CSIT Conference 2019, Yerevan, Armenia, September 23-27

Implementation of Formula Partial Sequence for Rough Solution of AI Problems in the Framework of the Logic-Predicate Approach

Tatiana, Kosovskaya

St. Petersburg State University
St. Petersburg, Russia

e-mail: kosovtm@gmail.com

ABSTRACT

The paper is the answer to the question posed to the author during a report at the CSIT-2017 conference: whether it is possible to change a logic-predicate network so that not only objects with descriptions from the training set are recognized, but also differ slightly from them. The notion of partial sequence of a predicate formula, introduced by the author earlier, makes it possible to change the content of network cells so that the degree of similarity of a recognizable object fragment to fragments of objects from the training set, and then the degree of certainty that the object belongs to a given class, are calculated. A brief description of the logic-predicate approach to AI problems, the information about a logic-predicate network, the notion of partial sequence are presented in the paper.

Keywords

Predicate calculus, logic-predicate approach to AI, level description, predicate network, partial sequence.

1. INTRODUCTION

Many Artificial Intelligence problems permit their formalization by means of predicate calculus language. Problems formulated in such a way are NP-complete or NP-hard ones [2]. The computational complexity of such problems, when being solved by an exhaustive search algorithm, coincides with the length of their encoding using a binary string [11].

A hierarchical level description of classes was suggested in [3] to decrease the computational complexity of these problems. To construct such a level description, “frequently appeared” sub-formulas of “small complexity” are extracted from class descriptions. This allows to decompose the main problem into a series of similar problems with input data with a less length.

Construction of a level logic-predicate network with the use of level description of classes was offered in [6] and described in [8]. It is based on the extraction from the descriptions of classes of objects of such fragments, which appear in many objects of classes. Recognition itself is reduced to the sequential solution of problems of the same type with a less length of input data.

A modification of the logic-predicate network allowing to recognize approximately new objects, is offered in the paper. The degree of coincidence is calculated for description of an object part and the formula, satisfiability of which is checked in the cell.

2. LOGIC-PREDICATE APPROACH TO AI PROBLEMS

A detailed description of the logic-predicate approach to solving AI problems is available in [9]. Here only the main problem setting and some methods of its solution are formulated.

Let an investigated object be represented as a set of its elements $\omega = \{\omega_1, \dots, \omega_t\}$ and be characterized by predicates $p_1, \dots, p_n$ which define some properties of the elements or relations between them. The description $S(\omega_1, \dots, \omega_t)$ of the object ω is a set of all constant literals with predicates $p_1, \dots, p_n$ which are valid on ω. There is a set of goal formulas $A_1(x_1, \dots, x_{m_1}), \dots, A_K(x_1, \dots, x_{m_K})$ in the form of elementary conjunctions of atomic formulas.

The solution of many Artificial Intelligence problems may be reduced to the proof of a series of formulas (for $k = 1, \dots, K$) in the form

$$S(\omega) \Rightarrow \exists (x_1 \dots x_{m_k})_{\neq}\, A_k(x_1, \dots, x_{m_k}).^1 \qquad (1)$$

If the number m of arguments in $A_k(x_1, \dots, x_{m_k})$ is not equal to the number t of elements in ω then the verification problem of such a formula is NP-complete [5].

If the number m of arguments in $A_k(x_1, \dots, x_{m_k})$ equals to the number t of elements in ω then the problem (1) becomes a GI-complete one [10]. It means that it is polynomially equivalent to an “open” problem *Graph Isomorphism* for which it is not proved its NP-completeness or its polynomiality is proved [1].

Note that both a logical algorithm and an exhaustive search algorithm allow not only to prove that there exist values for variables satisfying the formula $A_k(x_1, \dots, x_m)$ but to find these values.

So an algorithm verifying the formula (1) allows to solve the problem “what are the different values of $x_1, \dots, x_m$ from ω that satisfy the formula $A_k(x_1, \dots, x_m)$?”

$$S(\omega) \Rightarrow ?(x_1, \dots, x_m)_{\neq} A_k(x_1, \dots, x_m). \qquad (2)$$

[1]The notation $\exists (x_1 \dots x_m)_{\neq}\, P$ is used for the formula $\exists x_1 \dots x_m (\&_{i=1}^{m-1} \&_{j=i+1}^{m}\, x_i \neq x_j\ \&\ P)$.

This problem is NP-hard and its solving algorithms have the same upper bounds as for the problem (1).

3. COMMON UP TO THE NAMES OF ARGUMENTS SUB-FORMULA

Definition 1. *Elementary conjunctions P and Q are called isomorphic if there is an elementary conjunction R and substitutions $\lambda_{R,P}$ and $\lambda_{R,Q}$ of the arguments of P and Q, respectively, instead of the variables in R such that the results of these substitutions coincide up to the order of literals.*

The substitutions $\lambda_{R,P}$ and $\lambda_{R,Q}$ are called unifiers of R with P and Q, respectively.

Definition 2. *Elementary conjunction C is called a common up to the names of arguments sub-formula of two elementary conjunctions A and B if it is isomorphic to some sub-formulas A' and B' of A and B, respectively.*

An algorithm of extraction of a maximal (having a maximal number of literals) common up to the names of arguments sub-formula C of two elementary conjunctions A and B and determining the unifiers $\lambda_{C,A'}$ and $\lambda_{C,B'}$ is described in [7]. It is based on the notion of partial sequence.

The number of steps of this algorithm is $O(N_A^{N_A} N_B^{N_B})$, where N_A and N_B are the numbers of literals in A and B, respectively. The minimal number of steps of this algorithm is $O((N_A N_B)^2)$, the middle estimate is $O((N_A N_B)^{1/2 \log(N_A N_B)})$.

4. LEVEL DESCRIPTION

Level description of goal formulas allows essentially to decrease the number of steps for an algorithm solving problems (1) and (2). This notion is based on the extraction of common up to the names of arguments sub-formulas $P_i^1(\overline{y}_i^1)$ $(i = 1, \dots, n_1)$ of goal formulas $A_k(x_1, \dots, x_m)$ $(k = 1, \dots, K)$ with "small complexity" and then sequentially check formulas in the form (1) with essentially less lengths. Simultaneously we find unifiers of $P_i^1(\overline{y}_i^1)$ and sub-formulas of $A_k(x_1, \dots, x_m)$.

$$\left\{ \begin{array}{ccc} & A_k^L(\overline{x}_k^L) & \\ p_1^1(y_1^1) & \Leftrightarrow & P_1^1(\overline{y}_1^1) \\ & \vdots & \\ p_{n_1}^1(y_{n_1}^1) & \Leftrightarrow & P_{n_1}^1(\overline{y}_{n_1}^1) \\ & \vdots & \\ p_i^l(y_i^l) & \Leftrightarrow & P_i^l(\overline{y}_i^l) \\ & \vdots & \\ p_{n_L}^L(y_{n_L}^L) & \Leftrightarrow & P_{n_L}^L(\overline{y}_{n_L}^L) \end{array} \right. . \qquad (3)$$

Here p_i^l $(l = 1, \dots, L)$ are new l-level predicates with new l-level variables y_i^l for lists of a less-level variables defined by the equivalences $p_i^l(y_i^l) \Leftrightarrow P_i^l(\overline{y}_i^l)$. $A_k^L(\overline{x}_k^L)$ are results of substitutions of $p_i^l(y_i^l)$ into $A_k(\overline{x}_k)$ instead of their sub-formulas $P_i^l(\overline{y}_i^l)$.

Construction of a level description is described in [9, 8]. Let N be the maximal number of literals in $A_k(\overline{x}_k)$ $(k = 1, \dots, K)$. The upper bound of this algorithm number of steps is $O(K^2 N^{2N})$.

5. LOGIC-PREDICATE NETWORK

A logic-predicate network consists of two blocks: a training block and a recognition block [6, 9]. Let a training set of objects $\omega^1, \dots, \omega^K$ be given to form an initial variant of the network training block. Replace every constant ω_j^k in $S(\omega^k)$ by a variable x_j^k $(k = 1, \dots, K,\ j = 1, \dots, t^k)$ and substitute the sign & between the atomic formulas. Initial goal formulas $A_1(\overline{x}_1), \dots, A_K(\overline{x}_K)$ are obtained. Construct a level description for these goal formulas. The first approximation to the recognition block is formed.

If after the recognition block run an object is not recognized or has a wrong identification then it is possible to train the network anew. The description of the "wrong" object must be added to the input set of the training block. The training block extracts common sub-formulas of this description and previously received formulas forming the recognition block. Some sub-formulas in the level description would be changed. Then the recognition block is reconstructed.

The problem is that such a network recognizes only objects that have been presented in the training set or according to which it was re-trained. But it recognizes them exactly.

6. PARTIAL SEQUENCE

The problem of checking if the formula $A(\overline{x})$ or some its sub-formula $\widetilde{A}(\overline{y})$ is a consequence of the set of formulas $S(\omega)$ is under consideration in [4]. Here the list of arguments $\overline{y}$ is a sub-list of arguments $\overline{x}$.

Every sub-formula $\widetilde{A}(\overline{y})$ of the formula $A(\overline{x})$ is called its **fragment**.

Let a and $\widetilde{a}$ be the numbers of atomic formulas in $A(\overline{x})$ and $\widetilde{A}(\overline{y})$, respectively, m and $\widetilde{m}$ be the numbers of objective variables in $\overline{x}$ and $\overline{y}$, respectively.

Numbers q and r are calculated by the formulas $q = \frac{\widetilde{a}}{a}$, $r = \frac{\widetilde{m}}{m}$ and characterize the degree of coincidence between $A(\overline{x})$ and $\widetilde{A}(\overline{y})$. For every $A(\overline{x})$ and its fragment $\widetilde{A}(\overline{y})$ it is true that $0 < q \leq 1$, $0 < r \leq 1$. Besides, $q = r = 1$ if and only if $\widetilde{A}(\overline{y})$ coincides with $A(\overline{x})$.

Under these notations, the formula $\widetilde{A}(\overline{y})$ will be called a (q, r)**-fragment of the formula** $A(\overline{x})$.

If $S(\omega) \Rightarrow \exists \overline{x}_{\neq}\, A(\overline{x})$ is not valid but for some (q, r)-fragment $\widetilde{A}(\overline{y})$ $(q \neq 1)$ of $A(\overline{x})$ the sequent $S(\omega) \Rightarrow \exists \overline{y}_{\neq}\, \widetilde{A}(\overline{y})$ is true, we will say that $S(\omega) \Rightarrow_P \exists \overline{x}_{\neq}\, \widetilde{A}(\overline{x})$ is a **partial** (q, r)**-sequent**.

As the checking whether $S(\omega) \Rightarrow \exists \overline{y}_{\neq}\, \widetilde{A}(\overline{y})$ may be done by some constructive method (for example, by exhaustive search or by deduction in a sequential predicate calculus) then such values $\overline{\tau}$ $(\tau \subseteq \omega)$ for the list of variables $\overline{y}$ that $S(\omega) \Rightarrow \widetilde{A}(\overline{\tau})$ will be found.

Definition 3. *Conjunction of literals from $A(\overline{x})$ which* *are not in $\widetilde{A}(\overline{y})$ is called a* **complement** *of $\widetilde{A}(\overline{y})$ up to $A(\overline{x})$.*

A complement of $\widetilde{A}(\overline{y})$ up to $A(\overline{x})$ will be denoted by $C_{A(\overline{x})}\widetilde{A}(\overline{y})$.

Definition 4. A (q,r)-fragment $\widetilde{A}(\overline{y})$ of the formula $A(\overline{x})$ is called **contradictory** *to the description $S(\omega)$ on the list of constants $\overline{\tau}$ if $S(\omega)$ and $C_{[A(\overline{x})]^{\overline{y}}_{\overline{\tau}}}\widetilde{A}(\overline{\tau})$ lead to the contradiction, i.e., $S(\omega) \Rightarrow \neg C_{[A(\overline{x})]^{\overline{y}}_{\overline{\tau}}}\widetilde{A}(\overline{\tau})$.*

Here the denotation $[A(\overline{x})]^{\overline{y}}_{\overline{\tau}}$ is used for the result of substitution of the constants from the list $\overline{\tau}$ instead of the corresponding variables from the list $\overline{y}$.

7. FUZZY RECOGNITION BY A LOGIC-PREDICATE NETWORK

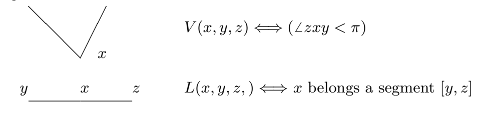
It is suggested to change the content of the network cells by replacing the checking of $S^{l-1}(\omega) \Rightarrow \exists \overline{x}^l_{i\neq} P^l_i(\overline{x}^l_i)$ in the ith cell of the lth level with the partial sequence checking $S^{l-1}(\omega) \Rightarrow_P \exists \overline{x}^l_{i\neq} P^l_i(\overline{x}^l_i)$.

While partial sequence checking, all lists of constants $\overline{\tau}^l_{i\ j}$ and maximal not contradictory on $\overline{\tau}^l_{i\ j}$ with $S^{l-1}(\omega)$ sub-formula $\widetilde{P}^l_{i\ j}(\overline{x}^l_{i\ j})$ of the formula $P^l_i(\overline{x}^l_i)$ are found. Parameters $q^l_{i\ j}$ and $r^l_{i\ j}$ are calculated with the use of the full form of formulas $P^l_i(\overline{x}^l_i)$ and $\widetilde{P}^l_i(\overline{y}^l_i)$, i.e., with the replacement in them of each atomic formula of the levels l' $(l' < l)$ by the defining elementary conjunction.

Except this, a "degree of certainty" $cert^l_i$ that the recognition would be valid is calculated in every cell. Denotation pre^l_i will be used for the number of the cell preceding the ith cell of the lth level while the current traversal of the graph. Initially, $cert^0_1 = 1$, $pre^1_i = 0$ $(i = 1, \ldots, n_1)$. While first visit of the ith cell of the lth level $cert^l_i := \min\{cert^{1-1}_{pre^l_i}, q^l_i\}$. This corresponds to conjunction of degrees of certainty while successive passage along one branch of network traversal. While next visit of the ith cell of the lth level $cert^l_i = \max\{cert^l_i, \min\{cert^{1-1}_{pre^l_i}, q^l_i\}\}$. This corresponds to disjunction of degrees of certainty while parallel passage along different branches of network traversal.

Example

To describe the contour images, there are two predicates V and L presented in Figure 1.

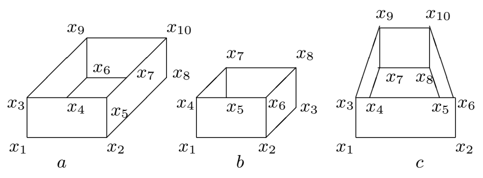


y
z
x
$V(x,y,z) \Longleftrightarrow (\angle zxy < \pi)$
y
x
z
$L(x,y,z,) \Longleftrightarrow x$ belongs a segment $[y,z]$

Figure 1. Initial predicates.

Representatives of the contour images of the "boxes" described by predicates V and L are given. After replacing the constants with variables, we obtain the training sample presented in Figure 2.

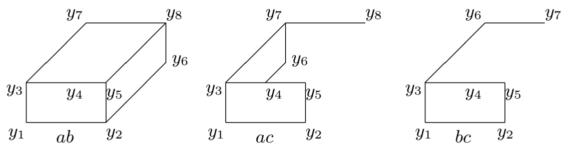
Extract from the formulas, corresponding to the images a, b and c, maximal sub-formulas isomorphic to each



x_9 x_{10} x_6 x_7 x_8 x_3 x_4 x_5 x_1 x_2 a
x_7 x_8 x_4 x_5 x_6 x_3 x_1 x_2 b
x_9 x_{10} x_7 x_8 x_3 x_4 x_5 x_6 x_1 x_2 c

Figure 2. Training set.

other. Thus, we obtain the common to the names of variables sub-formulas, corresponding to the images in Figure 3.



y_7 y_8 y_6 y_3 y_4 y_5 y_1 ab y_2
y_7 y_8 y_6 y_3 y_4 y_5 y_1 ac y_2
y_6 y_7 y_3 y_4 y_5 y_1 bc y_2

Figure 3. Images of the extracted sub-formulas.

The numbers of variables and predicates in the formulas, corresponding to these images, are $m_{ab} = 8$, $a_{ab} = 20$; $m_{ac} = 8$, $a_{ac} = 16$; $m_{bc} = 7$, $a_{bc} = 11$, respectively.

The formula corresponding to the image *bc* is isomorphic to sub-formulas of the formulas corresponding to the images *ab* and *ac*. Therefore, it forms the first level of the network. Formulas corresponding to the images *ab* and *ac* form the 2nd level of the network. Formulas corresponding to the training set will form the 3rd level of the network.

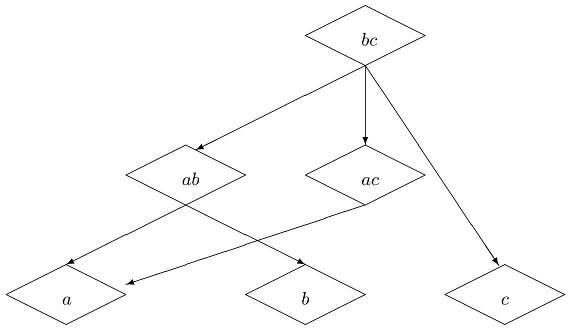


Figure 4. Logic-predicate network.

Schematically, this network is presented in Figure 4. At the same time, the content in the cells of the scheme means that the partial sequent $S(\omega) \Rightarrow_P \exists \overline{x}_{\neq}\ A(\overline{x})$ is checked in this cell. Here $S(\omega)$ is the description of the object being recognized, $A(\overline{x})$ is the formula that defines the corresponding image. In addition, values for variables are found and the consistency of the complement to $\widetilde{A}(\overline{x})$ is checked.

An object presented in Figure 5 is given for recognition.

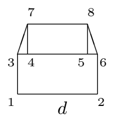


Figure 5. Image of a control object.

To recognize it, partial sequence "*d*"$(1,2,\ldots,8) \Rightarrow_P \exists y_1 \ldots y_7$"*bc*"$(y_1,\ldots,y_7)$ is checked in the 1st layer.[2]

[2]The notation "X"$(\overline{x})$ is used for the formula describing image X with arguments $\overline{x}$.

Sequence "d"$(1, 2, \ldots, 8) \Rightarrow \exists y_1 \ldots y_7$"$bc$"$(y_1, \ldots, y_7)$ is valid. Variables have values $y_1 = 1$, $y_2 = 2$, $y_3 = 3$, $y_4 = 4$, $y_5 = 6$, $y_6 = 7$, $y_7 = 8$. First-level variable y^1 has the value $a^1 = (1, 2, 3, 4, 6, 7, 8)$. $cert^d_{bc} = 1$.

At first partial sequence "d^1"$(a^1, 5) \Rightarrow_P \exists y_1 \ldots y_7$"$ab$"$(a^1, y_6)$ is checked in the 2nd layer. The only consistent fragment coincides with the formula describing bc. Hence, $\widetilde{m}_{ab} = 7$, $\widetilde{a}_{ab} = 11$, $cert^d_{ab} = \min\{1, \frac{11}{20}\} = \frac{11}{20}$.

When checking "$d^{1''}(a^1, 5) \Rightarrow_P \exists y_1 \ldots y_7$"$ac$"$(a^1, y_6)$ in the second layer, the same consistent fragment is selected. But since the number of predicates in the description of ac is 16, then $cert^d_{ac} = \min\{1, \frac{11}{16}\} = \frac{11}{16}$.

In the third layer in the jump from ab to a, the same consistent fragment bc is extracted. But since the number of predicates in the description of a is 31, then $cert^d_{ab \to a} = \min\{\frac{11}{20}, \frac{11}{31}\} = \frac{11}{31} \approx 0.355$.

In the third layer in the jump from ab to b, the segment $[7, 4]$ is added to the earlier exteacted fragment. It changes its number of predicates in the fragment $\widetilde{m}_b = 7$, $\widetilde{a}_b = 13$. $cert^d_{ab \to b} = \min\{\frac{11}{31}, \frac{13}{24}\} = \frac{13}{24} \approx 0.541$.

In the third layer in the jump from bc to c, the node 5 is added to the earlier exteacted fragment. It changes its number of predicates in the fragment $\widetilde{m}_c = 8$, $\widetilde{a}_c = 20$. $cert^d_{bc \to c} = \min\{1, \frac{20}{34}\} = \frac{20}{34} \approx 0.588$.

The general degree of certainty that a representative of the class " boxes " is given as a control image is $cert = \max\{\frac{11}{31}, \frac{13}{24}, \frac{11}{31}, \frac{20}{34}\} = \frac{20}{34} \approx 0.588$. Moreover, the $\frac{7}{8}$ elements of this image coincides with the $\frac{8}{10}$ elements of the standard image c.

8. CONCLUSION

The notion of partial sequence of the predicate formula, based on the notion of a common up to the names of arguments sub-formula of two elementary conjunctions of predicate formulas, made it possible to propose a modification of the logical-predicate recognition network. This modification allows not only to determine the exact class to which a control object belongs (if its description coincides with the description of some object from the training set), but also to calculate the "degree of certainty " that the recognition is correct for a control object with the description which is different from some standard one.

Moreover, if the "degree of certainty" is sufficiently high, then we can retrain the network and reconstruct it for exact recognition in the future for objects with a similar description. This is important because checking the partial sequence of a formula has a significantly greater computational complexity compared to checking the sequence of a formula, even if the formulas in each node are rather short (or have a small number of arguments).

REFERENCES

[1] M. Garey, D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness." Freeman Press. San Francisco, 1979.

[2] T.M. Kosovskaya. "Proofs of the number of steps bounds for solving some pattern recognition problems with logical description." *Vestnik of Saint-Petersburg University. Series 1: Mathematics, mechanics, astronomy*, pp. 82–90, 2007, issue 4. (In Russian)

[3] T.M. Kosovskaya. "Level descriptions of classes for decreasing of step number of solving of a pattern recognition problem described by predicate calculus formulas", *Vestnik of Saint-Petersburg University. Series 10. Applied mathematics. Computer science. Control processes.*, pp. 64 – 72, 2008, issue 1. (In Russian)

[4] T.M. Kossovskaya. "Partial deduction of predicate formula as an instrument for recognition of an object with incomplete description", *Vestnik of Saint-Petersburg University. Series 10. Applied mathematics. Computer science. Control processes.*, pp. 74 – 84, 2009, issue. 1. (In Russian)

[5] 8. T.M. Kosovskaya. "Some artificial intelligence problems permitting formalization by means of predicate calculus language and upper bounds of their solution steps", *SPIIRAS Proceedings*, **14**, pp. 58 – 75, 2010. (In Russian)

[6] N. Kosovskaya. "Self-modificated predicate networks", *International Journal on Information Theory and Applications*, Vol. 22, No 3, pp. 245 – 257, 2015.

[7] Kosovskaya T. M., Petrov D. A. "Extraction of a maximal common sub-formula of predicate formulas for the solving of some Artificial Intelligence problems", *Vestnik of Saint-Petersburg University. Series 10. Applied mathematics. Computer science. Control processes.*, pp. 250 – 263, 2017, issue 3. (In Russian)

[8] T. Kosovskaya. "Extraction of a common up to the names of arguments sub-formula of two elementary conjunctions and some AI problems", *Proceedings of 11th International Conference on Computer Science and Information Technologies (CSIT 2017), Yerevan, Armenia, 25 - 29 September 2017*, pp. 206 – 209, 2017.

[9] T. Kosovskaya. "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, pp. 1 – 20, 2018.

https://www.intechopen.com/books/intelligent-system/predicate-calculus-as-a-tool-for-ai-problems-solution-algorithms-and-their-complexity

[10] T.M. Kosovskaya, N.N. Kosovskii. "Polynomial equivalence of the problems PREDICATE FORMULAS ISOMORPHISM and GRAPH ISOMORPHISM", *Vestnik of Saint-Petersburg University. Series 1: Mathematics, mechanics, astronomy*, Vol. 6(64), Issue 3, pp. 430 – 439, 2019. (In Russian)

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