Mathematical Problems of Computer Science 62, 25–42, 2024.
doi:10.51408/1963-0118

UDC 004.81

Adequacy and Application of Models of Cognizing by Combinatorial Games

Sedrak V. Grigoryan and Zaven H. Naghashyan

Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
e-mail: sedrak.grigoryan@iiap.sci.am, znaghash@gmail.com

Abstract

We aim to provide constructive adequate models of human cognizing of the Universe. Arguing that combinatorial games are adequate models for studying human-universe problem, we introduce a class of Reproducible Game Trees (RGT) combinatorial games, generally, not limited in the representation of competitive, defense and communication problems.

We develop expert knowledge-based RGT Solver for unified searching of plausible RGT strategies arguing that such strategies are transferable to the entire RGT class.

We estimate the adequacy of models of cognizing, particularly, by progressing in solving RGT problems, which simultaneously provide solutions for urgent applications.

In this work, we outline our RGT approach to arguing the adequacy of cognizing models to the human one and bring together successful applications induced by such arguing.

Keywords: Modeling cognizing, Combinatorial games, Intrusion protection, Defense strategies, Marketing, Learning, Meaning processing.

Article info: Received 6 October 2024; sent for review 15 October 2024; accepted 26 November 2024.

1. Introduction

Following the founders of computer science [1, 2], we interpret Artificial Intelligence as a branch of science aimed to provide adequate constructive models of cognizing, at least, comparable by effectiveness with those of humans.

Our model of cognizing roots in developmental psychology by Jean Piaget [3], follows researchers in modeling cognizing by solvers of combinatorial games [4, 5, 6], enriches object-oriented representatives of realities by input classifiers and relationships in English, while tends to be consistent with questioning the origination of cognizing in nature [7].

Letś outline our approach to arguing the adequacy of models of cognizing to human ones, followed by comprising our successful applications of the models induced by arguing this adequacy.

1.1. We examine the adequacy of our models following Churchś idea, that the hypothesis on the adequacy of models is examined empirically until they are refitted by some interpretations of models or other alternative models not equal to the original ones [8].

We overcome the barrier of studying the incredibly complex Human-Universe problem by approximating it with game models [6, 7]. We assume that

- Combinatorial games with known hierarchies of utilities and solutions in spaces of possible strategies in game trees can represent the Human-Universe (HU) problem with proper adequacy.
- HU is a contemplation of problems, where the unsolved ones appear to be identified as combinatorial ones.
- Human cognizers are positioned as universal means of solving new problems appearing, as a rule, in combinatorial modes.

Then, we narrow HU to the Solvers of problems represented as Reproducible Game Trees (RGT) with only a few 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.

We argued [7, 9] that RGT problems and RGT Solvers are constructively regularized, are models of HU and human cognizers, correspondingly, moreover, computer models of RGT Solvers can be developed to become their adequate models.

We also argue that

- RGT, first of all, embrace combinatorial problems, have no visible limits on their enrichment up to ones of HU.
- RGT Solvers demonstrate an ability to successfully involve models of any means of cognizing of human cognizers to resolve RGT.
- RGT problems are reducible to each other, particularly, to some standard kernel RGT problem K, thus, we get an opportunity to integrate the best-known achievements in solving particular RGT problems into the entire RGT class, i.e., letting us apply those achievements to any of RGT one, as well as experimenting with particular RGT problems [10, 11].

1.2. RGT studying in a variety of modes have successfully been started since 1957 at the Institute for Informatics and Automation Problems of the Academy of Sciences of the Republic of Armenia

In what follows we bring together the descriptions of some successful applications of our models of cognizing induced by arguing their adequacy by RGT games and mainly attained since 2003 [7].

The applications include urgent problems of combinatorial nature in network protection from various types of intrusions by hackers, problems of decision-making in battlefields, marketing (oligopoly competitions) and management (supply chain management), chess and chess-like problems, etc., that can be reduced to RGT ones [11, 12, 13, 14, 15, 16, 17, 18, 19, 20].

2. Problems of Computer Network Protection Against Various Attacks

The game tree model for Intrusion Protection, in brief, is presented as a game between two playing in turn sides with opposite interests - the attacker (A) and the defender (D) playing in turn. The game is described by a set of states and a collection of conversion procedures from one position to another. The main goals of the attackers and defenders are to bring the system in critical states and avoid them, correspondingly. The counteraction game model is represented by AND/OR game. At first, the attacker moves from the initial state s0 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 [13, 21].

IGAF1 and IGAF2 algorithms were proposed, which, based on common knowledge planning and dynamic testing provided decision making for defender against attackers, while expert knowledge utilized in the decisions was acquired in the form of goals and rules.

The decision-making algorithms consisted of the following general steps:

1. Standard min-max technique with alpha-beta pruning was used, based on the range of critical/normal state values introduced as the goal 1. A 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.

2. Determines all suspicious resources.

3. Builds the game subtree for suspicious resources starting from the root state of the tree and using certain rules that determine the trajectories of attacks.

4. Calculates the values of the terminal states of the tree, finds the values of others by min-max procedure and determines the best min-max action from the root state.

5. Determines the trajectories of attacks induced by the best action from the root of the tree to its critical states and considers them as targets.

6. Builds the zones of counteractions for the target trajectories using the cetrain group of rules, then calculates the values of the states of the corresponding subtree using the minmax.

7. Chooses the defenders 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. Ends the defense analysis and waits for the attackers actions.

The provided approach concluded with the following remarkable results:

The viability of the IGAF algorithm was successfully tested in the network intrusion protection problems against the 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 IGAF and min-max algorithms are compatible.

- The number of nodes searched by the IGAF algorithm with all expert rules and subgoals is decreasing compared with other algorithms and the minimax one.

- The number of nodes and the time searched by the IGAF algorithm with all expert rules and sub-goals is the smallest compared with the number of nodes generated by existing at the time known solvers of the same problem.
- IGAF algorithms with all expert rules and sub-goals, for the depth of search 5 and 200 defending steps are overperforming the Productivity of min-max algorithm by 14 percent, using for that 6 times less computing time and searching 27 times less nodes of the tree.

3. Problems of Ship Defense Against Air Threats

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 ship. 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., missiles 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:

- All threat players are created at the start of the scenario.
- The maximum number of threat players is max threats N = 8.
- 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 own ship, - Polar angle between 0o and 90o (i.e, angle in the vertical plane), - Any azimuthal angle (i.e., angle in the horizontal plane).

Assumptions:

1. 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.
2. It is assumed that the defense player may bundle up concurrent defense actions. The admissible bundles must satisfy the engagement rules.
3. 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.

The objective of a defense strategy is to prescribe a unique defense bundle for every admissible threat action.

The game tree describes all the admissible sequence of threat and defense responses. The branching in the game tree is generated by the capabilities of the players and the uncertainties in the game, i.e., branches are created for:

- each category of threat and defense players,

- each admissible defense and threat bundle of actions,
- each possible outcome of the transformation rules (when the outcome of a response is uncertain).

With some certain steps applied to defensive actions, the following types of defense actions were revealed:

- launch a long-range surface-air missile,
- shoot the medium-range gun,
- shoot the short-range gun

The proposed scenario for the defense of the ship leads to the following main conclusions for this problem in [15].

For the case when waiting conditions are ignored search time in the NGT for the best strategies is significantly less compared with those for the TABU system [22].

It is shown that the solution of the game tree for deliberative planning maximizes the probability of survival of the own ship with concerning worst-case situation.

Scenarios with up to 8 threats are considered. Monte Carlo simulations are employed to statistically assess the benefits of the proposed deliberative planning.

4. Graphical Language Interpreter for RGT Problems

Human-computer interaction (HCI) has been a pivotal aspect of computer science since its inception, evolving significantly over the decades to accommodate more intuitive and efficient modes of interaction between humans and machines. Initially, computers operated strictly on machine code, a form of communication that was cumbersome and unintuitive for human operators. This barrier was partially alleviated by the introduction of assemblers, which translated more accessible, albeit still technical, instructions into machine code. The development of procedural languages further shifted HCI towards a more human-centric approach, adopting notations akin to those used in mathematics and physics to facilitate a clearer expression of ideas.

The advent of object-oriented programming marked a significant milestone, establishing a new industrial standard that mirrored human thought processes more closely. By organizing software design around concepts, their relationships, and interactions, object-oriented languages enabled programmers to model complex systems in a manner more aligned with human reasoning. The next frontier in HCI aimed to bridge the gap between human and computer communication through natural language processing, epitomized by developments such as ChatGPT [23]. This AI-driven approach has garnered popularity for its ability to understand and generate human-like text, making interactions more seamless and intuitive.

Despite these advancements, the research underscores a crucial aspect of human cognition: visual processing. Humans predominantly interpret and assimilate information through visual means, not just verbal or textual interactions. Consequently, the research emphasizes the importance of visual languages in HCI, advocating for enhanced visual representations of concepts to align more closely with human cognitive processes. This visual approach not only complements the textual and verbal advancements but also enriches the interaction, making it more holistic and reflective of human informational intake and processing.

To combine the existing efforts in the direction and enable utilizing the achievements in a generic Solver of RGT problems, a common graphical interface was developed [24].

By analyzing about 100 chess concepts, about 4 general construction blocks were identified using which it is possible to represent communicable concepts. They are classifiers (primitive, composite), relations, sets and actions.

Classifiers - there can be 2 types of classifiers - primitive and composite.

- Primitive classifiers - extend the initial classifiers by narrowing the range that can be matched.
- Composite classifiers - compose primitive any types of classifiers, sets and actions.
- Relations - units of knowledge that represent general relations between instances of classifiers.
- Sets - units of knowledge that are used for matching several instances of classifiers.
- Actions - units of knowledge that describe transformations of situations and their components
- instances of concepts.

1. The Knowledge Base module of the RGT Solver package is developed that allows representing, modifying and removing classifiers.
2. A graphical language for the graphical representation of construction blocks and their compositions is developed.
3. A graphical user interface, which provides graphical tools for representing the graphical language units is developed.
4. The viability of the approach and program was proved by experimenting with the representation of about 100 chess concepts and by representing chess concepts for the solution plan of Reti Etude. The method can be divided into the following steps:

- The expert analyzes the new concept and finds out the lower-level concepts and relational rules that are enough to represent the original concept.
- As the next step, the expert checks whether the lower-level concepts already exist in the system. If some lower-level concepts are not represented in the system, the expert tries to represent the corresponding concept by decomposing it into more primitive ones. The process continues until all the lower-level concepts are represented.

The framework for representing communicable lexical units for expert strategies of the RGT class was developed:

- For known types of personalized expert strategies of the RGT class the framework for representing and acquisition is developed.
- OOP constructions [25] and software for representing the variety of strategy knowledge of the RGT problems, interaction and modification are developed and realized. The graphical language interpreter and interface for the formation and acquisition of expert strategies in a regular way is designed and realized in extended Java language.

- The viability of the interpreter, interface and substantiation for the RGT problems is experimentally approbated for expert knowledge intensively using chess problems.

5. Knowledge Presentation, Acquisition and Matching Algorithms

5.1. In [10], the role and proportion of personalized expertise in comparison to common, communicable expertise were examined using chess, a typical representative of the RGT class. Expert requests to the game storage in natural language were simulated and analyzed to identify winning strategies of specified types. A correspondence was revealed between approximately 300 units of chess vocabulary and the Zermelo classes of chess positions and strategies, which argued for the constructive nature of these vocabulary units, thereby, in principle, allowing for their simulation. However, it was also stated that any real implementation of such content could only approximate the original winning game tree structures, due to the prohibitive computational complexity required to prove the correctness of the majority of the content. This provided a measurable precedent of human expertise specification, where the learned contents of realities that share the same vocabulary are, as a rule, essentially personalized.

5.2. Following successful results in developing generic graphical language and unified mechanisms for RGT problem knowledge interpretation and interface, studies were conducted to provide enhanced and generic knowledge presentation and storage mechanisms, along with matching algorithms. All these elements were designed to be adequate those of experts [18, 19].

A model has been built for the representation of classifiers analogical to Be- Have- Do dimensions of English grammar. Models of knowledge in RGT Solvers can be presented as a network of classifiers. It continuously acquires the defined classifiers into the internal network - Network of classifiers (NCl) by finding and constructing appropriate be-, have- or do connections with already existing nodes of Network.

The acquisition of knowledge leads to having different types of NC nodes:

NT - Nucleus Type, the types of the smallest representation units of knowledge. They compose the set of NA roots. The construction of more complex classifiers is started with them.

NC - Nucleus classifiers, the instances of Nucleus Types with additional restrictions (they represent the subset of the value of the types).

CR1 - Classifiers that hold a level between the nucleus and composite classifiers. In other words, a CR1 represents the description of a position in a space.

CC - Composite Classifiers, the most common form of the classifier representation. They are composed of any other Classifier types and are able to represent all kinds of regularities.

SC - Sets are composed of a single composite classifier element and a rule specifying the number of elements in the group.

Action - The following nodes are the only ones that represent actions in the NCl.

VC - Virtual Classifiers, which have undefined values for their attributes (in parallel with OOP virtual classes). Consequently, they inherit almost all the advantages of OOP interfaces (polymorphism, etc.).

The model for presenting and storing (store of prints) situations has been developed as a list of groups of contiguous nucleus attributes, referred to as CR1 instances. The store of prints has been modeled as a composition of t-prints and delta-prints. The former is a

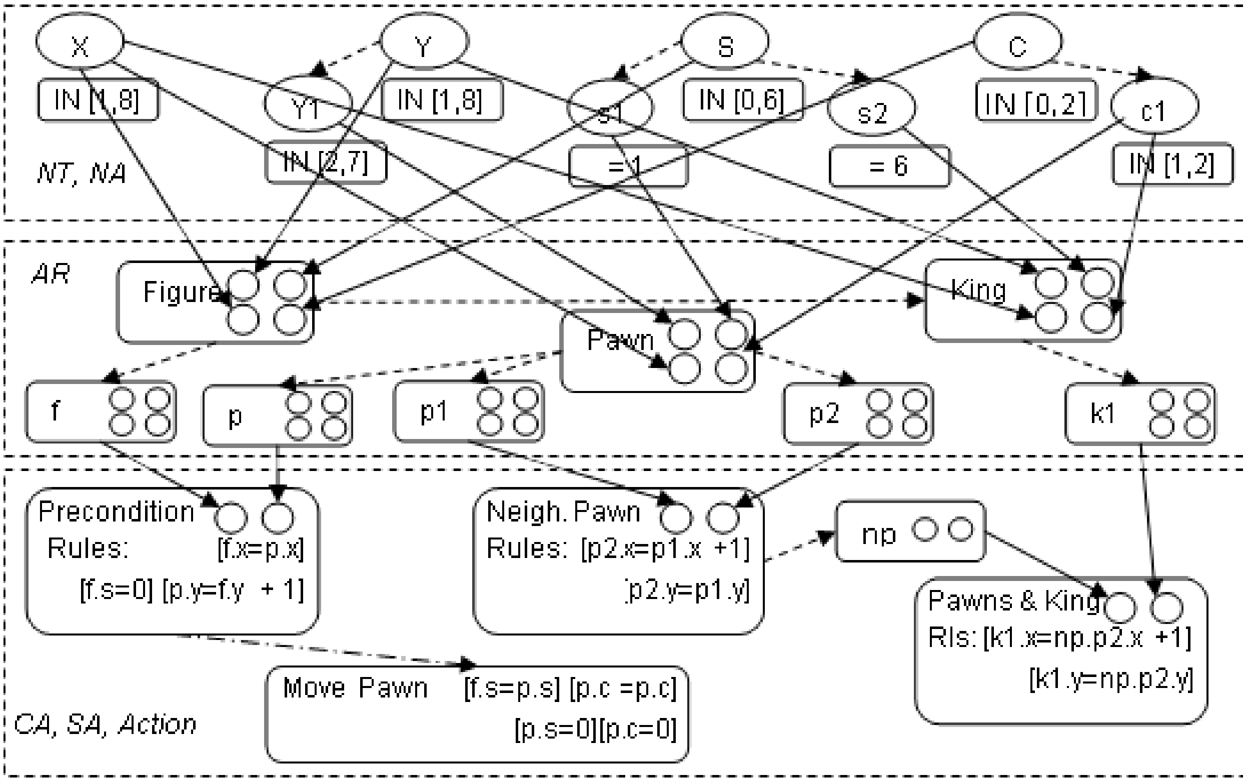

Figure 1: A fragment of a Network of Classifiers for the example of chess representing Be (dense), Have (dashed) and Do (dot-dashed) connections

snapshot of the perceived universe at a time t and is represented as a bunch of cr1 instances, while the latter is the difference between prints at time t+1 and t.

Algorithms have been developed in the generic RGT Solver, which triggers the matching of classifiers to situations by iterative matching of sub-classifiers via processing t-prints constituents over the network of classifiers. It is an adaptation of a classical CSP (constraint satisfaction problem) solving the Rete algorithm [26]. Distinct matching procedures have been developed for different types of nodes categorizing them into filtering and conjunction nodes. 5.3. To transform natural presentations of situations to RGT Solvers symbolic presentation, methods were proposed [27], using Neural Networks for classification and detection [28] of units, applicable to RGT problems, particularly to chess [27] and battlefield [29] scenarios. The chess scenario provides algorithms based on the classifiication and detection of chess units from the given images and the construction of the chess situations, accordingly. The battlefield scenario extends and continues the approach developed in the chess scenario, providing processing and classification of military units on UAV-based images.

6. Decision Making by Personalized Planning and Integrated Testing

Following the approach described in [30, 14] as well as in the above sections, algorithms for decision-making were developed based on personalized planning and integrated testing (PPIT), which enhanced the proposed algorithms making them generic and integrating effective knowledge-based goals searching algorithms of trajectories-zones technique (TZT) as well as initially proposed by Botvinnik [30]. Decision-making is interpreted as strategy searching and proposing actions for the given RGT situations [16, 31].

Thus, in PPIT decision making algorithms, we deal with plans, which are lists of goals by their priorities, while goals consist of a) precondition describing the situations where the goal can be applied; b) postcondition describing the final situation where the goal is achieved; c) depth describing the depth of the tree to be generated for processing the goal; d) criteria describing the situation after of postcondition to show how good the goal is achieved. PPIT algorithms consist of the following mean steps: Reducing Hopeless Plans (RHP), Choosing Plans with Max Utility (CPMU), Generating Moves by a Plan (GMP).

Action selection with TZT algorithms for a given goal is described in [32] which considers the trajectories of attack and zones of counteractions: a. Generation of a tree of situations with the defined depth that leads to goal achievement, b. Extension of situation chains by all possible counteractions by the opponent, and possible actions to intercept the counteractions, c. Checking if the goal can be achieved and evaluation if defined.

The adequacy of PPIT algorithms was demonstrated on chess etudes suggested by Botvinnik as tests for such decision-making frameworks as well as in other RGT problems, e.g., management, battlefield, testing of programs [32, 33, 41].

The experiments effectively cover the adequacy and effectiveness of the provided package modules described in Chapters 4 and 5, wrapping the main components of the system for generic RGT Solvers.

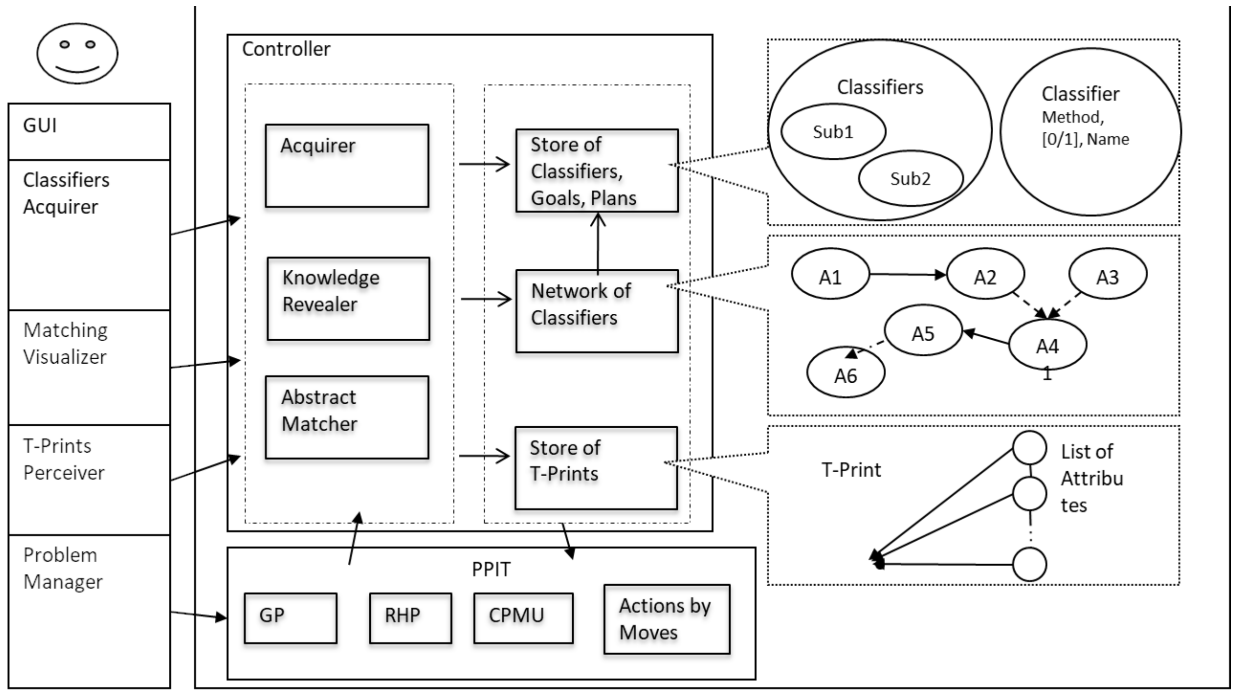

Figure 2: Generic structure of RGT Solvers

7. Testing Programs

The problem of testing programs was reduced to RGT class as follows [33]:

- The actors in software testing are the tested program and the tester.

- The actions are any valid elementary operations that can be performed with the program. While building the game tree, the Solver dynamically combines these actions, creates test cases and executes them depending on the response of the program.
- The situations are the current states of the program. We can estimate the current situations with [0;1] numbers, where 0 means that no bugs were found, 1- that the program is in a critical state and is not usable. The numbers in between 0 and 1 are intermediate values and situations with values closer to 1 are worse than situations with values closer to 0.

In the scope of the research, the following results were achieved:

1. Tools defining types of knowledge testing of the target application were described. The described knowledge was integrated into RGT Solver and used to run test cases and test scenarios with later evaluation of test results.
2. An approach for evaluating the state of the program during the testing was proposed.
3. The adequacy of the proposed approach was experimented with the open-source Blender application.
4. The proposed approach solves the drawbacks of the model-based testing approach [35], namely, allows to generate test cases dynamically.

8. Battlefield Problems

Following achievements in the games of strategy video games and modern UAV-based solutions the research was initiated to transfer battlefield problems into generic RGT Solvers and apply its achievements for decision-making and autonomy integration [29, 41].

The battlefield problems were interpreted as RGT problems as follows:

- The battling sides can be considered as interacting actors
- Military units' movements and attacks can be considered as actions
- The battlefield areas with the military units involved, can be considered as situations
- As goals, different situations can be considered, such as: capturing objects, destroying enemy units, pushing the frontline, holding the defense, etc.

The following results were achieved at the current state of the research:

1. Processing of aerial images to detect 8 military unit classes based on the constructed model. The training dataset of the model represents 8 classes of military unit groups defined by experts.
2. Certain expert-defined classifiers were integrated for proper processing of target selection algorithms.
3. Algorithms were developed to select the target based on input images, objects classified according to them, and the knowledge of the field.

4. Experiments on low-power computing devices demonstrated close to real-time processing efficiency. The solution effectively covers aerial image-based decision-making for a single UAV.

5. Situations for battlefield presentation were provided for RGT Solvers representing a certain composition of battlefield essential nuclear classifier instances.

6. Decision-making algorithms of PPIT and TZT available in RGT Solvers are applied to propose actions for the given situations.

9. Communication and Interaction with RGT Solvers

9.1. A group of urgent tasks in RGT Solvers and in expert systems, in general, relate to interaction and communication, particularly problems of acquisition of knowledge from available sources[36], explanation of acquired knowledge, etc. Studies in this direction were conducted. The research concluded in several directions, particularly: 9.2. Tools for testing and correction of the completeness of knowledge acquisition by autistic children were proposed [37], where the peculiarities of 10 years of successful training of an autistic child were considered and some patterns of positive tutoring of children both autistic and ordinary were extracted possible to realize in RGT Solver. The successive approaches to chess-based tutoring and testing were discussed. The assertions on conditions of positive outcomes of tutoring the communalized and personalized meanings of chess concepts were argued and the ways of their implementation by the Solver were considered. Finally, the perspectives of development of the Solver for tutoring the basics of Math, pre chess, testing and tutoring strategy knowledge were discussed.

9.3. Personalized interactive tutors for chess were developed [38, 39] with the following main results:

- Method and software for tutoring chess were developed in the frame of RGT Solvers, external tools were implemented providing students with the following advantages:
    1. The software provides personalized tutoring mechanisms for different types and levels of students and their performances.
    2. The software is interactive making level-by-level tutoring of chess concepts, their testing, with feedback provision and correction with detailed description.
- The designed algorithm and software are validated by testing them on tutoring for chess endgames.

9.4. Algorithms for automating acquisition and explanation of RGT strategy knowledge were developed [40]:

- Algorithms for extracting RGT knowledge from texts were developed and discussed, which included classifiers learning from strictly formatted texts, particularly virtual classifiers, as well as approaches for overcoming strict format restrictions, were discussed and some solutions were shown.
- Meantime enhancement of acquisition of classifiers by examples, using neural networks is performed, which allows adequate demonstration and learning from images, a way

of natural presentation of classifiers, as well as allows using these classifiers to properly demonstrate examples when explaining them.

- Algorithms for explaining acquired by RGT Solvers knowledge by texts were provided, which, based on the HBD model of knowledge presentation allows providing texts for RGT classifiers using have, be and do dimensions of English.

- Experiments to validate the algorithms were conducted for chess classifiers.

9.5. To advance in meaning processing, particularly, Text-Meaning-Text transformations problems, we aim to advance in Learning Expert Meaning Processing (LEMP) and consider this problem in the scope of the RGT class [41].

At first, we consider the interaction with natural language as usage for expert knowledge, and attempt to advance in LEMP in RGT Problems for a kernel RGT problem, e.g., chess. This task includes tasks, such as Natural Language Processing (NLP) for the entire Natural Language reduced to NLP for urgent essential RGT class, then reduced to a seed of RGT (chess) language processing.

9.5.1. The following subtasks are covered for the kernel problem:

- Preparing a Repository of RGT expert classifiers for the kernel RGT problem and ordering them by the complexity of learning by RGT Solvers.

- Advancing in learning expert classifiers by stages 1, 2, ... of their complexity

- Confirming workability of at the time already RGT Solver learned classifiers, particularly by demonstration of abilities of learning, identification of realities, meaning to text to meaning transition.

- Enhancing the versions to identify more realities and to provide better meaning to text to meaning transition.

9.5.2. Focusing on the chess RGT problem, 5 levels of chess classifiers were revealed (initial level and 1-4 main levels) and successfully acquired by RGT Solvers, particularly 1-3 levels of chess classifiers described, particularly, in [7] were acquired by Solvers and matched to chess situations confirming their workability for 100 various chess classifiers.

The initial level of classifiers represents the basic types similar to OOP built-in types, in this case, we start definition with coord x and coord y, figure type, figure color, then expand to field, figure and specific figure types, such as pawn, knight, etc. Some of the other classifiers acquired by the Solvers are:

1st level: vertical, diagonal, phalax of knight and pawn, etc.

2nd level: field under the attack of a pawn (either black or white), passant pawn, doubled pawn, etc.

3rd level: phalax of pawns in the center of the board, hit or defense by a figure (e.g., by knight), etc.

Results demonstrate the adequacy of chess classifiers acquired by the Solver to expert ones and the above-mentioned solutions can be used to transform to texts for explanation.

The successful results then can be used for:

- Expanding to the whole class of RGT.

- Expanding the results of the first and second stages to the natural language content.

10. Conclusions

10.1. We have observed some successful applications of our models of cognizing by arguing their adequacy through RGT games, with those achievements mostly occurring since 2003. Particularly, they include:

- Intrusion protection problem: an urgent problem in the field of information technologies, was successfully modeled as an RGT problem and the provided Solvers overcame the existing at the time approaches in providing a successful defense of the network again various types of hacker attacks.

- Single ownship defense: the problem was successfully modeled as an RGT problem and Solvers for the task of defending their own ship against specific known types of missile attacks were developed providing evidence of the approach maximizing the probability of the ownship survival.

- Generic Solver was developed that utilizes the following main components: 1. Advanced graphical language interpreter that allows users to insert RGT classifiers with specified types, 2. A network of classifiers was developed to present and store RGT classifiers in the form of a network based on the HBD language model and matching algorithms were developed to match these classifiers to RGT situations, 3. Decision-making algorithms were developed based on PPIT planning and TZT goal-searching algorithms. Interfaces for various RGT problems were developed.

- Testing of programs was successfully interpreted as an RGT problem, where a new approach for evaluation of testing program state was proposed. The adequacy of the approach was experimented on open-source applications and overcame some drawbacks of the popular model-based testing approach.

- Battlefield problems were interpreted as RGT problems, based on aerial images, where certain classes of military units were defined and the approach demonstrated adequacy to the experts of the field providing effective autonomous decision-making based on expert knowledge utilizing RGT Solver capabilities.

- To enhance the experience of communication with RGT Solvers, problems of communication were researched. 1. A personalized interactive tutor for chess was developed with comprehensive tutoring abilities. 2. Algorithms were developed to extract RGT knowledge from texts and vice versa: to convert RGT knowledge to texts. 3. The LEMP problem was considered as an interpretation of the NLP problem and certain levels of kernel RGT problem classifiers were identified and successfully acquired by the solver, while being adequate to human classifiers, allowing them to transform to human-understandable texts and applying explanation algorithms to interact with humans.

- Many more urgent problems were interpreted and successfully researched as RGT problems, such as the detection of anomalies in big data, problems in marketing and management, diagnoses of specific diseases, etc.

10.2. Note, that some other urgent problems were also interpreted as RGT problems and successfully resolved, including detection anomalies in big data, problems of marketing and management, diagnoses of specific diseases, etc. [20, 12, 37].

10.3. Future improvements in general Solvers based on these foundations, expanding their capabilities in learning expert meaning processing can allow them to easily acquire domain expert knowledge, evolve into comprehensive assistants that not only search for optimal strategies combinatorial problems of the class but also interact with users across a wide range of domains of the class. Continuous research and development in this direction is in progress.

References

[1] T. Burge and H. Enderton, *The Collected Works of Alonzo Church*, MIT Press, Cambridge, MA, USA, 2019.

[2] A. Turing, "Computing machinery and intelligence," *Mind*, vol. 49, reprinted in *Minds and Machines*, A. Anderson (ed.), Engelwood Cliffs, NJ, 1950.

[3] J. Flavell, *The Developmental Psychology of Jean Piaget*, D. Van Nostrand Comp. Inc., Princeton, 1962.

[4] R. Benergji, *Theory of Problem Solving*, Elsevier, 1969.

[5] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: an architecture for general intelligence," *Artificial Intelligence*, vol. 33, no. 1, pp. 1–64, 1987.

[6] E. Pogossian, "Adaptation of combinatorial algorithms," *Academy of Sciences of Armenia*, 1983.

[7] E. Pogossian, *Constructing Models of Being by Cognizing*, Academy of Sciences of Armenia, Yerevan, Armenia, 2020.

[8] E. Pogossian, "Artificial intelligence: alternating the highest human cognizing," in *Proceedings of International Conference of Computer Science and Information Technologies 2019*, pp. 36–40, Yerevan, Armenia, 2019.

[9] E. Pogossian, "Specifying adequate models of cognizers," *AIP Conference Proceedings*, vol. 2757, no. 1, article 020003, 2021. doi:10.1063/5.0135909.

[10] E. Pogossian, "Specifying personalized expertise," in *International Conference of Cognition and Exploratory Learning in Digital Age*, pp. 151–159, Barcelona, Spain, 2006.

[11] S. Grigoryan, "On validity of personalized planning and integrated testing algorithms in reproducible games," in *Proceedings of International Conference of Computer Science and Information Technologies 2015*, pp. 317–321, Yerevan, Armenia, 2015.

[12] E. Pogossian, "Focusing management strategy provision simulation," in *Proceedings of International Conference of Computer Science and Information Technologies 2001*, Yerevan, Armenia, 2001.

[13] E. Pogossian, A. Javadyan, and E. Ivanyan, "Effective discovery of intrusion protection strategies," in *Workshop on Agents and Data Mining*, pp. 263–274, St. Petersburg, Russia, 2005.

[14] E. Pogossian, V. Vahradyan, and A. Grigoryan, "On competing agents consistent with expert knowledge," in *Workshop on Autonomous Intelligent Systems - Agents and Data Mining*, St. Petersburg, Russia, 2007.

[15] E. Pogossian, D. Dionne, A. Grigoryan, J. Couture, and E. Shahbazian, "Developing goals directed search models empowering strategies against single ownship air threats," in *Proceedings of International Conference of Computer Science and Information Technologies 2009*, pp. 155–163, Yerevan, Armenia, 2009.

[16] 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, 2015.

[17] K. Khachatryan, "Developing algorithms and programs for formation of and search in knowledge bases of competition and combating problems," PhD Thesis, Yerevan, Armenia, 2013.

[18] K. Khachatryan and S. Grigoryan, "Java programs for presentation and acquisition of meanings in SSRGT games," in *SEUA Annual Conference*, pp. 135–141, Yerevan, Armenia, 2013.

[19] K. Khachatryan and S. Grigoryan, "Java programs for matching situations to the meanings of SSRGT games," in *SEUA Annual Conference*, pp. 127–135, Yerevan, Armenia, 2013.

[20] Z. Naghashyan, "Developing software for formation and acquisition lexical units in certain limited languages," PhD Thesis, Yerevan, Armenia, 2010.

[21] E. Pogossian and A. Djavadyan, "A game model for effective counteraction against computer attacks in intrusion detection systems," in *NATO ASI, Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*, Tsaghkadzor, Armenia, 2003.

[22] D. E. Blodgett, M. Gendreau, F. Guertin, J. Potvin, and R. Sguin, "A Tabu search heuristic for resource management in naval warfare," *Journal of Heuristics*, vol. 9, pp. 145–169, 2003.

[23] P. P. R. Partha, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023.

[24] Z. Naghashyan, "Developing graphical language interpreter for representing communicable knowledge of the concepts," in *SEUA Annual Conference 2010*, Yerevan, Armenia, 2010.

[25] D. Poo, D. Kiong, and S. Ashok, *Object-Oriented Programming and Java*, Springer Science & Business Media, London, 2007.

[26] C. L. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," *Artificial Intelligence*, vol. 19, pp. 17–37, 1982.

[27] N. Hakobyan, "System for transforming images to symbolic presentation for combinatorial defense and competition problems," *NAS RA and ESUA Series of Technical Sciences*, pp. 199–209, Yerevan, Armenia, 2019.

[28] K. He, S. Ren, J. Sun, and X. Zhang, "Deep residual learning for image recognition," *CoRR*, abs/1512.03385, 2016.

[29] S. Grigoryan and E. Pogossian, "Developing aerial unmanned effective decision makers," *Mathematical Problems of Computer Science*, vol. 58, pp. 52–60, 2022.

[30] M. Botvinnik, *Computers in Chess: Solving Inexact Search Problems*, Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag, New York, 1984.

[31] S. Grigoryan, "Research and development of algorithms and programs of knowledge acquisition and their effective application to resistance problems," PhD Thesis, Yerevan, Armenia, 2016.

[32] S. Grigoryan, N. Hakobyan, and T. Baghdasaryan, "Solvers of combinatorial problems adequate to experts," in *Proceedings of International Conference of Computer Science and Information Technologies 2019*, pp. 29–32, Yerevan, Armenia, 2019.

[33] M. Buniatyan, S. Grigoryan, and E. Danielyan, "Expert knowledge-based RGT solvers for software testing," *Mathematical Problems of Computer Science*, vol. 59, pp. 45–56, 2023.

[34] S. Grigoryan, "Context knowledge-based decisions for battlefields," in *Proceedings of International Conference of Computer Science and Information Technologies 2023*, pp. 42–44, Yerevan, Armenia, 2023.

[35] I. Schieferdecker and A. Hoffmann, "Model-based testing," *IEEE Software*, vol. 29, no. 1, pp. 14–18, 2012.

[36] P. Langley, H. Shrobe, and B. Katz, "Cognitive task analysis of rapid procedure acquisition from written instructions," in *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*, 2021.

[37] E. Arakelova and E. Pogossian, "Tools for testing and correction of the completeness of knowledge acquisition by autistic children," in *Proceedings of International Conference of Computer Science and Information Technologies 2011*, pp. 159–165, Yerevan, Armenia, 2011.

[38] S. Grigoryan and L. Berberyan, "Developing interactive personalized tutors in chess," *Mathematical Problems of Computer Science*, vol. 44, pp. 116–132, 2015.

[39] E. Pogossian, S. Grigoryan, and L. Berberyan, "Personalized Interactive Tutoring in Chess," in *London Chess Conference*, December 10–12, 2016.

[40] S. Grigoryan, "Automating acquisition and explanation of strategy knowledge," in *Proceedings of International Conference of Computer Science and Information Technologies 2021*, pp. 21–23, Yerevan, Armenia, 2021.

[41] S. Grigoryan and E. Pogossian, "Learning Expert Meaning Processing," RAU-IIAP Science Comimittee project 2023-2025.

Իմացության մոդելների ադեկվատությունը և կիրառությունները կոմբինատոր խնդիրներում

Սեդրակ Վ. Գրիգորյան և Զավեն Հ. Նաղաշյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
e-mail: sedrak.grigoryan@iiap.sci.am, znaghash@gmail.com

Ամփոփում

Մենք նպատակ ունենք տրամադրել մարդու կողմից տիեզերքի իմացության կառուցողական և ադեկվատ մոդելներ: Արձանագրելով, որ կոմբինատոր խաղերը համարժեք մոդելներ են Մարդ-Տիեզերք խնդրի ուսումնասիրության համար, մենք ներկայացնում ենք Վերարտադրվող Խաղային Ծառերի (RGT) կոմբինատոր խաղերի դաս, որն ընդգրկում է ոչ միայն մրցակցային, պաշտպանական և հաղորդակցական խնդիրներ:

Մենք մշակում ենք փորձագիտական գիտելիքահեն RGT Solver ծրագիր, որը ընդհանուր ձևով փնտրում է հնարավոր RGT ռազմավարություններ՝ հիմնավորելով, որ այդ ռազմավարությունները փոխանցելի են RGT ամբողջ դասի մեջ:

Մենք գնահատում ենք իմացական մոդելների ադեկվատությունը, մասնավորապես, RGT խնդիրների լուծման մեջ առաջընթացի միջոցով, միաժամանակ լուծումներ տալով արդի կիրառական խնդիրներին:

Այս աշխատանքում մենք ուրվագծում ենք մեր RGT մոտեցումը՝ փաստելով մարդու ադեկվատությունը իմացական մոդելների հարցում և ներկայացնում դրանից բխող որոշ հաջող կիրառություններ:

Բանալի բառեր՝ իմացության մոդելավորում, կոմբինատոր խաղեր, պաշտպանություն ներխուժումից, պաշտպանական ռազմավարություններ, մարկետինգ, ուսուցում, իմաստների մշակում:

Адекватность и применение моделей познания с использованием комбинаторных игр

Седрак В. Григорян и Заве О.Нагашян

Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
e-mail: sedrak.grigoryan@iiap.sci.am, znaghash@gmail.com

Аннотация

Мы стремимся предложить конструктивные и адекватные модели человеческого познания Вселенной.

Утверждая, что комбинаторные игры являются адекватными моделями для изучения проблемы человек-Вселенная, мы вводим класс Воспроизводимых Игровых Деревьев (RGT), которые в общем случае не ограничены представлением конкурентных, оборонительных и коммуникационных задач.

Мы разрабатываем основанный на экспертных знаниях RGT Solver для унифицированного поиска возможных стратегий RGT, утверждая, что такие стратегии могут быть перенесены на весь класс.

Мы оцениваем адекватность моделей познания, в частности, посредством прогресса в решении задач RGT, которые одновременно предоставляют решения для актуальных приложений.

В данной работе мы излагаем наш подход к RGT для аргументации адекватности моделей познания к человеческому и приводим успешные приложения.

Ключевые слова: моделирование познания, комбинаторные игры, защита от вторжений, оборонительные стратегии, маркетинг, Обучение, обработка значений..