PARTIAL DEDUCTION IN PREDICATE CALCULUS AS A TOOL FOR ARTIFICIAL INTELLIGENCE PROBLEM COMPLEXITY DECREASING

Tatiana M. Kosovskaya

Abstract: *Many artificial intelligence problems are NP-complete ones. To decrease the needed time of such a problem solving a method of extraction of sub-formulas characterizing the common features of objects under consideration is suggested. This method is based on the offered by the author notion of partial deduction. Repeated application of this procedure allows to form a level description of an object and of classes of objects. A model example of such a level description and the degree of steps number increasing is presented in the paper.*

Keywords: *Artificial Intelligence, pattern recognition, predicate calculus, level description of a class..*

1. Introduction

While simulation an Artificial Intelligence (AI) problem the most of researchers consider an investigated object as a unit which is characterized by some global features [12]. In particular, a researcher using methods of mathematical logic operates with propositional formulas or Boolean functions [2]. Such an approach is not convenient for a simulation of a complex object which is described by properties of its elements and relations between them.

At the same time there are many papers which offer to use predicate calculus and resolution method for the above mentioned problems [13; 14]. The predicate calculus language is enough adequate to simulate complex or changeable objects. But, unfortunately, the authors do not take into account the time complexity of a problem using such a simulation.

The point is that a problem using such a simulation is an NP-hard [6]. If **P**$\neq$**NP** then such a problem may be solved only in the time exponentially depended of the input [5; 3].

The upper bounds of number of steps for algorithms solving some AI problems described by the predicate calculus language were proved by the author in [6; 10; 7]. The analysis of thees upper bounds allowed to develop hierarchical many-level descriptions of the goal conditions which essentially decrease the solving time for the mentioned problems [8]. But at that time there was no tool for automatic construction of a level description. Intuitive construction of such a description showed that the time decreases.

The notion of partial deduction [9] earlier introduced by the author for the recognition of an object with incomplete information occurred to be such a suitable tool.

Some AI problems using predicate language description which may be simplified with the use of partial deduction are presented in this paper.

2. Logic-objective approach to a recognition problem

Let an investigated object ω is represented as a set of its elements $\omega = \{\omega_1, ..., \omega_t\}$ and predicates $p_1, ..., p_n$ define properties of these elements and relations between them.

Logical description $S(\omega)$ of an object ω is a collection of all true formulas in the form $p_i(\overline{\tau})$ or $\neg p_i(\overline{\tau})$ (where $\overline{\tau}$ is an ordered subset of ω) describing properties of ω elements and relations between them.

Let the set of all investigated objects Ω is a union of classes $\Omega = \cup_{k=1}^{K} \Omega_k$.

Logical description of the class Ω_k is such a formula $A_k(\overline{x})$ that if the formula $A_k(\overline{\omega})$ is true then $\omega \in \Omega_k$. The class description may be represented as a disjunction of elementary conjunctions of atomic formulas.

Here and below the notation $\overline{x}$ is used for an ordered list of the set x. To denote that there exist distinct values for variables from the list $\overline{x}$ the notation $\exists\overline{x}_{\neq}A_k(\overline{x})$ is used.

The introduced descriptions allow to solve many artificial intelligence problems which may be formulated as follows.

Identification problem. To pick out all parts of the object ω which belong to the class Ω_k.

Classification problem. To find all such class numbers k that $\omega \in \Omega_k$.

Analysis problem. To find and classify all parts τ of the object ω.

These problems are reduced in [1] to the following formulas respectively

$$S(\omega) \Rightarrow (?\overline{x}_k)A_k(\overline{x}_k), \tag{1}$$

$$S(\omega) \Rightarrow (?k)A_k(\overline{x}_k), \tag{2}$$

$$S(\omega) \Rightarrow (?k)(?\overline{x}_k)A_k(\overline{x}_k), \tag{3}$$

where $(?k)$ and $(?\overline{x})$ denote the words "what are the values of k?" and "what are the values of $\overline{x}$?".

It is proved in [6] that the corresponding recognition problems

$$S(\omega) \Rightarrow \exists\overline{x}_{k\neq}A_k(\overline{x}_k), \tag{4}$$

$$S(\omega) \Rightarrow \vee_{k=1}^{K}A_k(\overline{x}_k), \tag{5}$$

$$S(\omega) \Rightarrow \vee_{k=1}^{K}\exists\overline{x}_{k\neq}A_k(\overline{x}_k) \tag{6}$$

are NP-complete. Hence the problems (1), (2), (3) are NP-hard.

3. Methods of proof and upper bounds of their number of steps

If one can solve the problem (1) with $A_k(\overline{x})_k$ be a conjunction of atomic formulas then he can solve the problems (1) with arbitrary $A_k(\overline{x}_k)$, (2) and (3), and the number of steps of their solutions would differ from the first one polynomially. If we solve problems (4), (5), (6) by means of a "constructive" algorithm (i.e. algorithm not only proves the existence but also finds values for variables $\overline{x}$ and parameter k) then we simultaneously solve problems (1), (2), (3). That is why the complexity bounds of algorithms will be done for the problem (4) in the form

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

where $A(\overline{x})$ is a conjunction of atomic formulas.

The **exhaustive search method** is one which allows to finds values for variables $\overline{x}$. It is proved in [6] that its number of steps is

$$O(t^m), \tag{7}$$

where t is the number of the elements in ω, m is the number of variables in the formula $A(\overline{x})$. Note that this estimate coincides with the one for simulation of predicate approach to the artificial intelligence problems by boolean formulas [14].

Logical methods (namely logical derivation in a sequent calculus or by resolution method) also allow to finds values for variables $\overline{x}$. Both these methods has the number of steps

$$O(s^a), \tag{8}$$

where s is the maximal number of occurrences of the same predicate in the description $S(\omega)$ and a is the number of atomic formulas in the formula $A(\overline{x})$ [10].

4. Level approach to the decision of problems

To decrease the obtained step number estimates a level description of goal formulas was offered in [8; 11]. Let $A_1(\overline{x}_1)$, ..., $A_K(\overline{x}_K)$ be a set of goal conditions every of which is a conjunction of atomic formulas. Find all subformulas $P_i^1(\overline{y}_i^1)$ with a "small" complexity which "frequently" appear in goal formulas $A_1(\overline{x}_1)$, ..., $A_K(\overline{x}_K)$ and denote them by atomic formulas with new predicates p_i^1 and new first-level arguments z_i^1 for lists $\overline{y}_i^1$ of initial variables. Write down a system of equivalences

$$p_i^1(z_i^1) \Leftrightarrow P_i^1(\overline{y}_i^1), \quad i = 1, \ldots, n_1.$$

What object must be called a "common sub-formula" of two formulas A and B?

For example, let

$$A(x,y,z) = p_1(x)\&p_1(y)\&p_1(z)\&p_2(x,y)\&p_3(x,z),$$
$$B(x,y,z) = p_1(x)\&p_1(y)\&p_1(z)\&p_2(x,z)\&p_3(x,z).$$

If the formula

$$P(u,v) = p_1(u)\&p_1(v)\&p_2(u,v)$$

is their common sub-formula?

The formula $P(u,v)$ is their common up to the names of variables sub-formula with the substitutions $\lambda_{P,A} = |_{x\ y}^{u\ v}$ and $\lambda_{P,B} = |_{x\ z}^{u\ v}$ because

— $P(x,y) = p_1(x)\&p_1(y)\&p_2(x,y)$ is a sub-formula of $A(x,y,z) = p_1(x)\&p_1(y)\&p_1(z)\&p_2(x,y)\&p_3(x,z)$,

— $P(x,z) = p_1(x)\&p_1(z)\&p_2(x,z)$ is a sub-formula of $B(x,y,z) = p_1(x)\&p_1(y)\&p_1(z)\&p_2(x,z)\&p_3(x,z)$.

Definition. *The formula P is called a common up to the names of variables sub-formula of formulas A and B if there are such substitutions $\lambda_{P,A} = |_{\overline{t}_A}^{\overline{x}}$ and $\lambda_{P,B} = |_{\overline{t}_B}^{\overline{x}}$ of the lists of terms $\overline{t}_A$ and $\overline{t}_B$ instead of the list of variables $\overline{x}$ that the formula P turns into a sub-formula of A and B respectively.*

Such substitutions are called unifiers of P with A and B respectively.

Let $A_k^1(\overline{x}_k^1)$ be a formula received from $A_k(\overline{x}_k)$ by substitution of $p_i^1(z_i^1)$ instead of $P_i^1(\overline{y}_i^1)$. Here $\overline{x}_k^1$ is a list of all variables in $A_k(\overline{x}_k^1)$ including both some (may be all) initial variables of $A_k(\overline{x}_k)$ and first-level variables appeared in the formula $A_k^1(\overline{x}_k^1)$.

A set of all atomic formulas of the type $p_i^1(\omega_i^1)$ where ω_i^1 denotes some ordered list $\overline{\tau}_i^1$ of elements from ω for which the formula $P_i^1(\overline{\tau}_i^1)$ is valid is called a first-level object description and denoted by $S^1(\omega)$. Such a way extracted subsets $\overline{\tau}_i^1$ are called first-level objects.

Repeat the above described procedure with formulas $A_k^1(\overline{x}_k^1)$. After L repetitions L-level goal conditions in the following form will be received.

$$\begin{cases} & A_k^L(\overline{x}_k^L) & \\ p_1^1(z_1^1) & \Leftrightarrow & P_1^1(\overline{y}_1^1) \\ & \vdots & \\ p_{n_1}^1(z_{n_1}^1) & \Leftrightarrow & P_{n_1}^1(\overline{y}_{n_1}^1) \\ & \vdots & \\ p_i^l(z_i^l) & \Leftrightarrow & P_i^l(\overline{y}_i^l) \\ & \vdots & \\ p_{n_L}^L(z_{n_L}^L) & \Leftrightarrow & P_{n_L}^L(\overline{y}_{n_L}^L) \end{cases}.$$

Such L-level goal conditions may be used for efficiency of an algorithm solving a problem formalized in the form of logical sequent (3). To decrease the number of steps of an exhaustive algorithm (for every t greater than some t_0)

with the use of 2-level goal description it is sufficient

$$n_1 \cdot t^r + t^{s_1+n_1} < t^m, \tag{7}$$

where r is a maximal number of arguments in the formulas $P_i^1(\overline{y}_i^1)$, n_1 is the number of first-level predicates, s_1 is the number of atomic formulas in the first-level description, m is the number of variables in the initial goal condition.

Similar condition for decreasing the number of steps of a logical algorithm solving the problem (3) is

$$\sum_{k=1}^{K} s^{1^{a_k^1}} + \sum_{j=1}^{n_1} s^{\rho_j^1} < \sum_{k=1}^{K} s^{a_k}, \tag{8}$$

where a_k and a_k^1 are maximal numbers of atomic formulas in $A_k(\overline{x}_k)$ and $A_k^1(\overline{x}_k^1)$ respectively, s and s^1 are numbers of atomic formulas in $S(\omega)$ and $S^1(\omega)$ respectively, ρ_j^1 is the number of atomic formulas in $P_i^1(\overline{y}_i^1)$ [8].

5. Partial deduction

During the process of partial deduction instead of the proof of $A(\overline{x}) \Rightarrow \exists \overline{y}_{\neq} B(\overline{y})$ we search such a maximal (up to the names of variables) sub-formula $B'(\overline{y}')$ of the formula $B(\overline{y})$ that $A(\overline{x}) \Rightarrow \exists \overline{y}'_{\neq} B'(\overline{y}')$.

Let a and a' be the numbers of atomic formulas in $A(\overline{x})$ and $A'(\overline{x}')$ respectively, m and m' be the numbers of objective variables in $A(\overline{x})$ and $A'(\overline{x}')$ respectively. Parameters q and r are defined as $q = a'/a$ and $r = m'/m$. In such a case sub-formula $A'(\overline{x}')$ is called a (q, r)-fragment of the formula $A(\overline{x})$.

Definition. *The problem of partial deducibility of a formula $B(\overline{y})$ from $A(\overline{x})$*

$$A(\overline{x}) \Rightarrow_P \exists \overline{y}_{\neq} B(\overline{y})$$

is the problem of extraction of such a maximal (upon q) (q, r)-fragment $Q(\overline{u})$ of the formula $B(\overline{y})$ that

$$A(\overline{x}) \Rightarrow \exists \overline{u}_{\neq} Q(\overline{u}).$$

It may be proved that if $Q(\overline{u})$ and $R(\overline{v})$ are two maximal sub-formulas of $A(\overline{x})$ and $B(\overline{y})$ obtained while checking

$$A(\overline{x}) \Rightarrow_P \exists \overline{y}_{\neq} B(\overline{y})$$

and

$$B(\overline{y}) \Rightarrow_P \exists \overline{x}_{\neq} A(\overline{x})$$

then $Q(\overline{u})$ and $R(\overline{v})$ coincide up to the names of variables.

That is there exists their common unifier $\lambda = |_{\overline{z}}^{\overline{u}} {}_{\overline{z}'}^{\overline{v}}$.

6. Algorithm of level description construction

The below described algorithm was offered in [11].

Let $A_1(\overline{x}_1), ..., A_K(\overline{x}_K)$ be elementary conjunctions which are components of class descriptions.

1. For every pair $A_i(\overline{x}_i)$ and $A_j(\overline{x}_j)$ $(i \neq j)$ extract their maximal common up to the names of variables sub-formula $Q_{i,j}^1(\overline{x}_{i,j}^1)$ and find unifiers $\lambda_{(i,j),i}$ and $\lambda_{(i,j),j}$.
2. Repeat the extraction of maximal common up to the names of variables sub-formula for every pair of already extracted sub-formulas $Q_{i_1 \dots i_{2^{l-1}}}^{l-1}(\overline{x}_{i_1 \dots i_{2^{l-1}}})$ ï£¡ $Q_{j_1 \dots j_{2^{l-1}}}^{l-1}(\overline{x}_{j_1 \dots j_{2^{l-1}}})$ and obtain their common sub-formulas $Q_{i_1 \dots i_{2^{l-1}}, j_1 \dots j_{2^{l-1}}}^{l}(\overline{x}_{i_1 \dots i_{2^{l-1}}, j_1 \dots j_{2^{l-1}}})$ $(l = 2, , L)$ and the unifiers.

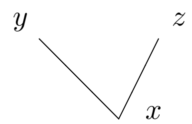
3. Select among the extracted sub-formulas $Q^l_{i_1 \dots i_{2^{l-1}}, j_1 \dots j_{2^{l-1}} l}(\overline{x}_{i_1 \dots i_{2^{l-1}}, j_1 \dots j_{2^{l-1}}})$ minimal ones and denote them by means of $P^1_i(\overline{y}^1_i)$ $(i = 1, , n_1)$. They are elementary conjunctions defining the first-level predicates $p^1_i(y^1_i)$ and the first-level variable y^1_i is the variable for the string of initial variables.
4. Sub-formulas of the higher levels $P^{l+1}_i(\overline{y}^{l+1}_i)$ $(i = 1, , n_{l+1},\ l = 2, , L)$ are constructed from the previously extracted sub-formulas $Q^l_{i_1 \dots i_l, j_1 \dots j_l}(\overline{x}_{i_1 \dots i_l, j_1 \dots j_l})$ with the substitution of $p^1_i(y^1_i)$ instead of $P^1_i(\overline{y}^1_i)$. Here y^1_i is the variable for the string of the less level variables.

The found unifiers are used here.

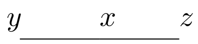
7. Example

The images for this example are taken from [4].

Let we must recognize contour images described by the following predicates.



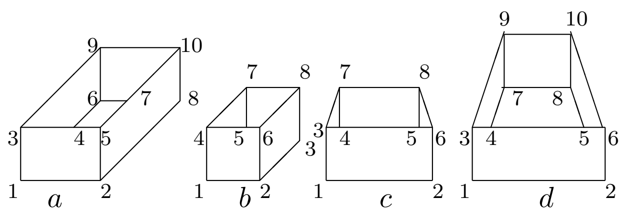
$V(x, y, z) \Longleftrightarrow (\angle yxz < \pi)$



$L(x, y, z,) \Longleftrightarrow x$ belongs to the segment $[y, z]$

Initial predicates.

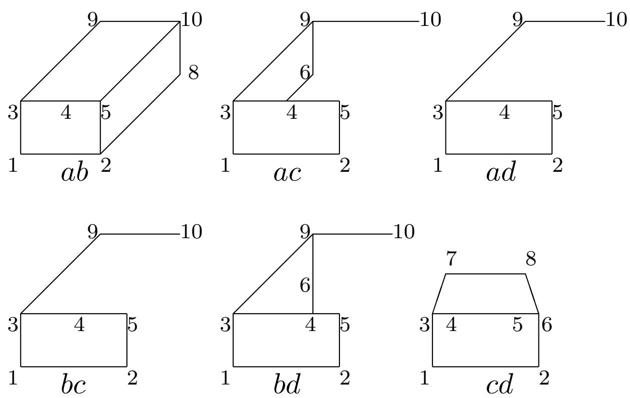
Given a set of contour images of "boxes" presented on the picture one can obtain the description of the class of "boxes" by means of changing the name of a node i by the variable x_i in the description of an object.



Training set

Given a complex image containing t nodes and not more than s occurrences of the same predicate in the image description, the number of steps needed for identification (and extraction) of a "box" is $O(t^{10})$ for an exhaustive algorithm and $O(s^{29})$ for a logical algorithm.

The first extraction of the common up to the names of variables subformulas gives 5 subformulas (subformulas corresponding to the images ad and bc coincide).



Images corresponding
to the extracted sub-formulas.

The second extraction of the common up to the names of variables subformulas gives 1 subformula. It defines the first-level sub-formula.

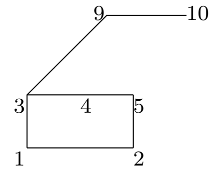


Image corresponding
to the first-level sub-formula.

This subformula contains 7 nodes and 10 relations between them. A new first-level variable x^1 for the string of variables $(x_1, x_2, x_3, x_4, x_5, x_9, x_{10})$ and a new first-level predicate p^1 such that $p^1(1, 2, 3, 4, 5, 9, 10)$ is true for the object a are introduced.

Images corresponding to the second-level sub-formulas are ab, ac, bd and cd. Their formulas have the first-level sub-formula which is changed by the first-level predicate. They have the variable x^1 and the initial variables

x_2, x_5, x_8, x_9, x_{10} for ab,

x_4, x_6, x_9, x_{10} for ac,

x_4, x_6,x_9, x_{10} for bd,

x_4, x_5, x_6, x_7, x_8 for cd.

At the same time the indicating the value of x^1 makes unknown only variables x_8, x_{10} for ab; x_6, x_{10} for ac; x_6, x_9, x_{10} for bd; x_4, x_8 or x_5, x_8 for cd.

Hence, these second-level sub-formulas contain respectively $m_{ab} = 3$, $m_{ac} = 3$, $m_{bd} = 3$ and $m_{cd} = 2$ essential variables (x^1 and some "old" ones).

Every of the second-level sub-formulas contain the first-level subformula $p^1(x^1)$ and some "old" atomic formulas. Their amounts are respectively $s_{ab} = 8$, $s_{ac} = 7$, $s_{bd} = 5$, $s_{cd} = 8$.

Elementary conjunctions corresponding the training set in the three-level descriptions contain one of the second-level subformulas $p^2_k(x^2_k)$ $(k = 1, ..., 6)$ and some "old" atomic formulas. Every of these formulas contain respectively $m_a = 4$, $m_b = 3$, $m_c = 2$ and $m_d = 4$ essential variables (x^2_k and some "old" ones).

The amounts of atomic formulas (with a second-level predicate and initial ones) are respectively $s_a = 8$, $s_b = 9$, $s_c = 5$, $s_d = 9$.

So the number of an exhaustive algorithm steps for the tree-level description is $O((t^3 + t^3 + t^3 + t^2) + (t^4 + t^3 + t^2 + t^4)) = O(t^4)$ instead of $O(t^{10})$ for the initial description.

The number of a logical algorithm steps for the tree-level description is $O((s^8 + s^7 + s^5 + s^8) + (s^8 + s^9 + s^5 + s^9)) = O(s^9)$ instead of $O(s^{29})$ for the initial description.

8. Discussion

The open question is ï£¡what extracted formula must be changed by an atomic one if it may be done in different ways?ï£¡ In the example above the formula corresponding to the image d contains both the subformula corresponding to the image bd and the siubformula corresponding to the image cd. What second-level predicate must appear in the tree-level description of d? To answer this question complexity investigation must be done.

While extracting a sub-formula it may happen that it contains several variables of a lower (not initial) level. In such a case the sub-formula defines a relation between parts of an object. If we must regard these parts as informative pair or a new informative part?

9. Conclusion

The use of predicate calculus language seems to be an adequate one for the simulation of Artificial Intelligence problems. But the NP-completeness of the problems appeared while such a simulation does not allow to implement algorithms directly.

The notion of partial deduction for a predicate formula allows to construct such a level description of classes that the exponent in the complexity upper bound of the problem solution decreases very much.

It does not mean that we can solve an NP-hard problem in a polynomial time. Because the construction of a level description is also an NP-hard problem with the almost same exponent in the complexity upper bound. It corresponds to the long time of learning and the quick implementation of the received knowledge.

10. Acknowledgment

The paper is published with financial support of RFBR grant 14-08-01276.

Bibliography

[1] T.M. Artyushkova (Kosovskaya) and A.V. Timofeev, *On one new approach to the forming of logical decision rules.* In: Vestnik of St.Petersburg University. 1985, No 8. P. 22 ï£¡– 29. (In Russian)

[2] D. Bugaychenko and D. Zubarevich, *Fast Pattern Recognition and Deep Learning Using Multi-Rooted Binary Decision Diagrams*, Lecture Notes in Artificial Intelligence, V. 8556, 2014. P. 73 —77.

[3] D.Z. Du and K.I. Ko, *Theory of Computational Complexity,* A Wiley-Interscience Publication. John Wiley & Sons, Inc. 2000.

[4] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, A Wiley-Interscience Publication, John Wiley & Sons, New York London Sydney Toronto, 1973.

[5] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman, New York (1979)

[6] T.M. Kossovskaya, *Proofs of the number of steps bounds for solving of some pattern recognition problems with logical description*, Vestnik Sankt-Peterburgskogo Universiteta. Seriya 1. 2007, No. 4, pp. 82–90. (In Russian)

[7] T. Kosovskaya, *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 — 99.

[8] T.M. Kossovskaya, *Level descriptions of classes for decreasing step number of pattern recognition problem solving described by predicate calculus formulas,* Vestnik SanktPeterburgskogo Universiteta. Seriya 10. 2008, No. 1, pp. 64 – 72. (In Russian)

[9] T.M. Kossovskaya. *Partial deduction of predicate formula as an instrument for recognition of an object with incomplete description,* Vestnik SanktPeterburgskogo Universiteta. Seriya 10. 2009, No. 1, pp. 74 – 84. (In Russian)

[10] *T.M. Kossovskaya*, Some artificial intelligence problems permitting formalization by means of predicate calculus language and upper bounds of their solution steps. //SPIIRAS Proceedings, 2010. No. 14, pp. 58 - 75. (In Russian)

[11] T.M. Kosovskaya. *Construction of Class Level Description for Efficient Recognition of a Complex Object,* International Journal ï£¡Information Content and Processingï£¡, Vol. 1,No 1. 2014. P. 92 ï£¡ 99.

[12] M.A. Lejeune, *Pattern definition of the p-efficiency concept*, ANNALS OF OPERATIONS RESEARCH V. 200, Issue 1, 2012. P. 23 — 36.

[13] N.J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*, McGRAW-HILL BOOK COMPANY, NEW YORK, 1971.

[14] S.J. Russel and P. Norvig, *Artificial Intelligence. A Modern Approach.* Pearson Education, Inc. 2003.

Authors' Information

Tatiana M. Kosovskaya *prof. of St. Petersburg State University,*
University av., 28, St. Petersburg, 198504, RUSSIA
Senior researcher of St. Petersburg Institute on Informatics and Automation of Russian Academy of Science, 14 line, 39, St. Petersburg, 199178, RUSSIA
E-mail: kosovtm@gmail.com