THE INVERSE MASLOV METHOD AND ANT TACTICS FOR EXHAUSTIVE SEARCH DECREASING

Tatiana Kosovskaya, Nina Petukhova

Abstract: *Algorithms for soling artificial intelligence problems allowing to formalize by means of predicate language are presented in the paper. These algorithms use the modified Maslov's inverse method. One of these algorithms is directly based on the inverse method, and the second one uses the inverse method and Ant algorithm's tactics. The model example using the described algorithms is shown.*

Keywords: *artificial intelligence, pattern recognition, predicate calculus, inverse method of S.Yu.Maslov, complexity theory, Ant algorithms.*

ACM Classification Keywords: *I.2.4 ARTIFICIAL INTELLIGENCE Knowledge Representation Formalisms and Methods – Predicate logic, I.5.1 PATTERN RECOGNITION Models – Deterministic, F.2.2 Nonnumerical Algorithms and Problems – Complexity of proof procedures.*

Introduction

NP-hardness of artificial intelligence problem, in particular, that allowing formalization by means of predicate calculus language [Kosovskaya, 2011] imposes the requirements to the algorithms for their solving. This requirements focused on optimize the sorting that occurs when searching for a suitable substitution, ensuring fulfillment of the target formula. Maslov's inverse method is developed to make more effective the procedure of deducibility in predicate calculus. An algorithm, which is a restriction of the inverse method to the proof of a special case formulas appearing in the decision of many artificial intelligence problems is described in [Kosovskaya, Petukhova, 2012]. A more detailed version of such an algorithm is considered in this paper. A model example of its application to the recognition of a contour image is described.

The use of multi agents involved in solving of the same problem simultaneously allows to find an analogy in the actions of a scout-ants of a real anthill. The use of this analogy in the literature has been called the Ant algorithm [Dorigo, Birattari, Stutzle, 2006]. This tactic allows to "parallelize" process of gathering information and to throw off obviously dead-end search solutions. It is suggested to apply the Ant tactic to the deduction search using the inverse method.

Initial definitions

The solving of many Artificial Intelligence problems permitting the use of predicate language may be reduced to the proof of a logical sequence in the form

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

where $\omega=\{a_1,\dots,a_k\}$ is a list of constants, $S(\omega)$ is a set of constant atomic formulas or their negations, $A(\overline{x})$ is an elementary conjunction of atomic formulas of the form $P_{k_i}(\overline{x})$ [Kosovskaya, 2011].

Such a logical sequence is equivalent to the truth of the formula

$$(\&\, S(\omega))\rightarrow \exists\overline{x}_{\neq}A(\overline{x})$$

for every value of constant ω. Using the fact that ω is an arbitrary constant set this formula may be reduced to the formula

$$\forall a_1,\dots,a_k\exists x_1,\dots,x_n\left(\underset{i=1}{\overset{\delta}{\&}}\left(\vee\neg S(a_1,\dots,a_k)\vee P_{k_i}(x_1,\dots,x_{n_i})\right)\right),$$

which is a particular case of a formula intended for the use of inverse Maslov method [Kosovskaya, Petukhova, 2012; Orevkov, 2003].

The formula (1) is deducible if and only if there exists such a substitution of the terms $t_1,\dots,t_{n_i}$ instead of variables $x_1,\dots,x_{n_i}$ that every elementary disjunction has a contrary pair. In the other words it is required to solve a system of equations in the form

$$i=\overline{1,\delta}\left\{\begin{array}{l} \overline{1,s}\left[\begin{array}{l}\{\\ \vdots\\ \{\end{array}\right.\\ \vdots\\ \overline{1,s}\left[\begin{array}{l}\{\\ \vdots\\ \left\{\begin{array}{l} t_1=a_1\\ \quad\vdots\\ t_{n_i}=a_{n_i}\end{array}\right.\\ \vdots\\ \{\end{array}\right.\\ \vdots\\ \overline{1,s}\left[\begin{array}{l}\{\\ \vdots\\ \{\end{array}\right.\end{array}\right.,$$

where *s* is the number of atomic formulas in $\neg S(a_1,\dots,a_k)$.

A solution of such a system may be found in an exponential number of steps. The inverse method is oriented on its essential decreasing. In the addition to the ideas of the inverse method it is suggested to order all the formulas $\vee\neg S(a_1,\dots,a_k)\vee P_{k_i}(x_1,\dots,x_{n_i})$ in (1) in the following way. As every formula $\vee\neg S(a_1,\dots,a_k)\vee P_{k_i}(x_1,\dots,x_{n_i})$ contains only one disjunctive number $P_{k_i}(x_1,\dots,x_{n_i})$ with variables then form groups with the same names $P_{k_i}(x_1,...,x_{n_i})$ and then order these groups according to their sizes. The same ordering must be done with the set $S(\omega)$.

First of all we search such a substitution which permits to assign variables in atomic formulas with the rarest predicate name. If such a substitution does not exist then the formula (1) is dot deducible.

If the substitution is found then it is made in all F-sets.

Repeat the procedure until all variables are changed by constants.

IMA – algorithm based on the inverse method

Definition. A list Γ of not repeated formulas in the form $\vee\neg S(a_1,...,a_k)\vee P_{k_i}(x_1,...,x_{n_i})$ is called an F-set for a formula (1) [Kosovskaya, Petukhova, 2012; Orevkov, 2003].

Definition. An F-set is called an empty one if all its formulas do not have variables and are tautological ones. [Kosovskaya, Petukhova, 2012; Orevkov, 2003].

Definition. An F-set is called a deadlock one if it contains at least one false formula without variables or a formula which is not a tautology nor a contradiction. [Kosovskaya, Petukhova, 2012]

IMA-algorithm

1. Construct an F-set corresponding to the formula (1). I.e. write down δ elementary disjunctions of the form $\vee\neg S(a_1,...,a_k)\vee P_{k_i}(x_1,...,x_{n_i})$.

2. Assign all variables in the following way:

2.1. Cancel all marks about deleting of a predicate formula from $S(\omega)$.

2.2. Consider a predicate formula $\vee\neg S(a_1,...,a_k)\vee P_{k_i}(t_1,...,t_{n_i})$ from the F-set containing such an atomic formula $P_{k_i}(t_1,...,t_m)$ that the list $t_1,...,t_m$ has at least one variable.

2.3. Check if $S(\omega)$ contains $\neg P_{k_i}(v_1,...,v_m)$ for some constants $v_1,...,v_m$. If it is so then mark $\neg P_{k_i}(v_1,...,v_m)$ as deleted and go to 2.4. Otherwise go to 3.

2.4. Solve the system of equations identifying the list of variables and constants $t_1,...,t_m$ with the list of constants $v_1,...,v_m$. If the system has a solution[1], then go to 2.5. Otherwise go to 2.3.

2.5. Substitute the values received in point 2.4 instead the variables of $t_1,...,t_m$ into all formulas in the F-set.

2.6. Delete repetitions of formulas in the received F-set.

2.7. If the received F-set is empty them the algorithm halts.

2.8. If the received F-set is a deadlock one then go to 3.

[1] Here the system may don't have a solution if some value for a variable is yet assigned to the other variable.

2.9. If all formulas containing variables is marked as deleted then go to 4. Otherwise go to 3.

3. Cancellation of assignments.

3.1. Cancel the last action of point 2.5 (if it is possible) and go to 2.3.

3.2. If cancellation of the last action of point 2.5 is not possible then mark $P_{k_i}(t_1,\ldots,t_m)$ as deleted and go to 2.

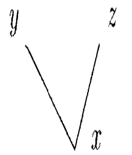
4. If all formulas in the f-set are marked as deleted then the formula is not deductible. The algorithm halts.

An upper bound of a similar algorithm is presented in [Kosovskaya, Petukhova, 2012]. It is O(s^{δ}) where s is the number of atomic formulas in $S(\omega)$. The upper bound of the presented algorithm is a similar one.

A model example

Show that in spite of a high upper bound of the IMA algorithm number of steps real number of steps is rather smaller.

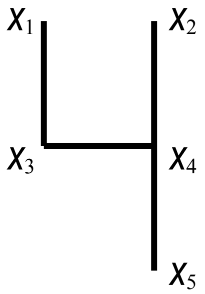
Let a set of contour images is done. It is described by means of the following predicates.



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

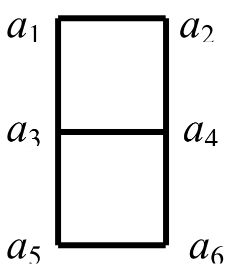
$L(x,y,z) \leftrightarrow "x\ belongs\ to\ the\ \mathrm{int}reval\ with\ the\ ends\ y\ and\ z"$

We have the class of images of the number four



which has the description $A(x_1,\ldots,x_5) = V(x_3,x_1,x_4)\ \&\ V(x_4,x_2,x_3)\ \&\ V(x_4,x_3,x_5)\ \&\ L(x_4,x_2,x_5)$.

It is required to extract an image of the number four from the image



This image has a description $S(a_1,\dots,a_6) = V(a_1,a_2,a_3) \& V(a_2,a_1,a_4) \& V(a_3,a_1,a_4) \& V(a_3,a_4,a_5) \& L(a_3,a_1,a_5) \& V(a_4,a_2,a_3) \& V(a_4,a_3,a_6) \& L(a_4,a_2,a_6) \& V(a_5,a_3,a_6) \& V(a_6,a_4,a_5)$.

An estimate of complexity for this example is

$$O(10^4) = O(10000).$$

To prove the belonging of a part of the image to the class of "number four" it is needed to prove deducibility of the formula

$$S(a_1,\dots,a_6) \Rightarrow \exists (x_1,\dots,x_5)_{\neq} A(x_1,\dots,x_5).$$

After reducing to the form (1) we have

$$\forall a_1,\dots,a_k \exists x_1,\dots,x_n \left(\begin{array}{l} \left(\begin{array}{c} V(x_3,x_1,x_4) \vee \neg V(a_1,a_2,a_3) \vee \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \\ \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \\ \vee \neg V(a_6,a_4,a_5) \vee \neg L(a_3,a_1,a_5) \vee \neg L(a_4,a_2,a_6) \end{array} \right) \& \\ \left(\begin{array}{c} V(x_4,x_2,x_3) \vee \neg V(a_1,a_2,a_3) \vee \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \\ \neg V(a_3,a_4,a_5) \vee \neg L(a_3,a_1,a_5) \vee \neg V(a_4,a_2,a_3) \vee \neg V(a_4,a_3,a_6) \vee \\ \vee \neg V(a_6,a_4,a_5) \vee \neg L(a_3,a_1,a_5) \vee \neg L(a_4,a_2,a_6) \end{array} \right) \& \\ \left(\begin{array}{c} V(x_4,x_3,x_5) \vee \neg V(a_1,a_2,a_3) \vee \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \\ \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \\ \vee \neg V(a_6,a_4,a_5) \vee \neg L(a_3,a_1,a_5) \vee \neg L(a_4,a_2,a_6) \end{array} \right) \& \\ \left(\begin{array}{c} L(x_4,x_2,x_5) \vee \neg V(a_1,a_2,a_3) \vee \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \\ \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \\ \vee \neg V(a_6,a_4,a_5) \vee \neg L(a_3,a_1,a_5) \vee \neg L(a_4,a_2,a_6) \end{array} \right) \end{array} \right)$$

The next F-set we have the result of point 1. It was received in 4 steps.

$$\left(\begin{array}{l} \left(\begin{array}{c} V(x_3,x_1,x_4) \vee \neg V(a_1,a_2,a_3) \vee \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \\ \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \neg V(a_6,a_4,a_5) \vee \neg L(a_4,a_2,a_6) \vee \neg L(a_3,a_1,a_5) \end{array} \right) \\ \left(\begin{array}{c} V(x_4,x_2,x_3) \vee \neg V(a_1,a_2,a_3) \vee \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \\ \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \neg V(a_6,a_4,a_5) \vee \neg L(a_4,a_2,a_6) \vee \neg L(a_3,a_1,a_5) \end{array} \right) \\ \left(\begin{array}{c} V(x_4,x_3,x_5) \vee \neg V(a_1,a_2,a_3) \vee \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \\ \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \neg V(a_6,a_4,a_5) \vee \neg L(a_4,a_2,a_6) \vee \neg L(a_3,a_1,a_5) \end{array} \right) \\ \left(\begin{array}{c} L(x_4,x_2,x_5) \vee \neg V(a_1,a_2,a_3) \vee \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \\ \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \neg V(a_6,a_4,a_5) \vee \neg L(a_4,a_2,a_6) \vee \neg L(a_3,a_1,a_5) \end{array} \right) \end{array} \right)$$

Point 2.1 is not applicable. According to the point 2.2 take the formula $V(x_3,x_1,x_4)$, according to the point 2.3 take the formula $\neg V(a_1,a_2,a_3)$ (+1 step) and mark it as a deleted one (+1 step). According to the point 2.4 solve the system of equations

$$\begin{array}{l} x_3 = a_1 \\ x_1 = a_2 \\ x_4 = a_3 \end{array} \text{ (+3 steps).}$$

According to the point 2.5 F-set has the form

$$\begin{pmatrix} \begin{pmatrix} V(a_1,a_2,a_3)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\neg V(a_2,a_1,a_4)\vee\neg V(a_3,a_1,a_4)\vee\neg V(a_3,a_4,a_5)\vee\neg V(a_4,a_2,a_3)\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{pmatrix} \\ \begin{pmatrix} V(a_3,x_2,a_1)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\neg V(a_2,a_1,a_4)\vee\neg V(a_3,a_1,a_4)\vee\neg V(a_3,a_4,a_5)\vee\neg V(a_4,a_2,a_3)\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{pmatrix} \\ \begin{pmatrix} V(a_3,a_1,x_5)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\neg V(a_2,a_1,a_4)\vee\neg V(a_3,a_1,a_4)\vee\neg V(a_3,a_4,a_5)\vee\neg V(a_4,a_2,a_3)\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{pmatrix} \\ \begin{pmatrix} L(a_3,x_2,x_5)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\neg V(a_2,a_1,a_4)\vee\neg V(a_3,a_1,a_4)\vee\neg V(a_3,a_4,a_5)\vee\neg V(a_4,a_2,a_3)\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{pmatrix} \end{pmatrix}$$

(+12 steps)

According to the point 2.9 (+4 steps) return to the points 2.2 and 2.3. The formula $V(a_4,x_2,a_2)$ does not have a pair for unification (+22 steps), that is why go to 3 (+9 steps). F-set differs from the initial one only by marking the formula $\neg V(a_2,a_1,a_4)$ as a deleted one. According to the point 2.2 take the formula $V(x_3,x_1,x_4)$, according to the point 2.3 take the formula $\neg V(a_3,a_1,a_4)$ (+2 steps) and mark it as a deleted one (+1 step).

According to the point 2.4 solve the system of equations

$$\begin{aligned} x_3 &= a_3 \\ x_1 &= a_1 \quad \text{(+3 steps).} \\ x_4 &= a_4 \end{aligned}$$

According to the point 2.5 F-set has the form

$$\begin{pmatrix} \begin{pmatrix} V(a_3,a_1,a_4)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\neg V(a_4,a_2,a_3)\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{pmatrix} \\ \begin{pmatrix} V(a_4,x_2,a_3)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\neg V(a_4,a_2,a_3)\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{pmatrix} \\ \begin{pmatrix} V(a_4,a_3,x_5)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\neg V(a_4,a_2,a_3)\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{pmatrix} \\ \begin{pmatrix} L(a_4,x_2,x_5)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\neg V(a_4,a_2,a_3)\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{pmatrix} \end{pmatrix}$$

(+12 steps)

According to the point 2.9 (+4 steps) return to the points 2.2. Take the formula $V(a_4,x_2,a_3)$ and according to the point 2.3 take the negation $V(a_4,x_2,a_3)$ (+9 steps) and mark it as a deleted one (+1 step). According to the point 2.4 solve the system of equations

$$x_2 = a_2 \text{ (+1 step)}$$

According to the point 2.5 F-set has the form

$$\left(\begin{array}{c}\left(\begin{array}{c} V(a_3,a_1,a_4)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\boxed{\neg V(a_4,a_2,a_3)}\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{array}\right) \\ \left(\begin{array}{c} V(a_4,a_2,a_3)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\boxed{\neg V(a_4,a_2,a_3)}\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{array}\right) \\ \left(\begin{array}{c} V(a_4,a_3,x_5)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\boxed{\neg V(a_4,a_2,a_3)}\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{array}\right) \\ \left(\begin{array}{c} L(a_4,a_2,x_5)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\boxed{\neg V(a_4,a_2,a_3)}\vee \\ \neg V(a_4,a_3,a_6)\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{array}\right)\end{array}\right)$$

(+4 steps)

According to the point 2.9 (+4 steps) return to the points 2.2. Take the formula $V(a_4,a_3,x_5)$ and according to the point 2.3 take the formula $\neg V(a_4,a_3,a_6)$ (+10 steps) and mark it as a deleted one (+1 step). According to the point 2.4 solve the system of equations

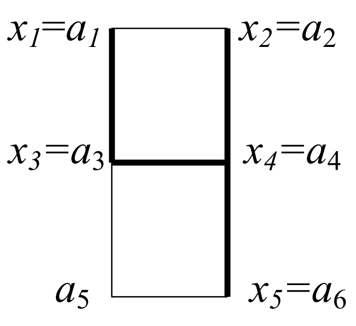
$x_5 = a_6$ (+1 step).

According to the point 2.5 F-set has the form

$$\left(\begin{array}{c}\left(\begin{array}{c} V(a_3,a_1,a_4)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\boxed{\neg V(a_4,a_2,a_3)}\vee \\ \boxed{\neg V(a_4,a_3,a_6)}\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{array}\right) \\ \left(\begin{array}{c} V(a_4,a_2,a_3)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\boxed{\neg V(a_4,a_2,a_3)}\vee \\ \boxed{\neg V(a_4,a_3,a_6)}\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{array}\right) \\ \left(\begin{array}{c} V(a_4,a_3,a_6)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\boxed{\neg V(a_4,a_2,a_3)}\vee \\ \boxed{\neg V(a_4,a_3,a_6)}\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{array}\right) \\ \left(\begin{array}{c} L(a_4,a_2,a_6)\vee\boxed{\neg V(a_1,a_2,a_3)}\vee\boxed{\neg V(a_2,a_1,a_4)}\vee\boxed{\neg V(a_3,a_1,a_4)}\vee\neg V(a_3,a_4,a_5)\vee\boxed{\neg V(a_4,a_2,a_3)}\vee \\ \boxed{\neg V(a_4,a_3,a_6)}\vee\neg V(a_5,a_3,a_6)\vee\neg V(a_6,a_4,a_5)\vee\neg L(a_4,a_2,a_6)\vee\neg L(a_3,a_1,a_5) \end{array}\right)\end{array}\right)$$

(+ 2 steps)

The empty F-set is received. According to point 2.7 the algorithm halts (+4 steps). An object “four” is extracted. The extracting needed 166 steps.



This example shows us the necessity of ordering formulas within F-set. If we order formulas before the algorithm run (as it was mentioned at the begin of the paper) then the initial formula would have the form

$$\forall a_1,\ldots,a_k \exists x_1,\ldots,x_n \left(\begin{array}{c} \left(\begin{array}{c} L(x_4,x_2,x_5) \vee \neg L(a_3,a_1,a_5) \vee \neg L(a_4,a_2,a_6) \vee \neg V(a_1,a_2,a_3) \vee \\ \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \\ \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \neg V(a_6,a_4,a_5) \end{array} \right) \& \\ \left(\begin{array}{c} V(x_3,x_1,x_4) \vee \neg L(a_3,a_1,a_5) \vee \neg L(a_4,a_2,a_6) \vee \neg V(a_1,a_2,a_3) \vee \\ \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \\ \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \neg V(a_6,a_4,a_5) \end{array} \right) \& \\ \left(\begin{array}{c} V(x_4,x_2,x_3) \vee \neg L(a_3,a_1,a_5) \vee \neg L(a_4,a_2,a_6) \vee \neg V(a_1,a_2,a_3) \vee \\ \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \\ \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \neg V(a_6,a_4,a_5) \end{array} \right) \& \\ \left(\begin{array}{c} V(x_4,x_3,x_5) \vee \neg L(a_3,a_1,a_5) \vee \neg L(a_4,a_2,a_6) \vee \neg V(a_1,a_2,a_3) \vee \\ \neg V(a_2,a_1,a_4) \vee \neg V(a_3,a_1,a_4) \vee \neg V(a_3,a_4,a_5) \vee \neg V(a_4,a_2,a_3) \vee \\ \neg V(a_4,a_3,a_6) \vee \neg V(a_5,a_3,a_6) \vee \neg V(a_6,a_4,a_5) \end{array} \right) \end{array} \right).$$

In such a case the problem would be solved after one execution of all algorithm steps without cancellation ones. We have not ordered the F-set for more clearness.

Ant tactics

An ant algorithm [Dorigo, Birattari, Stutzle, 2006] is sufficiently new approach to the solving of Artificial Intelligence problems. It reflexes actions of insects. Behavior of ants is the basis of a series of methods the most successful of which is optimization of ant colony.

Optimization of ant colony simulates actions of some kinds of ants. They put pheromone in the ground in order to mark successful ways for other colony members moving. More ants use the same way more the pheromone concentration on this way. More the pheromone concentration is on the way more preferable this way is in comparison with the other ones. In such a way an “ant logic” allows to choose a shorter way between two points.

Ant algorithms are iteration ones. Every iteration takes into account actions of artificial ants. Every ant constructs its decision during making some actions and does not repeat the same action. For every step an ant chooses the next action in dependence of pheromone quantity used for marking the actions before their fulfilling.

Ant algorithms successfully solve some NP-hard problems, for example, the problem of “traveling salesman”.

Idea of an algorithm using the ant tactics

Artificial ant is a program agent using to solve some problem and being a member of a big colony of artificial ants. Any ant has a set of simple rules which allow him to choose an action. It has a list of taboos, i.e. a list of actions which it has already done or cannot do at all.

A real ant put some pheromone while moving along the way. Artificial ant increases a mark of already fulfilled action.

The population of artificial ants distributes the actions between themselves in equal parts. Such a distribution in equal parts is necessary because every action may be the first.

In the problem under consideration it is needed to prove or to refute deducibility of a predicate formula. The most difficult stage of such a proving is to find a list of distinct values for variables $\overline{x}$ the existence of which is claimed by the formula. So the ant tactic will be used by means of decreasing the action mark in the following way.

Initially the mark of formulas $P(t_1,\ldots,t_m)$ and $\neg P(a_1,\ldots,a_m)$ unification equals 1. If the unification is possible and does not lead to a deadlock F-set then the mark their unification increases by 1. If a deadlock F-set is received then the mark becomes 0. Otherwise it decreases by 1.

Every artificial ant begins its actions with its own disjunct using rules of IMA algorithm. While assigning variables ants connect each other and compare results. Comparison of different ants results consists in the checking of non-contradictoriness of these results. The results of ant actions are contradictory if they assign the same variables with different values or the same value is assigned to different variables. If the result of two ants actions does not contradict each other then the assignment is fulfilled in the disjunctions of the both ants.

If an ant cannot assign any variable then this ant does not make any action further.

If deducibility of the formula is proved or all marks are 0, then the algorithm halts.

Bibliography

[Dorigo, Birattari, Stutzle, 2006] M. Dorigo, M. Birattari, T. Stutzle. Ant Colony Optimization. Artificial Ants as a Computational Intelligence Technique. In: IRIDIA – Technical Report Series, 2006, № 23, pp 1-2.

[Kosovskaya, 2011] T.Kosovskaya. Discrete Artificial Intelligence Problems and Number of Steps of their Solution. In: International Journal on Information Theories and Applications, Vol. 18, Number 1, 2011. P. 93 – 99.

[Kosovskaya, Petukhova, 2012] T.Kosovskaya, N.Petukhova. The Inverse Method for Solving Artificial Intelligence Problems in the Frameworks of Logic-Objective Approach and Bounds of its Number of Steps. In: International Journal «Information Models and Analyses», Vol. 1. 2012. P. 84-93

[Orevkov, 2003] Orevkov V.P. The inverse method of logical derivation In: Adamenko A. Kuchukov A. Programming Logic and Visual Prolog. St. Petersburg, "`BHV-Petersburg"'. (2003), pp.. 952-965. (in Russian)

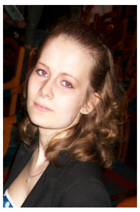
Acknowledgement

The paper is published with financial support of the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Authors' Information

Tatiana Kosovskaya – *Dr., Senior researcher, St.Petersburg Institute of Informatics and Automation of Russian Academy of Science, 14 line, 39, St.Petersburg, 199178, Russia; Professor of St.Petersburg State Marine Technical University, Lotsmanskaya ul., 3, St.Petersburg, 190008, Russia; Professor of St.Petersburg State University, University av., 28, Stary Petergof, St.Petersburg, 198504, Russia, e-mail: kosov@NK1022.spb.edu*

Major Fields of Scientific Research: Logical approach to artificial intelligence problems, theory of complexity of algorithms.



Nina Petukhova – *PHD student, St.Petersburg State Marine Technical University, Lotsmanskaya ul., 3, St.Petersburg, 190008, Russia, e-mail: ninka_mat@mail.ru*

Major Fields of Scientific Research: Logical approach to Artificial Intelligence problems.