0(R) \geq N_{k_2}^0(T),$$

$$N_{k_2}^0(T) = C_{t-1}^{k_2-1},$$

$N_{k_2}^0(S^*) \leq C_{t-1}^{k_2-1}$,

from which (1) follows.

1c2. From2e, lemma 11 it is implied that $R \subseteq \{G_t^{rl}\}$ for the case.

Assume $R_1 = \nu(R, G_t^{rl}, G_t^{lr})$, $S_t^* = \nu(S^*, G_t^l, G_{t-1})$, $S_1 = \nu(P \cup R_1, G_t^l, G_{t-1})$.

On the ground of the inclusion-exclusion formula for sets P and R, respectively P and R_1, we obtain that:

$$N^0_{k_2}(P\cup R)=|V_t(k_2,P)|+|V_t(k_2,R)|-|V_t(k_2,P)\cap V_t(k_2,R)|. \quad (10)$$

$$N^0_{k_2}(P\cup R_1)=|V_t(k_2,P)|+|V_t(k_2,R_1)|-|V_t(k_2,P)\cap V_t(k_2,R_1)|. \quad (11)$$

As $P\subseteq\{G^{ll}_t\}$, with consideration of assertions of 2 and 3 of lemma 25, applying (10) and (11) we get:

$$\begin{gathered}N^0_{k_2}(P\cup R)=\left|V^{ll}_t(k_2,P)\right|+\left|V^{ll}_t(k_2,R)\right|+\left|V^{rl}_t(k_2,R)\right|-\\ -\left|V^{ll}_t(k_2,P)\cap V^{ll}_t(k_2,R)\right|,\end{gathered} \quad (12)$$

$$\begin{gathered}N^0_{k_2}(P\cup R_1)=\left|V^{ll}_t(k_2,P)\right|+\left|V^{ll}_t(k_2,R_1)\right|+\left|V^{lr}_t(k_2,R_1)\right|-\\ -\left|V^{ll}_t(k_2,P)\cap V^{ll}_t(k_2,R_1)\right|.\end{gathered} \quad (13)$$

Remembering that $R_1=\nu(R,G^{rl}_t,G^{lr}_t)$ and using the assertion 4 of lemma 25 we obtain:

$$V^{ll}_t(k_2,R)=V^{ll}_t(k_2,R_1),$$
$$|V^{rl}_t(k_2,R)|=|V^{lr}_t(k_2,R_1)|.$$

These equalities with taking into account (12) and (13) mean that:

$$N^0_{k_2}(P\cup R)=N^0_{k_2}(P\cup R_1). \quad (14)$$

On the ground of 1b, lemma 25 for the sets S^* and $(P\cup R_1)$ we get that:

$$N^0_{k_2}(S^*)=|V_{t-1}(k_2-1,S^*_1)|, \quad (15)$$

$$N^0_{k_2}(P\cup R_1)=|V_{t-1}(k_2-1,S_1)|. \quad (16)$$

As $S^{\circ *}$? is a + -compacted sequence and $\Gamma^{k_1}_t(1)\in S^*$, assertions. 5, lemma 6* and 3a1, lemma 11 imply that $S^{\circ *}_1$? is as well a + -compacted sequence and $\Gamma^{k_1-1}_{t-1}(1)\in S^*_1$. Consequently on the base of our inductive assumption, we have that:

$$|V_{t-1}(k_2-1,S^*_1)|\le|V_{t-1}(k_2-1,S_1)|,$$

which with consideration of (14), (15) and (16) leads to (1).

1c3. We represent set S^* in the following form: $S^*=S^*_1\cup S^*_2$, where $S^*_1=P$ and $S^*_2=\{\Gamma^{k_1}_t(m_1+1),\dots,\Gamma^{k_1}_t(m)\}$.

As $S^{\circ *}$? is a + -compacted sequence and $\Gamma^{k_1}_t(1)\in S^*$, from corollary 1. 21we have that

$$N^0_{k_2}(S^*)=\psi_{k_2}(S^*),$$

From which by using lemma 17 and aapplying the second time corollary 1.21, we get:

$$N^0_{k_2}(S^*)=N^0_{k_2}(S^*_1)+\psi_{k_2}(S^*_2). \quad (17)$$

Applying the "inclusion-exclusion" formula to sets P and R, we observe that:

$$N^0_{k_2}(P\cup R)=|V_t(k_2,P)|+|V_t(k_2,R)|-|V_t(k_2,P)\cap V_t(k_2,R)|. \quad (18)$$

As $P\subseteq\{G^l_t\}$, $R\subseteq\{G^{rl}_t\}$ (lemma11, point 2a and 2e), by virtue of assertions 1a and 4d, lemma 25, from (18) we derive:

$$\begin{gathered}N^0_{k_2}(P\cup R)=\left|V^l_t(k_2,P)\right|+\left|V^{ll}_t(k_2,R)\right|+\left|V^{rl}_t(k_2,R)\right|-\\ -\left|V^l_t(k_2,P)\cap V^{ll}_t(k_2,R)\right|.\end{gathered} \quad (19)$$

As $m_1=|P|>C^{k_1-2}_{t-2}$ and, hence, $\{\Gamma^{k_1}_t(1),\dots,\Gamma^{k_1}_t(C^{k_1-2}_{t-2})\}\subseteq P$, we can easily derive that with some k_2 all $\Delta^{k_2}_t$ – super-bundles belonging to$\{G^{ll}_t\}$, would be in turn $\Delta^{k_2}_t$ –super-bundles of some bundles from P. Therefore $V^{ll}_t(k_2,R)\subseteq V^l_t(k_2,P)$ и $N^0_{k_2}(P\cup R)=\left|V^l_t(k_2,P)\right|+\left|V^{rl}_t(k_2,R)\right|$, or

$$N^0_{k_2}(P\cup R)=N^0_{k_2}(P)+|V^{rl}_t(k_2,R)|. \quad (20)$$

As $m_1>C^{k_1-2}_{t-2}$ and $m\le C^{k_1-1}_{t-1}$, then it follows from 2d, lemma 11, that $S^*_2\subseteq\{G^{lr}_t\}$.

Assume $S^*_3=\nu(S^*_2,G^{lr}_t,G_{t-2})$ и $R_1=\nu(R,G^{rl}_t,G_{t-2})$.

We derive from corollaries 1.10 and 1.6* that:

$$\psi_{k_2}(S_2^*) = \psi_{k_{2-1}}(S_2^*). \tag{21}$$

At the same time on the ground of 4f lemma 25 we have that:

$$\left|V_t^{rl}(k_2, R) = V_{n-2}(k_2 - 1, R_1)\right|. \tag{22}$$

Note that while using corollaries 1.10 and 1.6* we have not specified the cases when roots of graphs were among the considered bundles due to triviality of their parsing.

As R° ? is a $+$ -compacted sequence and $\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1) \in R$, we can imply from assertions 5, lemma 6* and 3c3, lemma 11 that $\mathring{R_1}$? will also be $+$ -compacted and $\Gamma_{n-2}^{k_1-1} + (1) \in R_1$.

It is easy to observe that $|R_1| = |S_3^*|$ and bundles from R_1 and S_3^* have an equal length.

Therefore we obtain from lemma 8* that:

$$\psi_{k_2-1}(R_1) \geq \psi_{k_{2-1}}(S_3^*), \tag{23}$$

And from corollary 1.21 we get that:

$$\psi_{k_2-1}(R_1) = |V_{n-2}(k_2 - 1, R_1)|. \tag{24}$$

From (23) with taking into account (24), (22) and (21) we observe that:

$$\psi_{k_2}(S_2^*) \leq |V_t^{rl}(k_2, R)|. \tag{25}$$

As $P = S_1^*$ then (1) is implied from 17), (20) and (25).

Thus case 1 is completely examined.

Consider case 2. The inequality $C_{t-1}^{k_1-1} + 1 \leq m \leq C_t^{k_1}$ holds in this case. This means that (see 2a, lemma 11), both right and left vertices of the graph G_t belong to the set S. Therefore the set S can be partitioned to two sub-sets P and R, such that $P \subseteq \{G_t^l\}$, $R \subseteq \{G_t^r\}$ and $P \cup R = S$.

Assume $|P| = m_1, |R| = m_2$. Evidently $m_1 \leq C_{t-1}^{k_1-1}$.

As it follows from lemma 27 the sufficient condition for study of case 2 is the case $P = \{\Gamma_t^{k_1}(1), \dots, \Gamma_t^{k_1}(m_1)\}$,

$$R = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-2} + m_2)\}.$$

Depending on parameter m_1 we will divide the case to the following sub-cases:

2a. $1 \leq m_1 \leq C_{t-2}^{k_1-2}$,

2b. $C_{t-2}^{k_1-2} < m_1 \leq C_{t-1}^{k_1-1}$.

Consider each of these cases.

Denote the difference $m_1 - C_{t-1}^{k_1-1}$ as r, definitely $r > 0$.

2a. Denote the difference $C_{t-2}^{k_1-2} - m_1$as *x*.

Let's express the parameter m_2 in terms of r and x:

$$m_2 = m - m_1 = \left(r + C_{t-1}^{k_1-1}\right) - \left(C_{t-2}^{k_1-2} - x\right) = r + x + C_{t-2}^{k_1-1}. \tag{26}$$

Sets S^* and R we will represent in the following form:

$$S^* = S_1^* \cup S_2^*,$$

where

$$S_1^* = \{\Gamma_t^{k_1}(1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1})\},$$
$$S_2^* = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + r)\},$$
$$R = R_1 \cup R_2 \cup R_3,$$

where:

$$R_1 = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1})\},$$
$$R_2 = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1} + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1} + r)\},$$
$$R_3 = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1} + r + 1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1} + r + x)\}.$$

As $m > C_{t-1}^{k_1-1}$ and (26) is true, sets S^* and R can be represented in the indicated form.

By analogy with the case 1c3, the set S^* can be written in the form similar (17), i.e.

$$N_{k_2}^0(S^*) = N_{k_2}^0(S_1^*) + \psi_{k_2}(S_2^*). \tag{27}$$

As $|S_1^*| = C_{t-1}^{k_1-1}$ and $S_1^* \subseteq \{G_t^l\}$, then using the assertions 2b, lemma 25 and 3a1, lemma 11, we obtain that:

$$N_{k_2}^0(S_1^*) = C_{t-1}^{k_1-1}. \tag{28}$$

It is implied from the assertion 2, lemma 11, that $R_1 \subseteq \{G_t^{rl}\}$, $R_2 \subseteq \{G_t^{rr}\}$.

Assume $R_2' = \nu(R_2, G_t^r, G_t^l)$ From 4a, lemma 11, we obtain that $R_2' = \{\Gamma_t^{k_1+1}(C_{t-2}^{k_1-1} + 1), \dots, \Gamma_t^{k_1+1}(C_{t-2}^{k_1-1} + r)\}$, while from 2d of the same lemma it follows that $R_2' \subseteq \{G_t^{lr}\}$.

Using assertions 4d, 5a, 5b, lemma 25, we obtain for sets R_1 and R_2 that:

$$V_t(k_2, R_1) = V_t^{ll}(k_2, R_1) \cup V_t^{rl}(k_2, R_1), \tag{29}$$

$$\begin{gathered} V_t(k_2, R_2) \supseteq (V_t^l(k_2, R_2) \cup (V_t^{rr}(k_2, R_2)) = \\ = (V_t(k_2, R_2') \cup (V_t^{rr}(k_2, R_2)) \supseteq (V_t^{lr}(k_2, R_2') \cup (V_t^{rr}(k_2, R_2)). \end{gathered} \tag{30}$$

As

$$V_t(k_2, P \cup R) \supseteq V_t(k_2, R) \supseteq V_t(k_2, R_1) \cup V_t(k_2, R_2)), \tag{31}$$

Then by substituting (29) and (33) into (31), we derive that:

$$\begin{gathered} N_{k_2}^0(P \cup R) \geq |V_t^{ll}(k_2, R_1)| + |V_t^{rl}(k_2, R_1)| + |V_t^{lr}(k_2, R_2')| + \\ + |V_t^{rr}(k_2, R_2)| = |V_t(k_2, R_1)| + |V_t^{lr}(k_2, R_2')| + |V_t^{rr}(k_2, R_2)|. \end{gathered} \tag{32}$$

In analogy with the case 1b2, we conclude for the set R_1 that:

$$|V_t(k_2, R_1)| = C_{t-1}^{k_1-1}. \tag{33}$$

Assume $R_2'' = \nu(R_2', G_t^{lr}, G_t^{rl})$. Clearly $R_2'' = \nu(R_2, G_t^{rr}, G_t^{rl})$.

Using 4c, lemma 11, we derive that:

$$R_2'' = \{\Gamma_t^{k_1+1}(C_{t-1}^{k_1} + 1), \dots, \Gamma_t^{k_1+1}(C_{t-1}^{k_1} + r)\}.$$

It can be implied from assertions 4c, 4f an 4i of lemma 25 that:

$$|V_t^{lr}(k_2, R_2') = |V_t^{rl}(k_2, R_2'')|. \tag{34}$$

Assume $T_1 = \nu(S_2'', G_t^r, G_{t-1})$, $T_2 = \nu(R_2'', G_t^r, G_{t-1})$ and $T_3 = \nu(R_2, G_t^r, G_{t-1})$.

By applying 3a2, lemma 11, we derive:

$$\begin{gathered} T_1 = \{\Gamma_{t-1}^{k_1}(1), \dots, \Gamma_{t-1}^{k_1}(r)\}, \qquad T_2 = \{\Gamma_{t-1}^{k_1+1}(1), \dots, \Gamma_{t-1}^{k_1+1}(r)\}, \\ T_3 = \{\Gamma_{t-1}^{k_1}(C_{t-1}^{k_1-1} + 1), \dots, \Gamma_{t-1}^{k_1}(C_{t-1}^{k_1-1} + r)\}. \end{gathered}$$

From the assertion 4a, lemma 11, follows that $T_2 = \nu(T_2, G_{t-1}^r, G_{t-1}^l)$. Therefore on the ground of 1 and 2 of lemma 25 we have that:

$$|V_{t-1}^l(k_2, T_2) + |V_{t-1}^r(k_2, T_3) = |V_{t-1}(k_2, T_3)|. \tag{35}$$

Proceeding from the inductive assumption we make conclusion for sets T_1 and T_2:

$$|V_{t-1}(k_2, T_1)| \leq |V_{t-1}(k_2, T_3)|. \tag{36}$$

Under isomorphism of graphs G_t^r and G_{t-1}, the sub-graphs G_t^{rl} and G_{t-1}^l, G_t^{rr}, G_{t-1}^r correspond each other, taking into account the assertion 2, lemma 24, we obtain that:

$$|V_t^{rl}(k_2, R_2'') = |V_{t-1}^l(k_2, T_2)|, \tag{37}$$

$$|V_t^{rr}(k_2, R_2) = |V_{t-1}^r(k_2, T_3)|. \tag{38}$$

At the same time for sets S_2^* and T_1 we derive from corollaries 1.10 and 1.6* that:

$$\psi_{k_2}(S_2^*) = \psi_{k_2}(T_1)$$

Or with justification of the corollary 1.21

$$\psi_{k_2}(S_2'') = |V_{t-1}(k_2, T_1)|. \tag{39}$$

By substituting (37) and (38) to (35) and then the resulting expression and (39) to (36), we obtain:

$$\psi_{k_2}(S_2^*) \le |V_t^{rl}(k_2, R_2'') + |V_t^{rr}(k_2, R_2)|$$

Or taking into account (34)

$$\psi_{k_2}(S_2^*) \le |V_t^{lr}(k_2, R_2') + |V_t^{rr}(k_2, R_2)|. \tag{40}$$

Comparing (27) with (32) and taking into account (28), (33) and (40), we come to (1).

2b. Let's introduce the following notations:

$$x = m_1 - C_{t-2}^{k_1-2}, \qquad y = C_{t-1}^{k_1-1} - m_1.$$

Then

$$m_2 = m - m_1 = C_{t-1}^{k_1-1} + r - m_1 = r + y. \tag{41}$$

Depending on the value of the parameter m_2 we will divide the case 2b to the following sub-cases:

2б1. $m_2 \le C_{t-2}^{k_1-1}$,

2б2. $m_2 > C_{t-2}^{k_1-1}$.

Consider each of these cases.

2b1. Let's represent the set S^* in the following form:

$$S^* = S_1^* \cup S_2^* \cup S_3^*,$$

where

$$S_1^* = P; \quad S_2^* = \{\Gamma_t^{k_1}(m_1+1), \dots, \Gamma_t^{k_1}(m_1+y)\};$$
$$S_3^* = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1}+1), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1}+r)\}.$$

From definitions of parameters r and y it is easily seen that S^* can be presented in the indicated form.

Using the corollary 1.21 and lemma 17 for the sets S^* we obtain:

$$N_{k_2}^0(S^*) = N_{k_2}^0(S_1^*) + \psi_{k_2}(S_2^*) + \psi_{k_2}(S_3^*). \tag{42}$$

For the set $S = P \cup R$ the following relation can be written:

$$V_t(k_2, S) = V_t(k_2, P) \cup V_t(k_2, R). \tag{43}$$

It follows from assertion 2, lemma 11 that $P \subseteq \{G_t^l\}$ and $R \subseteq \{G_t^{rl}\}$. Then on the ground of 1a and 4d of lemma 25 from (43) we find:

$$V_t(k_2, S) \supseteq (V_t^l(k_2, P) \cup V_t^{rl}(k_2, R)),$$

it follows then that:

$$N_{k_2}^0(S) \ge \left|V_t^l(k_2, P)\right| + \left|V_t^{rl}(k_2, R)\right|,$$

or

$$N_{k_2}^0(S) \ge N_{k_2}^0(P) + \left|V_t^{rl}(k_2, R)\right|. \tag{44}$$

Assume $S_2' = \nu(S_2^*, G_t^{lr}, G_t^{rl})$. From lemma 11 we get that:

$$S_2' = \{\Gamma_t^{k_1}(m_1 + 1 + C_{t-2}^{k_1-1}), \dots, \Gamma_t^{k_1}(C_{t-1}^{k_1-1} + C_{t-2}^{k_1-1})\}.$$

Having determined that $y = C_{t-1}^{k_1-1} - m_1$, we derive the following:

$$|S_2'| = y. \tag{45}$$

Then we find from corollaries 1.10 and 1.6* that

$$\psi_{k_2}(S_2^*) = \psi_{k_2}(S_2'). \tag{46}$$

Assume $R' = \nu(R, G_t^{lr}, G_{t-2})$, $S_3' = \nu(S_3^*, G_t^{lr}, G_{t-2})$, $S_2'' = \nu(S_2', G_t^{lr}, G_{t-2})$.

From 3b3, lemma 11 we determine that:

$$R' = \{\Gamma_{t-2}^{k_1-1}(1), \dots, \Gamma_{t-2}^{k_1-1}(m)\}, \quad S_3' = \{\Gamma_{t-2}^{k_1-1}(1), \dots, \Gamma_{t-2}^{k_1-1}(r)\},$$

$S_2'' = \{\Gamma_{t-2}^{k_1-1}(m_1+1), \dots, \Gamma_{t-2}^{k_1-1}(C_{t-2}^{k_1-1})\}$.

We apply 4f of lemma 24, corollaries 1.10 and 1.6* we find:

$$\left|V_t^{rl}(k_2,R)\right| = |V_{t-2}(k_2-1,R')|, \tag{47}$$
$$\psi_{k_2}(S_3'') = \psi_{k_2-1}(S_3'), \tag{48}$$
$$\psi_{k_2}(S_2') = \psi_{k_2-1}(S_2''). \tag{49}$$

We represent the set R'in the following form:

$$R' = R_1' \cup R_2',$$

where

$$R_1' = S_3', \qquad R_2' = \{\Gamma_{t-2}^{k_1-1}(r+1),\dots,\Gamma_{t-2}^{k_1-1}(m_2)\}.$$

From (41) it follows that:

$$|R_2'| = y,$$

Then with justification of equalities $|S_2^*| = |S_2'|$ and (45) we see that

$$|R_2'| = |S_2^*|. \tag{50}$$

Applying lemma 7 and corollary 1.21 to sets R' and S_3' we find:

$$|V_{t-2}(k_2-1,R')| = |V_{t-2}(k_2-1,R_1')| + \psi_{k_2-1}(R_2'), \tag{51}$$
$$\psi_{k_2}(S_2') = |V_{t-2}(k_2-1,S_3')|. \tag{52}$$

From lemma 8*, we get for sets R_2' and S_2^* that:

$$\psi_{k_2-1}(S_2^*) \le \psi_{k_2-1}(R_2'). \tag{53}$$

From (51) with (52) and (53) given that:

$$|V_{t-2}(k_2-1,R')| \ge \psi_{k_2-1}(S_3') + \psi_{k_2-1}(S_2^*), \tag{54}$$

Or after corresponding substitutions of (47), (48), (49) and (46) to (54), we have:

$$\left|V_t^{rl}(k_2,R)\right| \ge \psi_{k_2}(S_3^*) + \psi_{k_2}(S_2^*). \tag{55}$$

As $P = S_1^*$ and given (55) and (42) we find (1) from (44).

2b2. Represent sets S^*, P and R in the following form:

$$S^* = S_1^* \cup S_2^*; \quad P = P_1 \cup P_2; \quad R = R_1 \cup R_2,$$

where

$$S_1^* = \{\Gamma_t^{k_1}(1),\dots,\Gamma_t^{k_1}(C_{t-1}^{k_1-1})\}; \quad S_2^* = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1}+1),\dots,\Gamma_t^{k_1}(C_{t-1}^{k_1-1}+r)\};$$
$$P_1 = \{\Gamma_t^{k_1}(1),\dots,\Gamma_t^{k_1}(C_{t-2}^{k_1-2})\}; \quad P_2 = \{\Gamma_t^{k_1}(C_{t-2}^{k_1-2}+1),\dots,\Gamma_t^{k_1}(C_{t-2}^{k_1-2}+x)\};$$
$$R_1 = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1}+1),\dots,\Gamma_t^{k_1}(C_{t-1}^{k_1-1}+C_{t-2}^{k_1-1})\};$$
$$R_2 = \{\Gamma_t^{k_1}(C_{t-1}^{k_1-1}+C_{t-2}^{k_1-1}+1),\dots,\Gamma_t^{k_1}(C_{t-1}^{k_1-1}+C_{t-2}^{k_1-1}+r-x)\}.$$

The indicated representation of sets S^* P and R immediately follows from definition of parameters r and x.

Using lemma 7 and corollary 1.21 for the set S^*, we obtain:

$$N_{k_2}^0(S^*) = N_{k_2}^0(S_1^*) + \psi_{k_2}(S_2^*). \tag{56}$$

For the set $S = P \cup R$ we can write the following relation.

$$V_t(k_2,S) \supseteq (V_t^{ll}(k_2,P_1) \cup V_t^{lr}(k_2,P_2) \cup V_t^{rl}(k_2,R_1) \cup$$
$$\cup V_t^{lr}(k_2,R_2) \cup V_t^{rr}(k_2,R_2)).$$

From this expression we get that:

$$N_{k_2}^0(S) \ge |V_t^{ll}(k_2,P_1)| + |V_t^{lr}(k_2,P_2) \cup V_t^{lr}(k_2,R_2)| +$$
$$+|V_t^{rl}(k_2,R_1)| + |V_t^{rr}(k_2,R_2)|. \tag{57}$$

From 2, lemma 11, it follows that:

$$P_1 \subseteq \{G_t^{ll}\}, \qquad P_2 \subseteq \{G_t^{lr}\}, \qquad R_1 \subseteq \{G_t^{rl}\}, \qquad R_2 \subseteq \{G_t^{rr}\}.$$

In the way similar to the case 2a we get for the set S_2^*:

$$N_{k_2}^0(S^*) = C_{t-1}^{k_1-1}. \tag{58}$$

As $|P_1| = C_{t-2}^{k_1-2}$ and $|R_1| = C_{t-2}^{k_1-1}$, using lemmas 11 and 25, we can easily determine that:

$$|V_t^{ll}(k_2,P_1)|+|V_t^{rl}(k_2,P_1)|=C_{t-1}^{k_1-1}. \tag{59}$$

Let's prove that:

$$\psi_{k_2}(S_2^*)\le |V_t^{lr}(k_2,P_2)\cup V_t^{lr}(k_2,R_2)|+|V_t^{rr}(k_2,R_2)|. \tag{60}$$

Assume

$$R_2'=\nu(R_2,G_t^r,G_t^l),\quad R_2''=\nu(R_2,G_t^{rr},G_t^{rl}),\quad P_2'=\nu(P_2,G_t^{lr},G_t^{rl}),$$

Посредством утверждений п. 2 и 4 леммы 25 легко проверить, что
$R_2'\subseteq\{G_t^{lr}\}$ и $R_2''=\nu(R_2',G_t^{lr},G_t^{rl})$.

Используя утверждения лемм 23 и 24, заключаем, что

$$|V_t^{lr}(k_2,P_2)|\cup|V_t^{lr}(k_2,R_2)|=|V_t^{rl}(k_2,P_2')|\cup|V_t^{rl}(k_2,R_2'')|. \tag{61}$$

Пусть

$$S_2'=\nu(S_2'',G_t^r,G_{t-1}),\quad T_1=\nu(P_2',G_t^r,G_{t-1}),\quad T_2=\nu(R_2'',G_t^r,G_{t-1})$$

и

$$T_3=\nu(R_2,G_t^r,G_{t-1}).$$

Так как при изоморфизме G_t^r и G_{t-1}, подграфы G_t^{rl} и G_{t-1}^l, G_t^{rr} и G_{t-1}^r соответствуют друг другу, то используя п. 2 леммы 24, получаем, что

$$|V_t^{rl}(k_2,P_2')\cup V_t^{rl}(k_2,P_2'')|=|V_{t-1}^l(k_2,T_1)\cup V_{t-1}^l(k_2,T_2)|, \tag{62}$$

$$|V_t^{rr}(R_2)|=|V_{t-1}^r(T_3)|. \tag{63}$$

Так как $R_2''=\nu(R_2,G_t^{rr},G_t^{rl})$, то легко получить, что

$$T_2=\nu(T_3,G_{t-1}^r,G_{t-1}^r).$$

Поэтому, из п. 1 и 2 леммы 25 следует, что

$$|V_{t-1}^l(k_2,T_1)\cup V_{t-1}^l(k_2,T_2)|+|V_{t-1}^r(k_2,T_3)|=|V_{t-1}(k_2,T_1\cup T_3)|. \tag{64}$$

Из п. 3а2 леммы 11 получаем, что

$$S_2'=\{\Gamma_{t-1}^{k_1}(1),\dots,\Gamma_{t-1}^{k_1}(r)\}.$$

Так как

$$|S_2'|=|T_1\cup T_3|$$

и наборы из S_2' и $T_1\cup T_3$ одинаковой длины, то на основе индуктивного предположения имеем:

$$N_{k_2}^0(S_2')\le|V_{t-1}(k_2,T_1\cup T_3)|. \tag{65}$$

В то же время, как следует из следствий 1.10 и 1.6* и следствия 1.21,

$$\psi_{k_2}(S_2^*)=N_{k_2}^0(S_2'). \tag{66}$$

Substituting expressions (66), (64), then (62) and (63) into (65) and taking into account (61) we get (60).

From (57) with given (59) and (60) we obtain:

$$N_{k_2}^0(S)\ge C_{t-1}^{k_1-1}+\psi_{k_2}(S_2^*), \tag{65}$$

And it along with (56) and (58) proves (1).

Thus, case 2 is completely parsed as well as all possible cases.

The inductive step is conducted completely.

Lemma 19 and at the same time lemma 19* are proved.

L e m m a 28. $L_\chi=\{L_n^{l_1}(1),L_n^{l_1}(2)\dots,L_n^{l_1}(m_1),L_n^{l_2}(\widetilde{m}_1+1),\dots,L_n^{l_2}(\widetilde{m}_1+m_2),\dots,L_n^{l_k}(\widetilde{m}_{k-1}+1),\dots,L_n^{l_2}(\widetilde{m}_{k-1}+m_k)\}$ *is a compacted set with c.s.* $\chi=(n,l_1,\dots,l_k,\ m_1,\dots,m_k)$, *then*:

1. *Bundles of the set* L_χ *are pairwise incompatible;*
2. *The set* L_χ *can be represented in the form*

$$L_\chi=\{L_n^{l_1}(1),\ L_n^{l_1}(2)\dots,L_n^{l_1}(t_{l_1}),\ L_n^{l_2}\big(\theta(n-l_2,n-l_1,1,\ t_{l_1})+1\big),\dots,$$

$$L_n^{l_2}(t_{l_2}),\dots,L_n^{l_k}\big(\theta(n-l_k,n-l_{k-1},\ t_{l_{k-1}})+1\big),\dots,L_n^{l_2}(\ t_{l_k})\},$$

where

$t_{l_1} = m_1$; $t_{l_t} = \theta(n - l_i, n - l_{i-1}, 1, t_{l_{i-1}})$ at $i = 2, \dots, k$; $\theta(k_2, k_1, r_1, r_2)$ – the number of bundles in k_2-extension in the sequence $\Gamma_n^{k_1}(r_1), \dots, \Gamma_n^{k_1}(r_1+1), \dots, \Gamma_n^{k_1}(r_2)$ (see lemma 14).

The assertion of lemma 28 directly follows from the definition of compacted sets, corollary 1.21 and definition of function $\theta(x, y, z, u)$.

It should be noted also that with fixed c.s. denotations $t_{l_1}, \dots, t_{l_k}$ in the sequel will be used only in defined sense.

L e m m a 29. *If S is an arbitrary compacted set with c.s. $\chi = (n, l_1, \dots, l_k, m_1, \dots, m_k)$, then the set of all $\Delta_n^{l_k}$-sub-bundles of bundles from S coincides with the set* $\{L_n^{l_k}(1), \dots, L_n^{l_k}(t_{l_k})\}$.

P r o o f.

It is implied from lemma 20 and connection between algorithms L_n^l и Γ_n^{n-1} that to prove lemma 29 it is sufficient to prove that the set of all $\Delta_n^{n-l_k}$-super-bundles of complements of bundles from S coincides with the set

$$\{\Gamma_n^{n-l_k}(1), \dots, \Gamma_n^{n-l_k}(t_{l_k})\}.$$

Using the definition of the compacted set we obtain that the set of complements of of bundles from S is the set:

$$S' = \{\Gamma_n^{n-l_1}(1), \dots, \Gamma_n^{n-l_1}(t_{l_1}), \dots, \Gamma_n^{n-l_k}(\theta(n - l_k, n - l_{k-1}, 1, t_{l_{k-1}}) + \\ +1), \dots, \Gamma_n^{n-l_k}(t_{l_k})\}.$$

On the ground of corollary 1.21 and definition of the function $\theta(x, y, z, u)$ for the set S' we sequentially find:

$$V_n(n - l_1, S') = \{\Gamma_n^{n-l_1}(1), \dots, \Gamma_n^{n-l_1}(t_{l_1})\},$$
$$V_n(n - l_2, S') = V_n(n - l_2, V_n(n - l_1, S')) =$$
$$= \{\Gamma_n^{n-l_2}(1), \dots, \Gamma_n^{n-l_2}(\theta(n - l_2, n - l_1, 1, t_{l_1})),$$
$$\Gamma_n^{n-l_2}(\theta(n - l_2, -l_1, 1, t_{l_1})+), \dots, \Gamma_n^{n-l_2}(t_{l_2})\},$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$V_n(n - l_k, S') = V_n(n - l_k, V_n(n - l_{k-1}, S')) =$$
$$= \{\Gamma_n^{n-l_k}(1), \dots, \Gamma_n^{n-l_k}(\theta(n - l_k, n - l_{k-1}, 1, t_{l_{k-1}})),$$
$$\Gamma_n^{n-l_k}(\theta(n - l_k, n - l_{k-1}, 1, t_{l_{k-1}}) + 1), \dots, \Gamma_n^{n-l_k}(t_{l_k})\},$$

Where $V_n(l, S')$ is equal to the set of Δ_n^0-super-bundles of bundles from S'.

The assertion of lemma 29 follows from the last equality.

For an arbitrary set of Δ_n-bundles T as $W_l(T)$ we will denote the set of all Δ_n^l-sub-bundles of bundles of the set T, where $0 \le i \le n$.

P r o o f o f t h e o r e m 1.

Let K_χ be a non –empty class s. p. i. b. for χ and S be an arbitrary element K_χ.

We will demonstrate that a compacted set L_χ with c.s x. χ can be produced. .

As S is s. p. i. b. then the following inequalities are true for S:

$$N_i(S) < C_n^i \quad \text{at} \qquad l_k + 1 \le i \le n. \tag{1}$$

$$N_{l_k}(S) \le C_n^{l_k}, \tag{2}$$

Since violation of these inequalities would lead to disturbing the condition of pairwise incompatibility of bundles from S.

The following equalities are evident for sets S and L_χ.

$$W_i(S) = W_i\left(\bigcup_{j=1}^{r} P_j\right), \tag{3}$$

$$W_i(S) = W_i\left(\bigcup_{j=1}^{r} Q_j\right), \tag{4}$$

Where

$$P_j = \{A \mid A \in S \& |A| = l_j\}, \quad Q_j = \{A \mid A \in L_\chi \& |A| = l_j\}$$

with $j = 1, \dots, k;\quad l_{r+1} + 1 \le i \le n,\quad 1 \le r \le k-1.$

...

Since

$$|P_1| = |Q_1| = m_1,$$

with (I) and (3) given $i = l_1$ we obtain that the set $Q_1 = \{L_n^{l_1}(1), \dots, L_n^{l_1}(t_{l_1})\}$ can be produced.

From lemma 19*, relations (1), (3), (4) at $r = 1;\ l_2 + 1 \le i \le n$, we find that:

$$N_i(L_\chi) = N_i(Q_1) \le N_i(P_1) = N_i(S) < C_n^l\,. \tag{5}$$

As

$W_{t_2}(L_\chi) = W_{t_2}(Q_1) \cup Q_2,\quad W_{t_2}(Q_1) \cap Q_2 = \emptyset,\quad W_{t_2}(S) = W_{t_2}(P_1) \cup P_2,$
$W_{t_2}(P_1) \cap P_2 = \emptyset,\quad |P_2| = |Q_2| = m_2,$

then from (5) with given (l) we get:

$$N_{l_2}(L_\chi) \le N_{l_2}(S) < C_n^{l_2}\,, \tag{6}$$

Then it follows from (6) that set Q_2 can be also produced.

Arguing by analogy with $r = 2, 3, \dots, k-1$ we will get sets $Q_2, \dots, Q_k$, which owing to the equality $L_\chi = \bigcup_{j=1}^{k} Q_j$? means the existence of the compacted set L_χ with c.s. χ. Along with it we derived that

$$N_i\big(L_\chi\big) \le N_i(S), \tag{7}$$

at $l_k \le i \le n$.

As $W_{l_k}\big(L_\chi\big) = \{L_n^{l_k}(1), \dots, L_n^{l_k}\big(t_{l_k}\big)\}$, lemma(29),

$W_{l_k+j}(L_\chi) = \{W_{l_k+j} W_{l_k}(L_\chi))$ and $W_{l_k+j}(S) = \{W_{l_k+j} W_{l_k}(S))$

at $j = 0, 1, \dots, n - l_k$, then from lemma 19* with given (7), we obtain that

$$N_i(L_\chi) \le N_i(S) \tag{8}$$

at $0 \le i \le l_k$.

Inequalities (7) and (8) prove the second part of the statement of theorem 1.

To accomplish the proof of the theorem it is necessary to note that in case the compacted set L_χ with the given c.s. χ can be produced the class $K_\chi \ne \emptyset$, since $L_\chi \in K_\chi$. Theorem 1 is completely proved.

Corollary 2. For an arbitrary c.s. χ, $\chi = (n, l_1, \dots, l_k,\ m_1, \dots, m_k)$, *if* K_χ *is a non − empty class* s. p. i. b. *with c.s.* χ *and* L_χ *is a compacted set with c.s.* . χ then:

$$N_t(L_\chi) = \min_{S \in K_\chi}\left(\sum_{t=1}^{m}(-1)^{t-1}\sum_{\{S_{p_1},\dots,S_{p_t}\}\subseteq S} C^l{}_{|S_{p_1}\cap S_{p_2}\cap\dots\cap S_{p_t}|}\right).$$

where $0 \le i \le n,\ \ m = \sum_{j=1}^{k} m_j$.

where $0 \le i \le n,\ \ m = \sum_{j=1}^{k} m_j$.?

P r o o f.

Let's prove that:

$$N_t(S) = \sum_{t=1}^{m}(-1)^{t-1}\sum_{\{S_{p_1},\dots,S_{p_t}\}\subseteq S} C^l\,\left|S_{p_1}\cap S_{p_2}\cap\dots\cap S_{p_t}\right|,$$

where we set C_r^i at $i > r$.

We use the principle of «inclusion-exclusion», which is stated in the following way [11]:

If from N objects $N(a_1)$ have the property $a_1, N(a_2)$ – property $a_2, \dots, N(a_r)$ – property a_r, $N(a_1, a_2)$ have both property $a_1$and property $a_2, \dots, N(a_{r-1}, a_r)$ – properties a_{r-1} and $a_r, \dots, N(a_1, \dots, a_r)$ have properties $a_1, \dots, a_r$, then the number of objects $N(a_1', a_2', \dots, a_r')$, that possess none of the properties $a_1, \dots, a_r$, is found by the formula:

$$N(a_1', a_2', \dots, a_r') = N - \sum_{p_1 \epsilon \{1,\dots,r\}} N(a_{p_1}) +$$

$$+ \sum_{\substack{p_1 \neq p_2 \\ p_1, p_2 \epsilon \{1,\dots,r\}}} N(a_{p_1}, a_{p_2}) - \dots + (-1)' N(a_1, \dots, a_r).$$

In this case $N = N_t\ (S)$, i.e. objects of our study are all different bundles of the length i, derived from bundles of the set S.

The denotationa_r, $r = 1, \dots, m$ in this case is interpreted as a sub-bundle of the bundle S_r, while a_r^1 is meant as the absence of the property.

Thus $N(a_1', \dots, a_m') = 0$, as each bundle of the length i under consideration is derived from some bundle $S_r \in S$ and hence possesses at least the property a_r.

Given that $N(a_{p_1}, \dots, a_{p_t})C^l{}_{|S_{p_1}\cap S_{p_2}\cap\dots\cap S_{p_t}|}$, $t = 1, \dots, m$, we get the equality (1)

formula of "inclusion-exclusion", from by virtue of the theorem 1 we obtain the required assertion.

In conclusion we note that theorem 1 and corollary 2, apart from application in the theory of tests and independent interest in combinatorics can be useful in the theory of control systems, theory of coding, etc.

Է. Մ. ՊՈՂՈՍՅԱՆ

ՆՎԱԶԱԳՈՒՅՆ ԹՎՈՎ ԵՆԹԱԲԱԶՄՈՒԹՅՈՒՆՆԵՐ ՈՒՆԵՑՈՂ ԱՆՀԱՄԵՄԱՏԵԼԻ ԲԱԶՄՈՒԹՅՈՒՆՆԵՐԻ ՀԱՄԱԿԱՐԳԵՐ

Դիտարկված է վերջավոր բազմությունների համակարգերի հատկությունների ուսումնասիրման կոմբինատոր ապարատ, որի օգնությամբ լուծվում է վերնագրում նշված համակարգերի կառուցման էքստրեմալ կոմբինատոր պրոբլեմը: Նախկինում այդ ապարատը կիրառվել է նվազագույն թվով ենթաբազմություններ ունեցող համեմատելի բազմությունների համակարգերի կառուցման համար: Վերջինս հիմք հանդիսացավ նկարագրությունների վերծանման պրոբլեմների էնտրոպիկ բարդության ներքին գնահատականների կառուցման համար:

References

1. Macaulay F. S. Some properties enumeration in the theory of modular systems, Proc. London Math. Soc.", v. 26, 531-555, 1927.
2. Clements G.F., Lindstro¨m B. A generalization of a Combinatorial theorem of Macaulay. J. Comb. Theory, v.7, 230-238, 1969.
3. Pogossian E. M. On lower bounds of minimal test lengths, Reports of National Academy of Sciences of RA, N 2, 1970. in Russian.
4. Pogossian E. M. Adaptation of Combinatorial Algorithms. Publishing House of National Academy of Sciences of RA,Yerevan, p. 286, 1983.
5. Korshunov A.V. On the length of minimal tests for rectangular tables, "Kybernetika". N 6, 1970. in Russian.
6. Andrejev A. E., On deadlock and minimal tests., Reports of Academy of Sciences, USSR, v. 256, N3, 1981. in Russian.
7. Yaglom A. M., Yaglom E.M. Non elementary tasks in elementary presentation., Moscow, "Gostekgizdat" ("State technical Publication House"), 1954.
8. Vilenkin N.Y., Combinatorics, Moscow, "Nauka", 1969. in Russian.
9. Sarkissyan O.A. Generalized -graphs. "Mathematical Problems of Cybernetics and Computer Science", Yerevan, 176-187, 1975. in Russian.
10. Pogossian E. M. On one combinatorial problem and its applications in the theory of tests., Dissertation for the Degree of Candidate of Physical-Mathematical Sciences, Moscow, 1970. in Russian.
11. Riordan J., Introduction to Combinatorial Analysis., M., Publishing House for Foreign Literature.,1963, in Russian.

INDECIES

Indices refer to paragraphs-chapters/parts **of the key words and abbreviations of Parts 1,2 (***communities* 4-2/2 means you can address *communities* at the 4th paragraph of Ch. 2, of Part 2)

Իմացությամբ գոյատևման /լինելիության մոդելների կառուցում

Էդվարդ Պողոսյան
Իմացական ալգորիթմներ և մոդելներ ուղղություն
ՀՀ ԳԱԱ ԻԱՊԻ
epogossi@aua.am

Ամփոփում

1. Հետևելով Ժան Պիաժեին, մենք ներակայցնում ենք իմացությունը/cognizing որպես մտավոր համակարգեր սովորելու և կազմակերպելու մտավոր գործնեություն, միաժամանակ շեշտելով, որ մտավոր համակարգերի սովորելը և կազմակերպումը հիմնականում բերվում են նրանց համայնքներում կուտակվածների ձեռքբերմանը։
Այնուհետև հիմք ընդունելով (grounding) Պիաժեի հիմնարար վարկածը, ըստ որի իմացական գործողությունները սովորում են քայլ առ քայլ որոշակի արմատային գործողություններից մինչև բարձրագույնները ընդամենը մի քանի կանոնների միջոցով, մենք ներայացնում ենք Պիաժեի վարկածը հիմնավորող որոշակի մոդելներ և նախադրյալներ։
2. Ենթադրելով, որ իմացությունը մտավոր համակարգերի (mss) մի որոշակի գործողություն է, ալգորիթմները հաշվարկելիության կառուցողական mss մոդելներ են, իսկ օբյեկտ-կողմնորոշված լեզուրները (OOP) կառուցողական շերտ են ալգորիթմների վրա, մենք մշակում ենք OOP-ի կառուցողական ընդլայնում՝ ***մենթալներ*** (mentals)՝ գոյերը ներկայացնելու բնական լեզուների արտահայտչականության հզորությանը մոտենալու համար։
2.1.Մենթալների ադեկվատությունը իմացության մոդելավորման ալգորիթմները հաստատվում են մի քանի չափանիշներով, ներառյալ՝ ալգորիթմների համար Ալոնզո Չարչի եւ Ալան Թյուրինգի չափանիշի միջոցով, մենթալների և արհեստական նեյրոնային ցանցերի դասակարգման միասնականության միջոցով, ինչպես նաև մարդու եւ մենթալների իմացական գործունեության կատարողականությունը փորձագիտական համեմատման միջոցով՝ կոմբինատոր խնդիրների RGT դասի շրջանակներում, որը մոդելավորում է մարդու բնության մեջ լինելու տարատեսակ խնդիրներ։
2.2.Այնուհետև հիմք ընդունելով (grounding) Պիաժեի հիմնարար վարկածը, ըստ որի իմացական գործողությունները սովորում են քայլ առ քայլ որոշակի արմատային գործողություններից մինչև բարձրագույնները ընդամենը մի քանի կանոնների միջոցով, մենք պնդում ենք, որ

- մենթալները, ներառյալ դրանց իմացական զարգացման կանոնները, բերվում են որոշակի արմատների, ներառյալ՝ 1-, 2- տեղանի դասակարգիչների։
- դիտարկված է 1-, 2- տեղանի դասակարգիչների շղթաների զարգացումը տարատեսակ իմացական մենթալներին (cognitive mentals), ներառյալ 1-, 2- տեղանի դասակարգիչների, դրանց հավաքագրմանը տարբեր mss-ներում, և մշակումը մի քանի իմացական գործողությունների համար։

3. Վերը նշվածը թույլ է տալիս ենթադրել, որ՝

- 1-, 2- տեղանի դասակարգիչները քիչ ինքնակազմակերպվող կանոններով, կազմված միևնույն դասակարգիչներից, կարող են զարգացնել իրենց ֆիզիկայի շրջանակներում ընդունված բարձրագույն մարդկային իմացիչների, այդպիսով դառնալով ի վիճակի վերարտադրել իրենց, այնպես ինչպես այսօրվա ԱԲ ջանքերն են ռոբոտաշինության մեջ։

3.1. Այնուհետև հարցադրել՝

- արդյոք 1-, 2- տեղանի դասակարգիչները կարող են առաջանալ բնության մեջ, և հետևելով վերևում բերված ենթադրությանը, զարգանալ դեպի բարձրագույն մակարդակի իմացիչների՝ թույլ տալով կառուցել իրենց իսկ ԱԲ ռոբոտները։

Նշենք, որ այս հարցի դրական պատասխանը կարող է հանգուցալուծել բջջային գոյերի առաջացման առեղծվածը, քանի որ դրանց ահռելի բարդությունը չի կարող ձեռք բերվել պատահականությամբ, այլ կարող է միայն հասցվել կառուցման միջոցով ։

3.2. Ենթադրվում է, որ հարցի դրական պատասխանը հիմված է հետևյալ նախադրյալների վրա։

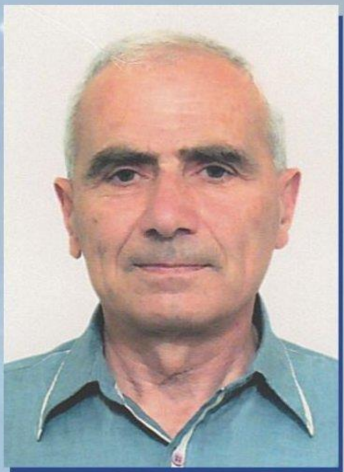
Առաջինը՝ հիշենք, որ նեգենտրոպիկությունը ըստ Շրյոդինգերի և ինֆորմացիան ըստ Շենոնի ունեն նույն չափողականությունը, մինչդեռ ֆիզիկոսները, հղվելով Պարոնդոյի վրա՝ հասնում են ինֆորմացիայի, այդպիսով նաև նեգենտրոպիկների բնության մեջ առաջացման ապացույցին։

Այսպիսով, ենթադրելով, որ ինֆորմացիան և դասակարգումը միմյանցից անբաժան են, կարելի է պնդել, որ նեգենտրոպիկության տարբեր ձևերի, ինչպես նաև 1-, 2- տեղանի դասակարգիչների առաջացումը տիեզերքում, այնուհետև դրանց զարգացումը բարձրագույն իմացիչների եւ ինքնավերարտադրությունը տարբեր ձևերով, սկզբունքորեն հնարավոր է։

7. Հետևաբար, հարց է առաջանում, արդյո՞ք այս առաջացումը միակն է մեր արեգակնային համակարգում, թե այդպիսիք բազմազան են։

Անթրոպիկ սկզբունքին/ **anthropic principle** հետևելով, մեր արեգակնային համակարգին նման պայմանները տիեզերքում բազմազան են։

Առանցկային բառեր: Piaget, իմացություն/cognition, adequate, կառուցողական /constructive, մոդելավորում, դասակարգիչներ, նեյրոնային ցանցեր, ոչ էնտթրոպիկ/ negentropics, տեղեկատվություն/ information, cellular/ բճիճային, առաջացում/ origination, anthropic principle.



Edward M. Pogossian

Head of Direction, Cognitive Algorithms and Models, Institute for Informatics and Automation Problems, Academy of Sciences of Armenia (1973-present).
E-mail: epogossi@aua.am

Education: Doctoral degree in Mathematical Cybernetics, Moscow, Computing Centre of the USSR Academy of Sciences, 1985; Thesis: Adaptation of Combinatorial Algorithms of Classification and Control.
PhD in Mathematical Cybernetics, Moscow, Computing Centre of the USSR Academy of Sciences, 1970; Thesis: Systems of Non-comparable Sets with Minimum Number of Subsets and their Applications to Testing (complete solution of Shroder's problem).
M.S. in Computer Science, Saint Petersburg Electrotechnical University, 1966; **Thesis**: Learning in 4 Layers Neuron Networks.

Teaching and PhD/MSc supervision in the field of Artificial Intelligence at leading universities of Armenia: Yerevan State, Engineering, American, Russian (1970-2014); from 2014, teaching and supervising at the Academy of Sciences of Armenia.
Taught modeling of Strategy Management and Marketing at American University of Armenia (1994-98).

Research. Constructing models of cognizing and origination of cognizing examined on combinatorial games.
It builds on the following key assumptions and research findings:
1. Cognitive systems and means of their construction are various compositions of basic 1-/2- place classifiers. Only a few means are sufficient to realize those constructions and compositions.
2. The highest cognitive power of humans brings them close to the constructive modeling of self-reproduction of their own cognizing, both based and not based on the models of biological cells.
3. Information can originate in Nature.
4. Information and classifying are inseparable from each other.

ISBN 978-5-8080-1437-4

9 785808 014374