ISOMORPHISM OF PREDICATE FORMULAS IN ARTIFICIAL INTELLIGENCE PROBLEMS

Tatiana Kosovskaya

Abstract: *The paper discusses various aspects of application of the notion of isomorphism of elementary conjunc- tions of predicate formulas in Artificial Intelligence (AI) problems that can be formalized by means of predicate calculus. The notion of isomorphism of various objects is widespread in mathematics. Moreover, isomorphic objects have a large number of identical properties. The definition of isomorphic elementary conjunctions of predicate formulas is given in the paper. The main property of such formulas is that they define the same relation between their arguments. The main difference between the notion of isomorphism and the notion of equivalence is that the equivalent formulas must have the same arguments, and the arguments of isomorphic formulas may be significantly different. In the framework of the logic-objective approach to solving AI problems, the following problems, for solving which the notion of isomorphism is used, are considered: the problem of object classification; creating a level description of classes to decrease the computational complexity of the analysis problem of a complex object; creating a level description of the database to decrease computational complexity while multiple solution of the problem Conjunctive Boolean Query; definition of a metric in the space of elementary conjunctions of predicate formulas.*

Keywords: *Logic-objective recognition, isomorphism of predicate formulas, NP-completeness, GI-completeness.*

ITHEA Keywords*: G.2. Mathematics of Computing, Discrete mathematics.*

Introduction

When solving an AI problem, much attention is paid to the methods of this problem formalizing. Among such methods, logical methods are widely used. Usually an object is simulated by a string of binary attributes. With this simulation, the structure of an object is lost, but the solving algorithms are usually polynomial or even linear.

An example of economic problem simulation using a binary string is presented in Russel, etc., [2008]. At the same time, the length of such a string is exponential under the length of the initial data presented as a set of properties and relations between elements of an object under consideration.

In the frameworks of a logic-objective approach to AI problems Kosovskaya, [2018], an object under consideration is represented as a set of its elements, and a description of an object is a set of literals (atomic formulas or their negations) with predicates that define properties of an object elements or relations between them. Such a way formulated problems are usually NP-complete or NP-hard (if it is required not only to recognize the presence of a part with specified properties in a complex object, but also to select such a part) Kosovskaya, [2007].

The notion of isomorphism of elementary conjunctions of predicate formulas is under consideration in this paper. Isomorphic formulas define the same relation between their arguments and, therefore, the extraction of sub-formulas isomorphic to each other from a set of more complex formulas allows us to find out the structure of objects. Extracting of such a structure in the input data makes it possible to decrease computational complexity of many AI problems.

The following problems, for solving which the notion of isomorphism is used, are considered: the problem of object classification; creating a level description of classes to decrease the computational complexity of solving the analysis problem of a complex object; creating a level description of the database to decrease computational complexity while multiple solution of the problem Conjunctive Boolean Query; definition of a metric in the space of elementary conjunctions of predicate formulas.

Logic-objective approach to AI problems solving

A detailed description of the logic-objective approach to AI problems is presented in Kosovskaya, [2018]. Here only general setting of the problems and the main methods for their solving will be formulated.

Let an investigated object be represented as a set of its elements $\omega = \{\omega_1, \ldots, \omega_t\}$. A set of predicates $p_1, \ldots, p_n$, which characterize properties of elements of ω and relations between them, is defined on ω. The logical description $S(\omega)$ of an object ω is a set of all literals true on ω. The set of all objects is divided into classes $\Omega = \cup_{k=1}^{K} \Omega_k$. The logical description of the class Ω_k is the formula $A_k(\overline{x}_k)$ given as a disjunction of elementary conjunctions, such that if $A_k(\overline{\omega})$ is true, then $\omega \in \Omega_k$.[1]

The following problems may be formulated in the framework of logic-objective approach.

Identification problem. *To check whether the object ω or its part satisfies the description of the class $A_k(\overline{x}_k)$ and to extract this part of the object.*[2]

$$S(\omega) \Rightarrow \exists \overline{x}_{k\neq}\, A_k(\overline{x}_k) \tag{1}$$

Classification problem. *To find all such numbers k that the formula $A_k(\overline{\omega})$ is true.*

$$S(\omega) \Rightarrow \bigvee_{k=1}^{M} A_k(\overline{\omega}) \tag{2}$$

Analysis problem. *To find and classufy all parts τ of an object ω such that $A_k(\overline{\tau})$ is true for some permutation $\overline{\tau}$ of elements of τ.*

$$S(\omega) \Rightarrow \bigvee_{k=1}^{M} \exists \overline{x}_{k\neq}\, A_k(\overline{x}_k) \tag{3}$$

The problems (1) and (3) are NP-complete ones Kosovskaya, [2007].

[1] Hereinafter, $\overline{x}$ denotes the list of elements of a finite set x corresponding to a certain permutation of its elements. The fact that the elements of the list $\overline{x}$ are elements of the set y will be written in the form $x \subseteq y$.

[2] To denote that values for the list of variables $\overline{x}$, satisfying the formula $A(\overline{x})$ are different, instead of $\exists x_1 \ldots \exists x_m (\&_{i=1}^{m} \&_{j=i+1}^{m} (x_i \neq x_j) \;\&\; A(x_1, \ldots, x_m))$ the notation $\exists \overline{x}_{\neq} A(\overline{x})$ will be used.

Note that to prove (1) or (3), it is sufficient to be able to prove the logical sequent

$$S(\omega) \Rightarrow \exists \overline{x}_{\neq} A(\overline{x}), \quad (4)$$

where $A(\overline{x})$ is an elementary conjunction of atomic formulas or their negations. Estimates of the number of steps of the algorithms solving problem (4), as well as problems (1) and (3), are proved in Kosovskaya, [2007]. These bounds have an exponential on the length of formula $A(\overline{x})$ form.

So, for example, for an algorithm based on exhaustive search method of complete enumeration of all possible substitutions of constants from ω instead of variables from $A(\overline{x})$, the number of steps is $O(t^m \cdot \sum_{i=1}^{n} a_i \cdot s_i)$, where t is the number of constants in ω, n are the numbers of variables in $\overline{x}$, s_i, a_i are the numbers of occurrences of atomic formulas with the predicate p_i in $S(\omega)$ and in $A(\overline{x})$, respectively. Note, that this estimation coincide with the one for simulation of an economic problem by a binary string, given in Russel, etc., [2008].

For algorithms based on the construction of the derivation in the predicate calculus, the estimates are $O(s_1^{a_1} \cdot ... \cdot s_n^{a_n}) = O(s^a)$, where $s = \max_i(s_i)$, $a = \sum_{i=1}^{n} a_i$.

The problem (2) may be reduced to a sequential check for $k = 1, \ldots, M$ of an elementary conjunction $A_k(\overline{\omega})$ and a conjunction of literals from $S(\omega)$ isomorphism.

Isomorphism of predicate formulas

Definition. *Two elementary conjunctions of atomic predicate formulas P and Q are called isomorphic if there exists such an elementary conjunction R and substitutions of arguments $a_1, \ldots, a_m$ and $b_1, \ldots, b_m$ of formulas P and Q, respectively, instead of all occurrences of the variables $x_1, \ldots, x_m$ of the formula R, that the results of these substitutions in R coincide with the formulas P and Q, respectively, up to the order of literals.*

The recieved substitutions $(a_1 \to x_{i_1}, \ldots, a_m \to x_{i_m})$ and $(b_1 \to x_{j_1}, \ldots, b_m \to x_{j_m})$ are called unifiers of the formulas P and Q with formula R.

Note that the arguments of elementary conjunctions P and Q may be either object variables or object constants. In addition, the notion of isomorphism of elementary conjunctions of atomic predicate formulas differs from the notion of these formulas equivalence, since they may have significantly different arguments. In fact, for isomorphic formulas there are such permutations of their arguments, that they define the same relation between their arguments.

An algorithm for checking the isomorphism of formulas, which has an exponential computational complexity, as well as an approximate polynomial algorithm for solving this problem is in Petrov, [2016]. If the approximate algorithm gives the answer No, then the formulas are not isomorphic. An example of formulas that are not isomorphic but the algorithm gives the answer Yes is given. If Yes, the computational experiment showed that in 99.95% of cases the formulas are really isomorphic.

Consider two problems.

Predicate formula isomorphism (PFI).

Instance. Two elementary conjunctions of atomic predicate formulas P and Q.

Question. Are P and Q isomorphic?

Graph isomorphism (GI) Garey, etc., [1979].

Instance. Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.

Question. Are $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ isomorphic?

It is not now known if **GI** belongs to the class **P** nor it is an NP-complete one. A special term GI-complete problem is introduced for problems that are polynomially equivalent to it. Bellow a sketch of the proof that **PFI** and **GI** are polynomially equivalent is presented.

Theorem 1. GI *is polynomially reducible to* **PFI**.

Proof. **GI** is a restriction of **PFI**. The restriction consists in the condition that there is the only one binary predicate p and elementary conjunctions P and Q do not contain negations. In such a case the set of arguments of every elementary conjunction is the set of nodes, and the set of argument pairs in literals is the set of edges. □

To prove the polynomial equivalence of the problems it would be shown that according to every pair of elementary conjunctions P and Q it s possible to construct two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ which are isomorphic if and only if P and Q are isomorphic.

Introduce the notion of **graph associated with an elementary conjunction** of predicate formulas every literal of which does not contain the same arguments.

For each predicate symbol p_i, we associate two nodes w_{2i-1} and w_{2i}. The first one corresponds to the occurrences of literals with p_i without negations, and the other to the occurrences of literals with p_i with negation. In addition, each of these nodes w_j has $j + 2$ adjacent "hanging" nodes (nodes with the degree 1) to indicate the index of a predicate symbol.

For each occurrence of a literal corresponding to the node w_i $(1 \leq i \leq 2n)$, we associate a sub-graph obtained by successively connecting of k_i (the number of predicate arguments) nodes with edges (the first of them mark with the help of a hanging node), as well as with the node w_i. The jth node $j = 1, \ldots, k_i$ is connected with the node corresponding to the jth argument of the literal.

For example, the sub-graph associated with the literal $\neg p_i(a, b, c)$ is presented on the Figure 1.

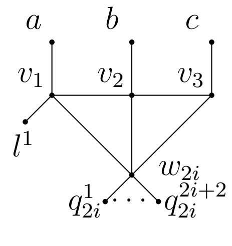

Figure 1. Sub-graph associated with the literal $\neg p_i(a, b, c)$.

All such sub-graphs, associated with different literals with the same predicate, have a common node w_{2i-1} (or w_{2i} if its occurrence is with negation) with $2i + 1$ (or $2i + 2$) adjacent "hanging" nodes. For example, elementary conjunction $p(a)\ \&\ q(x, y)\ \&\ q(x, a)\ \&\ \neg q(a, y)$ is associated with the graph presented on Figure 2. Here the nodes w_1, w_3 and w_4 correspond to literals with p, q and $\neg q$, respectively.

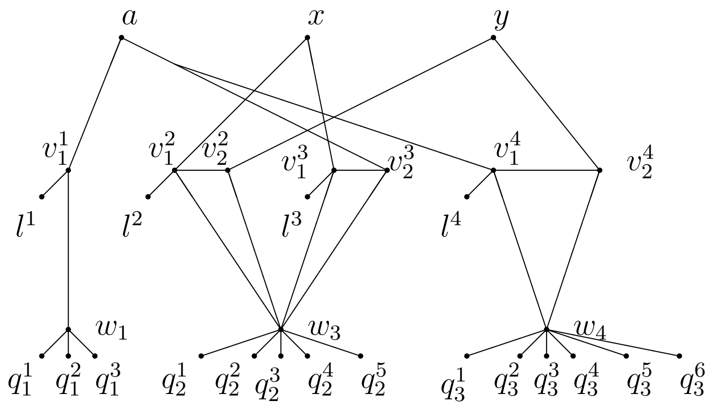

Figure. 2. Graph associated with the elementary conjunction $p(a)\ \&\ q(x,y)\ \&\ q(x,a)\ \&\ \neg q(a,y)$.

Theorem 2. PFI *is polynomially reducible to* **GI***.*

Scheme of proof. According to input data P and Q for the problem of the **PFI** we construct input data G_P and G_Q in the form of graphs associated with P and Q. Graphs G_P and G_Q may be written down in number of steps bounded by a polynomial under the length of P and Q.

Elementary conjunctions P and Q are isomorphic if and only if graphs G_P and G_Q are isomorphic.

For an arbitrary isomorphism of graphs G_P and G_Q, sub-graphs with the same node w_j may not necessarily be identicall. This corresponds to a permutation of literals in elementary conjunctions. □

Polynomial equivalence of the problems of **PFI** and **GI** directly follows from thees theorems.

Theorem 3. PFI *is polynomially equivalent to* **GI***.*

Classification problem

The classification problem (2) formulated in the first section is the one which is the most spread among recognition problems in the frameworks of Artificial Intelligence. While its solving it is not needed to extract the object from a more complicated one. There is no a so called "vicious circle": extract for recognition and recognize to extract.

The solution of this problem is a multiple (for $k = 1, \ldots, M$) logical sequence checking

$$S(\omega) \Rightarrow \exists \Pi(\omega) A_k(\Pi(\omega)),$$

where $\Pi(\omega)$ is a permutation of the elements of the set ω.

That is, if the sign $\&$ is inserted between the literals in the description of the object $S(\omega)$, then the resulting formula $S_f(\omega)$ is isomorphic to one of the disjunctive terms of the class description $A_k(\overline{x}_k)$.

So, the classification problem, in contradistinction to identification problem and analysis problem, is proved to be GI-complete.

Level description of classes

To decrease the number of steps while running an algorithm that solves the described problems, a level description of classes of recognizable objects is proposed in Kosovskaya, [2008], which is essentially a hierarchical description of classes. It is based on the extraction from the class description of sub-formulas that are isomorphic to each other and which define the generalized characteristics of objects of the same class Kosovskaya, [2014].

In particular, this can be done by extracting the formulas $P_i^1(\overline{y})$ ($i = 1, \ldots, n_1$), isomorphic to the "often occurred" sub-formulas of the formulas $A_k(\overline{x}_k)$ with "small complexity". In such a case, a system of equivalences of the form $p_i^1(y_i^1) \leftrightarrow P_i^1(\overline{y}_i)$ is written, where p_i^1 are new predicates, which will be called the first-level predicates, and the variables y_i^1 are new variables for the lists of the initial variables, which will be called the first-level variables.

Denote the formulas obtained from $A_k(\overline{x}_k)$ by replacing all occurrences of subformulas isomorphic to $P_i^1(\overline{y}_i^1)$ with the atomic formulas $p_i^1(x_{i\ k}^1)$ by $A_k^1(\overline{x}_k^1)$. Here $\overline{x}_k^1$ is the list of all variables of the formula $A_k^1(\overline{x}^1)$, consisting of some (perhaps all) initial variables of the formula $A_k(\overline{x}_k)$, and of first-level variables that appear in the formula $A_k^1(\overline{x}_k^1)$. Such formulas $A_k^1(\overline{x}_k^1)$ can be considered as class descriptions in terms of predicates of the initial (zero) and the first levels.

The first-level description $S^1(\omega)$ is a union of $S(\omega)$ and the set of all atomic formulas of the form $p_i^1(\omega_{ij}^1)$ for which the defining sub-formula $P_i^1(\overline{\tau}_{ij}^1)$ is true with $\tau_{ij}^1 \subset \omega$, and the first-level object ω_{ij}^1 is a list of initial objects $\overline{\tau}_{ij}^1$.

The procedure for extracting sub-formulas that are isomorphic to "often" occurring subformulas with "small complexity" can be repeated with formulas $A_k^1(\overline{x}_k^1)$.

As a result of constructing predicates of various levels and a level class description, the original class description system can be written using the equivalent level class description system

$$\left\{ \begin{array}{rcl} & A_k^L(\overline{x}_k^L) & \\ p_1^1(x_1^1) & \Leftrightarrow & P_1^1(\overline{y}_1^1) \\ & \vdots & \\ p_{n_1}^1(x_{n_1}^1) & \Leftrightarrow & P_{n_1}^1(\overline{y}_{n_1}^1) \\ & \vdots & \\ p_i^l(x_i^l) & \Leftrightarrow & P_i^l(\overline{y}_i^l) \\ & \vdots & \\ p_{n_L}^L(x_{n_L}^L) & \Leftrightarrow & P_{n_L}^L(\overline{y}_{n_L}^L) \end{array} \right. .$$

The solution of the considered recognition problems can be reduced to sequential implementation of the next operation for $l = 1, \ldots, L$.

- For each $j = 1, \ldots, n_l$ check the sequence from $S^{l-1}(\omega)$ the formula $\exists \overline{x}_{j\neq}^l\ P_j^l(\overline{x}_j^l)$ and find all such l-th level objects ω_j^l, whose existence is stated on the right-hand side of the logical sequence, and therefore the atomic formulas of $p_j^l(\omega_j^l)$ are true. In this case, a description of the l-th level $S^l(\omega)$ object will be obtained.

It should be noted that for each l a logical sequence of the form (4) with a shorter recording length of the right-hand side is checked. So, it is possible to clarify the concept of "small complexity".

For the exhaustive algorithm, this means a "small" number of object variables in the formulas. For algorithms based on the derivation in the predicate calculus, "small complexity" means "small" number of atomic formulas.

Note, that in spite of GI-completeness of the classification problem, the problems of idendification and of analysis remain to be NP-complete. This corresponds the facts that the problem **Isomorphism of a sub-graph** Garey, etc., [1979] is NP-complete.

Conjunctive Boolean Query problem

The following problem is presented in Garey, etc., [1979].

Conjunctive Boolean Query (CBQ)

Instance: Finite domain D, a collection $R = \{R_1, R_2, \dots, R_n\}$ of predicates, where each R_i defines a d_i-ary relation between entries from D, a set $S(D)$ of all atomic formulas with predicates from R which are true on D, and a conjunctive Boolean query Q over R and D, where such a query Q is of the form $A_1 \,\&\, A_2 \,\&\, \dots \,\&\, A_r$ with each A_i of the form $R_j(u_1, u_2, \dots, u_{d_j})$ where each $u \in \{y_1, y_2, \dots, y_l\} \cup D$.

Question: Is $\exists y_1, y_2, \dots, y_l Q$, when interpreted as a statement about R and D, true?

That is whether

$$S(D) \Rightarrow \exists y_1, y_2, \dots, y_l (A_1 \,\&\, A_2 \,\&\, \dots \,\&\, A_r)?$$

Such setting of the problem **CBQ** is very similar to the earlier investigated in Kosovskaya, [2014] problem Satisfiability in a Finite Interpretation appeared while recognition of an object in the frameworks of logic-objective approach to the pattern recognition and the question of which is the formula (4).

Essential difference in implementation of these problems consists in the following:

— data base may be not changeable at all or have very small changes, but queries may differ every time ($S(D)$ is fixed, but the query Q often may be changed);

— while pattern recognition the set of goal formulas (description of classes) may be not changeable at all or has changes very rarely, but the recognized objects may be different every time (the set of all possible formulas $A(\overline{y})$ is fixed, but the object ω and its description $S(\omega)$ often may be changed).

While multiple solution of the CBQ problem, the researcher does not have a set of elementary conjunctions. Analogues of frequently occurred sub-formulas have to be extracted from the set of literals, and the algorithm for constructing a multi-level class description cannot be applied. When creating a database, we cannot say with certainty what queries the user may have. However, the database itself with all requests remains almost unchanged. Therefore, regularities should be sought in the base itself (the set of constant literals $S(D)$).

An algorithm for finding such regularities (sets of conjunctions of atomic formulas that are isomorphic to each other) is given in Kosovskaya, [2018a]. This algorithm has an exponential upper estimate of computational complexity of the database record length. However, this algorithm may be applied to a database formulas only once.

Metrics in the space of elementary conjunctions of predicate formulas

When solving many AI problems, the question arises of how similarly two objects under study are alike. When simulation an object using binary (or finite-valued) strings, there naturally arises a feature space of a given dimension n. The distance between the descriptions of objects $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ in this space can naturally be calculated using the formula $\sqrt[k]{\sum_{i=1}^{n} |x_i - y_i|^k}$. Or, if the "weights" w_i of the features are known, then by the formula $\sqrt[k]{\sum_{i=1}^{n} w_i |x_i - y_i|^k}$.

In the logic-objective approach, the descriptions of objects can have a different number of literals, and the same predicate symbol can enter the description of an object a different number of times and with different arguments.

A metric for elementary conjunctions of predicate formulas is proposed in Kosovskaya, [2012].

Let $S(\omega_1)$ and $S(\omega_2)$ be descriptions of ω_1 and ω_2, respectively. Elementary conjunctions $S_f(\omega_1)$ and $S_f(\omega_2)$ are obtained from $S(\omega_1)$ and $S(\omega_2)$ by means of writing the sign $\&$ between literals. $S_f(\omega_i)$ ($i = 1, 2$) contains a_i literals.

Let us find a maximal, over the number of arguments, elementary conjunction of C, isomorphic to the subformulas of the formulas $S_f(\omega_1)$ and $S_f(\omega_2)$. Let C contains a' literals. Denote by $\Delta a_i = a_i - a'$ the number of literals from $S_f(\omega_i)$ not included in C. Then the formula

$$\rho(\omega_1, \omega_2) = \Delta a_1 + \Delta a_2 = a_1 + a_2 - 2a'$$

defines the distance between the descriptions of $S(\omega_1)$ and $S(\omega_2)$, satisfying all properties of the distance, namely:

— non-negativity,

— symmetry,

— equality to zero if and only if the descriptions are isimorphic,

— triangle inequality, i.e. for every three objects ω_1, ω_2 and ω_3 the inequality $\rho(\omega_1, \omega_2) + \rho(\omega_2, \omega_3) \geq \rho(\omega_1, \omega_3)$ is true.

The disadvantage of such a way defined distance is that it does not reflect (illustrate) the degree of similarity of objects. For example, for objects with descriptions containing a large number of literals, if they coincide by 90%, there can be a large distance. For objects with a small number of literals, if they coincide by 10%, there may be a small distance. You can overcome of this disadvantage by normalizing the distance.

$$d(\omega_1, \omega_2) = \frac{\rho(\omega_1, \omega_2)}{a_1 + a_2}.$$

Unfortunately, this formula does not define a metric, since the triangle inequality is not satisfied. Therefore, the function d can be called the degree of similarity of the descriptions of the objects ω_1 and ω_2.

Conclusion

The paper considers various aspects of the application of the notion of isomorphism of elementary conjunctions of predicate formulas in AI problems. The term of isomorphism of various objects is widespread in mathematics. However, in the most of the author's works cited here, instead of the term "isomorphism", the term "coincidence up to the names of variables" was used.

It was the understanding of the fact that the relation under consideration is an isomorphism relation that led to the appearance of this article. Moreover, a connection between problems using this relation and the graph isomorphism problem was realized. For the last one Laslo Babai in 2017 Babai, [2017] proposed a quasi-polynomial algorithm for solving it. His estimate $2^{O((\log\ n)^3)}$ is still being verified.

Bibliography

Babai L. "Graph Isomorphism in Quasipolynomial Time" (Version 2.1 Unfinished revision of Version 2 posted on arXiv May 23, 2017). http://people.cs.uchicago.edu/∼laci/17groups/version2.1.pdf

Garey M.R., Johnson D.S., "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman, New York, 1979.

Kosovskaya T. "Discrete Artificial Intelligence Problems and Number of Steps of their Solution" // International Journal on Information Theories and Applications, Vol. 18, Number 1, 2011. P. 93 ï£¡ 99.

Kossovskaya T. M. "Level descriptions of classes for decreasing step number of pattern recognition problem solving described by predicate calculus formulas" // Vestn. St. Petersburg Univ.: Seria 10, Issue. 1. P. 64 – 72 (2008). (In Russian)

Kosovskaya T. "Distance between objects described by predicate formulas" // International Book Series. Information Science and Computing. Book 25. Mathematics of Distances and Applications (Michel Deza, Michel Petitjean, Krasimir Markov (eds)), ITHEA ï£¡ Publisher, Sofia, Bulgaria, 2012. P. 153 ï£¡ 159.

Kosovskaya T. "Construction of Class Level Description for Efficient Recognition of a Complex Object" // International Journal "Information Content and Processing", Vol. 1,No 1. 2014. P. 92 – 99.

Kosovskaya T.M. "Predicate Calculus as a Tool for AI Problems Solution: Algorithms and Their Complexity", Chapter 3 in: Intelligent System. Open access peer-reviewed Edited volume. Edited by Chatchawal Wongchoosuk Kasetsart University (2018) 1 – 20. https://www.intechopen.com/books/intelligent-system/predicate-calculus-as-a-tool-for-ai-problems-solution-algorithms-and-their-complexity

Kosovskaya T.M. "Construction of a level data base for decreasing of computational complexity of the Conjunctive Boolean Query problem" // 11th Russian Multiconference on the Control Problems. Proc. of the conference "Information Technology in Management". St. Petersburg, 2018. P. 33 –38. (In Russian)

Petrov D.A. "Algorithms of extraction of a maximal common up to the names of variables predicate formulas and their implementation" // Proc. of the 9-th Conference "Information Technology in Management". St.Petersburg, 2016. P. 97 –102. (In Russian)

Russell S. and Norvig P. "Artificial Intelligence: A Modern Approach", Third edition, Prentice Hall Press Upper Saddle River, NJ, 2009.

Authors' Information

Tatiana Kosovskaya *- Dr., Professor of St.Petersburg State University, University emb., 7 – 9, St.Petersburg, 199034, Russia; e-mail: kosovtm@gmail.com*
Major Fields of Scientific Research: Logical approach to artificial intelligence problems, Theory of Computational Complexity of Algorithms.