MULTI-AGENT DESCRIPTION OF AN OBJECT BY MEANS OF A PREDICATE CALCULUS LANGUAGE

Tatiana Kosovskaya

***Abstract**: A problem of multi-agent description of a complex object is under consideration. It is supposed that an object may be described in the terms of its parts properties and relations between these parts. But every agent has only a part of the object description and does not know the true names of elements and gives them names arbitrary. An algorithm solving the posed problem is described and the upper bound of its steps is proved.*

***Keywords**: multi-agent description of an object, predicate calculus, partial deduction, computational complexity of an algorithm.*

***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

The use of a predicate calculus language for the Artificial Intelligence problems was proposed in the middle of the XX century (see, for example, [Duda, Nilsson]) and up to now is offered in theoretical papers [Russel]. Just a predicate calculus language allows adequately describe a complex object characterizing by properties of its parts and relations between them.

Nevertheless the difficulty of a practical implementation of such an approach is connected with NP-hardness of many problems appearing in the frameworks of this approach. The proven upper bound of an algorithm solving such a problem allows for every problem to find such restrictions upon input data which essentially decrease the number of the algorithm run steps. That's why upper the upper bound of the offered algorithm number of steps are proved bellow.

A problem of multi-agent description of a complex object is under consideration in the presented paper. It is supposed that every agent has only a part of an investigated object description. Moreover, she does not know the true names of elements and gives them names arbitrary. It is similar to the parable about tree blind men who feel an elephant. To overcome such a paradox, it is supposed that every two agents have information concerning some common part of an object. The main difficulty in this problem is to find and identify these parts.

Multi-agent description problem setting

Let an investigated object is represented as a set of its elements $\omega = \{\omega_1, ..., \omega_t\}$ and is characterized by the set of predicates $p_1, ..., p_n$, every of which is defined on the elements of ω and gives properties of these elements and relations between them.

Information (description) of the object is an elementary conjunction of atomic formulas with predicates $p_1, ..., p_n$.

There are *m* agents $a_1, ..., a_m$ which can measure some values for some predicates of some elements of ω. The agent a_j does not know the true number of the ω elements and suppose that she deals with the object $\omega^j = \{\omega_1^j, ..., \omega_{tj}^j\}$. That is the agent a_j has the information $I_j(\omega_1^j, ..., \omega_{tj}^j)$ in the form of elementary conjunction of atomic formulas.

It is required to construct the description of ω $I(\omega_1, ..., \omega_t)$.

As every agent uses her own notifications for the names of the object elements, it is needed to find all common up to the names of arguments sub-formulas of the informations $I_j(\omega_1^j, ..., \omega_{tj}^j)$ ($j = 1, ..., m$) and their unifiers, i.e. such substitutions for the argument names that the extracted pairs of sub-formulas are identical.

For example, two formulas $A = p(a,b) \& p(b,a) \& q(b,a,c)$ and $B = p(b,a) \& p(b,d) \& q(a,b,d)$ have common up to the names of arguments sub-formula. If one substitute arguments *u, v, w* instead of the constants
a, b, c in the formula A and substitute arguments v, *u, w* instead of the constants *a, b, d* in the formula B we receive formulas $C1 = p(u,v) \& p(v,u) \& q(v,u,w)$ and $C2 = p(u,v) \& p(u,w) \& q(v,u,w)$ respectively. The formula $C = p(u,v) \& q(v,u,w)$ is a common up to the names of arguments sub-formula of *A* and *B* with unifiers λ_{CA} (substitution of *u, v, w* instead of *a, b, c*) and λ_{CB} (substitution of v, *u, w* instead of *a, b, d*).

A notion of partial deduction (introduced in [Kosovskaya, 2009]) allows to extract a maximal common up to the names of arguments sub-formula of *A* and *B* and to find their unifiers. During the process of partial deduction instead of the proof of $A(\boldsymbol{\omega}) \Rightarrow B(\boldsymbol{\tau})$ we search such a maximal (up to the names of arguments) sub-formula $B'(\mathbf{x}')$ of the formula $B(\mathbf{x})$ that $A(\omega) \Rightarrow \exists \mathbf{x}'_{\neq} B'(\mathbf{x}')$. In the process of this sequent proof the unifier of $B'(\mathbf{x}')$ and a sub-formula of $A(\boldsymbol{\omega})$ will be found. The notation $A(\boldsymbol{\omega}) \Rightarrow_P$

□$B(\tau)$ will be used for the partial deduction process. A detailed description of the partial deduction checking and the search of the unifier is in [Kosovskaya, 2014].

Algorithm of multi-agent description

Let every agent a_j has information I_j about the described object ω ($j = 1, \ldots, m$). To constract a description of ω the following algorithm is offered.

1. Change all constants in $I_1, \ldots, I_m$ by variables in such a way that different constants are changed by different variables and the names of variables in I_i and I_j ($i \neq j$) does not coinside.
2. For every pair of elementary conjunctions I_i and I_j ($i = 1, \ldots, m-1, j = i+1, \ldots, m$) check partial deduction $I_i \Rightarrow_P I_j$ with the extraction of their common up to the names of arguments sub-formula C_{ij} and unifiers $\lambda_{i,ij}$ and $\lambda_{j,ij}$. Every argument of C_{ij} has a unique name.
3. For every pair i and j ($i > j$) check if I_i and I_j contain a contradictory pair of atomic formulas or two sub-formulas which can not be satisfiable simultaneously (for example, "*x is green*" and "*x is red*"). If such a contradiction is established then delete from C_{ij} atomic formulas containing the variables which are in the contradictory sub-formulas. Change the unfiers.
4. For every i identify the variables in C_{ij} ($i \neq j$) which are substituted in I_i and I_j instead of the same variable. The names of the identified variables are changed in unifiers by the same name.
5. With the use of the unifiers obtained in 2 – 4 change the names of variables in $I_1, \ldots, I_m$.
6. Write down the conjunction $I_1 \& \ldots \& I_m$ and delete the repeating atomic formulas.

Upper bound of the algorithm run steps

To estimate the number of the algorithm run steps we estimate every item of the algorithm.

1. The change all constants in $I_1, \ldots, I_m$ requires not more than $\sum_{j=1}^{m} ||I_j||$ «steps».
2. The checking of partial deduction $I_i \Rightarrow_P I_j$ requires $O(t_i^{t_j} \cdot 2^{||I_i||})$ «steps» for an exhaustive algorithm and $O(||I_i||^{||I_j||} \cdot ||I_i||^3)$ «steps» for an algorithm based on the derivation in the predicate calculus. (These estimates are proved in [Kosovskaya, 2011].) It is needed to summarize the above estimates for $i = 1, \ldots, m-1$, $j = i, \ldots, m$. So we have $O(t^t\, 2^{||I||} m^2)$ «steps» for an exhaustive algorithm and $O(||I||^{||I||+3}\, m^2)$ «steps» for an algorithm based on the derivation in the predicate calculus. Here t and $||I||$ are respectively the maximal numbers of variables and atomic formulas in I_j ($j = 1, \ldots, m$).
3. Consistency checking of the formulas I_i and I_j requires $||I_i||\ ||I_j||$ «steps». This item of the algorithm requires not more than $\sum_{i=1}^{m} (m - i)\, ||I_i||$ «steps» that is $O(m^2\, ||I||)$ «steps».
4. For every i identification of the variables in C_{ij} ($i > j$) consists in the comparison of the replaced part of the unifiers $\lambda_{i,ij}$ and $\lambda_{j,ij}$. It requieres not more than $(m - i)t_i^2$ «steps». Summarizing it for $i = 1, \ldots, m$ we have not more than $\sum_{i=1}^{m} (m - i)t_i^2 = O(m^2 t^2)$ «steps».

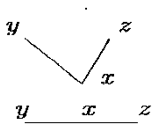
5. The number of «steps» required for the changing of the names of variables in $I_1, \ldots, I_m$ is linear under $\sum_{i=1}^{m} ||I_i|| = O(m\ ||I||)$ «steps».
6. The number of «steps» required for the deleting of the repeated conjunctive terms is not more than $\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} ||I_i||\ ||I_j|| = O(m^2\ t^2)$ «steps».

The number of the algorithm run steps is $O(t^t\ m^2\ ||I||^2)$ for an exhaustive algorithm and $O(||I||^{||I||+3}\ m^2)$ for an algorithm based on the derivation in the predicate calculus.

The analysis of the received estimation shows that the main contribution to it is made by the summarized number of partial deduction checking (item 2).

Examples

Consider an example of description of a contour image of a “box” by 3 agents in the terms of three predicates V and L represented on the Figure 1.[1]



$V(x,y,z) \Leftrightarrow$ “$\angle yxz < \pi$”

$L(x,y,z) \Leftrightarrow$ “*x belongs the segment (y,z)*”

Figure 1. Initial predicates

These predicates characterize the position of the node *x* relatively the nodes *y*, *z* and have the following properties: $V(x,y,z) \Leftrightarrow \neg V(x,z,y) \vee L(x,y,z)$ (and hence $V(x,y,z)\ \&\ \neg V(x,z,y)$ is a contradiction) and $L(x,y,z) \Leftrightarrow L(x,z,y)$.

Every agent has a description of one of the fragment represented on the Figure 2.

---

[1] The extracting of the maximal common up to the names of arguments sub-formula of two elementary conjunctions, the search of their unifier and expression the formulas through the extracted sub-formulas was made with the use of a software support implemented by a student of faculty of mathematics and mechanics of Sankt-Petersburg State University Petrov D.A.

As the arguments in the descriptions of these fragments are very important, they are written down explicitly.

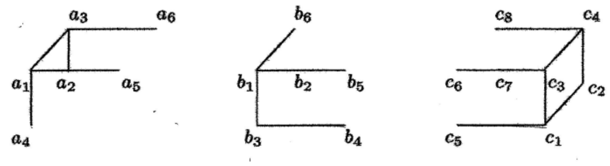


Figure 2. Fragments of the image received by three agents

According to the item 1 of the algorithm all constants in the fragment descriptions are replaced by variables in such a way that different constants are changed by different variables and the names of variables in I_i and I_j ($i \neq j$) does not coincide. After this the fragment descriptions have the form:

I1(x1,...,x6) = V(x1,x2,x4) & V(x1,x5,x4) & V(x1,x3,x2) & V(x1,x3,x5) & V(x1,x3,x4) & V(x2,x1,x3) & V(x2,x3,x5) & V(x3,x2,x1) &V(x3,x6,x2) & V(x3,x6,x1) & L(x2,x1,x5),
I2(y1,...,y6) = V(y3,y1,y4) & V(y1,y2,y3) & V(y1,y5,y3) & V(y1,y6,y2) & V(y1,y6,y5) & V(y1,y6,y3) & L(y2,y1,y5),
I3(z1,...,z8) = V(z1,z5,z3) & V(z1,z3,z2) & V(z1,z5,z2) & V(z3,z1,z7) & V(z3,z1,z6) & V(z3,z7,z4) & V(z3,z6,z4) & V(z3,z4,z1) & V(z4,z2,z3) & V(z4,z3,z8) & V(z4,z2,z8) & L(z7,z6,z3).

According to the item 2 of the algorithm check pairwise partial deduction $I_i \Rightarrow_P I_j$.

Maximal common up to the names of arguments sub-formula of *I1(x1,...,x6)* and *I2(y1,...,y6)* is *C12(u0,...,u4)* in the form

C12(u0,...,u4) = V(u0,u1,u2) & V(u0,u3,u2) & V(u0,u4,u1) & V(u0,u4,u3) & V(u0,u4,u2) & L(u1,u0,u3).

It has unifiers $\lambda_{I1,C12}$ – substitution of *u0, u1, u4, u2, u3* instead of *x1, x2, x3, x4, x5* respectively and $\lambda_{I2,C12}$ – substitution of *u0, u1, u2, u3, u4* instead of *y1, y2, y3, y5, y6* respectively. Besides,

I1(u0,u1,u2,u3,u4,x6) =

= V(u1,u0,u4) & V(u1,u4,u3) & V(u4,u1,u0) & V(u4,x6,u1) & V(u4,x6,u0) & C12(u0, ..., u4),
I2(u0,u1,u2,y4,u3,u4) = V(u2,u0,y4) & C12(u0,...,u4).

Maximal common up to the names of arguments sub-formula of *I2(y1,...,y6)* and *I3(z1,...,z8)* is *C23(v0,v2,v4,v5,v6,v7)* in the form

C23(v0,v2,v4,v5,v6,v7) = V(v6,v2,v7) & V(v2,v4,v6) & V(v2,v5,v6) &V(v2,v0,v4) & V(v2,v0,v5).

It has unifiers $\lambda_{I2,C23}$ *– substitution of v2, v4, v6, v7, v5, v0 instead of y1, y2, y3, y4, y5, y6* respectively *and* $\lambda_{I3,C23}$ – *substitution of v0, v2, v6, v5, v4, v7 instead of z1, z3, z5, z6, z7, z8* respectively. Besides,

I2(v2,v4,v6,v7,v5,v0) = V(v2,v0,v6) & L(v4,v2,v5) & C23(v0,v2,v4,v5,v6,v7),
I3(v0,z2,v2,v6,z5,v5,v4,v7) =

= V(v2,v6,v0) & V(v0,z5,v2) & V(v0,v2,z2) & V(v0,v5,z2) & V(v6,z2,v2) & V(v6,v2,v7) & L(v4,v5,v2) &

& C23(v0,v2,v4,v5,v6,v7).

As *I2(v2,v4,v6,v7,v5,v0)* contains *V(v2,v0,v6)* and I3(v0,z2,v2,v6,z5,v5,v4,v7) contains V(v2,v6,v0) and according to the definition of the predicate *V* the formula *V(x,y,z) & V(x,z,y)* is a contradiction so substitutions with this unifiers can not give a consistent description of the object. After deleting from *I2(y1,...,y6)* and *I3(z1,...,z8)* the variables *y1* and *z3* respectively a new maximal common up to the names of arguments their sub-formula *C'23(v0,v2,v4,v5,v6,v7)* in the form

C'23(v0,v1,v2) = L(v1,v0,v2)

will be received with the unifiers $\lambda_{I2,C'23}$ – substitution *of v0, v1, v2* instead of *y1, y2, y3* respectively and $\lambda_{I3,C'23}$ – substitution of *v2, v0, v1* instead of *z3, z6, z7* respectively. Besides,

I2(v0,v1,v2,y4,y5,y6) =

= V(v2,v0,y4) & V(v0,v1,v2) & V(v0,y5,v2) & V(v0,y6,v1) &V(v0,y6,y5) & V(v0,y6,v2) & C'23(v0,v1,v2),
I3(z1,z2,v2,z4,z5,v0,v1,z8) =

= V(z1,z5,v2) & V(z1,v2,z2) & V(z1,z5,z2) & V(v2,z1,v1) & V v2,z1,v0) & V(v2,v1,z4) & V(v2,v0,z4) & V(v2,z4,z1) & V(z4,z2,v2) & V(z4,v2,z8) & V(z4,z2,z8) & C'23(v0,v1,v2).

Maximal common up to the names of arguments sub-formula of *I1(x1,...,x6)* and *I3(z1,...,z8)* is *C13(w0, ...,w6)* in the form

C13(w0, ...,w6) = V(w2,w4,w6) & V(w2,w5,w6) & V(w2,w0,w4) & V(w2,w0,w5) & V(w0,w1,w2).

It has unifiers $\lambda_{I1,C13}$ *– substitution of w2, w4, w0, w6, w5, w6 instead of x1, x2, x3, x4, x5, x6* respectively *and* $\lambda_{I3,C13}$ – *substitution of w0, w2, w6, w1, w5, w2 instead of z1, z3, z4, z5, z6, z7* respectively. Besides,

I1(w2,w4,w0,w6,w5,w1) = V(w2,w0,w6) & V(w0,w1,w4) & V(w0,w4,w2) & L(w2,w4,w5) & C13(w0,...,w6),
I3(w0,z2,w2,w6,w1,w5,w4,z8) =

= V(w0,w2,w3) & V(w0,w1,w3) & V(w2,w6,w0) & V(w6,w3,w2) & V(w6,w2,w7) & V(w6,w3,w7) &

& C13(w0,...,w6).

As *I1(w2,w4,w0,w6,w5,w1)* contains *V(w2,w0,w6), I3(w0,z2,w2,w6,w1,w5,w4,z8)* contains *V(w2,w6,w0)* and according to the definition of the predicate *V* the formula *V(x,y,z) & V(x,z,y)* is a contradiction so substitutions with this unifiers can not give a consistent description of the object. After deleting from

I1(x1,...,x6) and I3(z1,...,z8) the variables *x1* and *z3* respectively a new maximal common up to the names of arguments their sub-formula *C'13(w0,w1,w2)* in the form

C'13(w0,w1,w2) = L(w1,w0,w2)

will be received with the unifiers $\lambda_{I1,C'13}$ – substitution *of* w*0, w1, w2* instead of *x1, x2, x5* respectively and $\lambda_{I3,C'13}$ – substitution of *w2, w1, w0* instead of *z3, z4, z5* respectively.

Besides,

I1(w0,w1,x3,x4,w2,x6) =

= V(w0,w1,x4) & V(w0,w2,x4) & V(w0,x3,w1) & V(w0,x3,w2) & V(w0,x3,x4) & V(w1,w0,x3) &

& V(w1,x3,w2) & V(x3,w1,w0) & V(x3,x6,w1) & V(x3,x6,w0) & C'13(w0,w1,w2),

I3(z1,z1,w2,w1,w0,z6,z7,z8) =

= V(z1,w0,w2) & V(z1,w2,z2) & V(z1,w0,z2) & V(w2,z1,z7) & V(w2,z1,z6) & V(w2,z7,w1) &

& V(w2,z6,w1) & V(w2,w1,z1) & V(w1,z2,w2) & V(w1,w2,z8) & V(w1,z2,z8) & C'13(w0,w1,w2).

According to the item 4 of the algorithm we identify new variables substituted instead of the same initial variable. That is we identify the following variables

u0 and *w0* (are substituted instead of the variable *x1*),
u1 and *w1* (are substituted instead of the variable *x2*),
u2 and *w2* (are substituted instead of the variable *x4*),
u0 and *v0* (are substituted instead of the variable *y1*),
u1 and *v1* (are substituted instead of the variable *y2*),
u2 and *v2* (are substituted instead of the variable *y3*),
v0 and *w0* (are substituted instead of the variable *z6*),
v1 and *w1* (are substituted instead of the variable *z3*),
v2 and *w2* (are substituted instead of the variable *z7*).

The identified variables denote as $\alpha 0, \alpha 1, \alpha 2$. So we have the equalities *u0 = v0 = w0 =* $\alpha 0$, *u1 = v1 = w1 =* $\alpha 1$, *u2 = v2 = w2 =* $\alpha 2$.

As a result we have the following descriptions of the fragments

I1(I1($\alpha 0,\alpha 1$*,u4,u2,*$\alpha 2$*,x6) =*

= V($\alpha 0,\alpha 1$*,u2) & V(*$\alpha 0,\alpha 2$*,u2) & V(*$\alpha 0$*,u4,*$\alpha 1$*) & V(*$\alpha 0$*,u4,*$\alpha 2$*) & V(*$\alpha 0$*,u4,u2) & V(*$\alpha 1,\alpha 0$*,u4) &*

& V($\alpha 1$*,u4,*$\alpha 2$*) & L(*$\alpha 1,\alpha 0,\alpha 2$*) & V(x3,*$\alpha 1,\alpha 0$*) & V(u4,x6,*$\alpha 1$*) & V(u4,x6,*$\alpha 0$*),*

$I2(\alpha0,\alpha1,u2,y4,\alpha2,u4) =$
$= V(u2,\alpha0,y4)\ \&\ V(\alpha0,\alpha1,u2)\ \&\ V(\alpha0,\alpha2,u2)\ \&\ V(\alpha0,u4,\alpha1)\ \&\ V(\alpha0,u4,\alpha2)\ \&\ V(\alpha0,u4,u2)\ \&$
$\&\ L(\alpha1,\alpha0,\alpha2),$

$I3(z1,z2,\alpha2,z4,z5,\alpha0,\alpha1,z8) =$

$= V(z1,z5,\alpha2)\ \&\ V(z1,\alpha2,z2)\ \&\ V(z1,z5,z2)\ \&\ V(\alpha2,z1,\alpha1)\ \&\ V(\alpha2,z1,\alpha0)\ \&\ V(\alpha2,\alpha1,z4)\ \&$

$\&\ V(\alpha2,\alpha0,z4)\ \&\ V(\alpha2,z4,z1)\ \&\ V(z4,z2,\alpha2)\ \&\ V(z4,\alpha2,z8)\ \&\ V(z4,z2,z8)\ \&\ L(\alpha1,\alpha0,\alpha2).$

Their conjunction

$V(\alpha0,\alpha1,u2)\ \&\ V(\alpha0,\alpha2,u2)\ \&\ V(\alpha0,u4,\alpha1)\ \&\ V(\alpha0,u4,\alpha2)\ \&\ V(\alpha0,u4,u2)\ \&\ V(\alpha1,\alpha0,u4)\ \&\ V(\alpha1,u4,\alpha2)\ \&\ V(x3,\alpha1,\alpha0)\ \&\ V(u4,x6,\alpha1)\ \&\ V(u4,x6,\alpha0)\ \&\ V(u2,\alpha0,y4)\ \&\ V(z1,z5,\alpha2)\ \&\ V(z1,\alpha2,z2)\ \&\ V(z1,z5,z2)\ \&\ V(\alpha2,z1,\alpha1)\ \&\ V(\alpha2,z1,\alpha0)\ \&\ V(\alpha2,\alpha1,z4)\ \&\ V(\alpha2,\alpha0,z4)\ \&\ V(\alpha2,z4,z1)\ \&\ V(z4,z2,\alpha2)\ \&\ V(z4,\alpha2,z8)\ \&\ V(z4,z2,z8)\ \&\ L(\alpha1,\alpha0,\alpha2)$
allows to "stick together" the images of fragments according to the same variable. The image corresponding to the result of "sticking" is presented on the Figure 3.

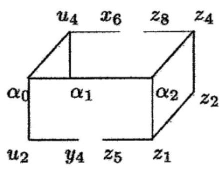


Figure 3. Image corresponding to the result of "sticking"

If a description of the investigated object is presented in the database it may be found according the principle "the nearest neighbor" with the use of metric for predicate formulas presented in [Kosovskaya, 2012].

Conclusion

An algorithm solving a rather complicated problem of multi-agent description of a complex object in the terms of the predicate calculus language with the condition that different agents may give different names to the same elements of the object is presented in the paper. The length of such an object description in the terms of the propositional calculus language is exponential in comparison with that in the terms of the predicate calculus language [Rassel]. It explains the exponential upper bound of the algorithm number of steps. Moreover, it may be easily proved that the problem under consideration is NP-hard.

The analysis of the received estimations allows to formulate restrictions upon the initial predicates for decreasing of the practical time of the algorithm run. For example, if we deal with a great number of initial predicates every of which has very small number of occurrences in the object description then the practical time of the algorithm run decreases.

Acknowledgements

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

Bibliography

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

[Kosovskaya, 2009] T. M. Kosovskaya. "Partial deduction of a predicate formula as an instrument for recognition of an object with incomplete description", In: Vestnik of St. Petersburg University. Ser. 10, 2009, No. 1, pp. 77 – 87 (In Russian)

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

[Kosovskaya, 2012] Kosovskaya T., "Distance between objects described by predicate formulas" // In: International Book Series. Informational Science and Computing. Book 25. Mathematics of Distances and Applications (Michel Deza, Michel Petitjean, Krasimir Markov (eds)), ITHEA-Publisher, Sofia, Bulgaria, 2012, pp. 153 – 159

[Kosovskaya, 2014] 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, pp. 92 – 99

[Nilsson] Nilsson N.J., "Problem-solving methods in artificial intelligence", mcgraw-hill book company, New York, 1971.

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

Authors' Information

Tatiana Kosovskaya *– Dr., Professor of St. Petersburg State University, University av., 28, Stary Petergof, St .Petersburg, 198504, Russia; e-mail: kosovtm@gmail.com*

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