# Conjunctive Boolean Query as a logic-objective recognition problem

**Tatiana Kosovskaya**

***Abstract:*** *A well-known NP-complete problem Conjunctive Boolean Query is considered as the one of the logic-objective recognition problems. Both these problems have the same formulation but their implementations are rather different. It is offered to adapt the technique which involves a decreasing of computational complexity for a logic-objective recognition problem by means of construction of a level class description to the solution of Conjunctive Boolean Query.*

***Keywords:*** *NP-completeness, Conjunctive Boolean Query, logic-objective recognition, multi-level description*

***ACM Classification Keywords***: *F.1.3 Complexity Measures and Classes, Reducibility and completeness; F.2.1 Analysis of Algorithms and Problem Complexity, Numerical Algorithms and Problems, Number-theoretic computations*

## Introduction

Many recent scientific investigations are devoted to the analysis of algorithms solving different NP-complete problems. Essential attention is given to repeatedly solved ones with very big input data. Two problems "Conjunctive Boolean Query" and "Satisfiability in a Finite Interpretation" (used for the solving of a logic-objective recognition problem) having the same formulations but essentially different implementations are under consideration in the paper. In particular, the input data of these problems may be divided into two parts. While solving the first problem one part of input data remains practically fixed and the other changes while every query. While solving the other problem the first part changes while every query and the other remains practically fixed.

This difference does not allow to use directly the technique which involves a decreasing of computational complexity for a logic-objective recognition problem by means of construction of a level class description Kosovskaya, [2008] to the solution of Conjunctive Boolean Query.

While creating and the use of a data base the time of data processing is one of the most important parameters. It is essentially significant because of huge volume of information stored in contemporary data bases. Conjunctive Boolean Query is one of NP-complete problems concerning data bases. Here is its formulation in the form as it is done in Garey, Johnson, [1979].

**Conjunctive Boolean Query** (Garey, Johnson, [1979])

Instance: Finite domain set $D$, a collection $R = \{R_1, R_2, \dots, R_n\}$ of relations, where each $R_i$ consists of a set of $d_i$-tuples with entries from $D$, and a conjunctive Boolean query $Q$ over $R$ and $D$, where such a query $Q$ is of the form

$$\exists y_1, y_2, \dots, y_l (A_1 \ \& \ A_2 \ \& \ \dots \ \& \ A_r)$$

with each $A_i$ of the form $R_j(u_1, u_2, \dots, u_{d_j})$ where each $u \in \{y_1, y_2, \dots, y_l\} \cup D$.

Question: Is $Q$, when interpreted as a statement about $R$ and $D$, true?

As far as NP-complete problems are the problems of the form $\exists Y \ P(X, Y)$, where $X$ is input data, let's give another formulation of the above mentioned problem.

**Conjunctive Boolean Query**

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

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

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

Such setting of the problem Conjunctive Boolean Query is very similar to the earlier investigated in Kosovskaya, [2007, 2008] problem Satisfiability in a Finite Interpretation appeared while recognition of an object in the frameworks of logic-objective approach to the pattern recognition.

**Satisfiability in a Finite Interpretation**

Instance: *A set* $\omega = \{\omega_1, \ldots, \omega_t\}$,
*a collection of predicates* $\{p_1, \ldots, p_n\}$, *setting properties of elements from* $\omega$ *and relations between them,*
*a collection* $S(\omega)$ *of true constant atomic predicate formulas of the form* $p_i(\overline{\tau})$, *where* $i = 1, \ldots, n$, $\tau \subseteq \omega$,
*quantifier-less formula* $A(\overline{y})$, *presented in the form of disjunction of elementary conjunctions of atomic predicate formulas*[1].
Question: *Is there exist a list of values for* $\overline{y}$ *from* $\omega^a$, *such that the formula* $A(\overline{y})$ *is true?*
That is whether

$$S(\omega) \Rightarrow \exists \overline{y} A(\overline{y})?$$

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

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

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

## Problems of logic-objective recognition

Investigated objects in many Artificial Intelligence problems may be described in the terms of properties of their parts and relations between them. In such a case an investigated object $\omega$ may be represented as a set of its parts $\omega = \{\omega_1, ..., \omega_t\}$ and the properties and relations between these parts are described by predicates $p_1, ..., p_n$ defined on them. Facts which are known for an investigator over an object $\omega$ are defined by the set of constant atomic formulas $S(\omega)$ which is called the description of $\omega$.

Below the notation $\overline{x}$ will be used to designate an ordered list of variables $\overline{x} = (x_1, ..., x_m)$. In particular, $\overline{\omega}$ designate some ordered list of elements of $\omega$ corresponding to some permutation of its elements.

The goal condition of the problem may be represented by such a formula $A(\overline{x})$ of a formalized language that if the formula $A(\overline{\omega})$ is valid for an investigated object $\omega$ then the problem has a positive solution. Moreover the goal condition may be represented by a quantifier-free formula in the form of disjunction of elementary conjunctions of atomic predicate formulas.

The following problem may be considered as formalization for an essential example of Artificial Intelligence problems.

**Identification problem.** *To extract a part of the object* $\omega$ *satisfying the goal condition* $A(\overline{x})$.

This problem may be reduced to checking the formula Kosovskaya, [2007]

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

(where $\exists \overline{x}_{\neq}$ means "$\exists \overline{x}$ and its elements are distinct in pairs") and is an NP-complete one.

[1] $\overline{y} = (y_1, \ldots, y_a)$ be a list of objective variables of the formula

If one can solve the problem $S(\omega) \Rightarrow \exists\overline{x}_{\neq}A(\overline{x})$, where $A(\overline{x})$ is a conjunction of atomic formulas then he can solve the problem with $A(\overline{x})$ be a disjunction of elementary conjunctions, and the number of steps of its solution would differ from the first one polynomially. That is why the complexity bounds of algorithms will be done for this problem with $A(\overline{x})$ be a conjunction of atomic formulas.

The **exhaustive search method** is one which allows not only to prove the sequent but also to finds values for variables $\overline{x}$. It is proved in Kosovskaya, [2007] that its number of steps is

$$O(t^m),$$

where $t$ is the number of elements in $\omega$, $m$ is the number of variables in the formula $A(\overline{x})$.

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

$$O(\sum_{i:p_i\ is\ in\ A(\overline{x})} s_i^{a_i}),$$

where $s_i$ and $a_i$ are the numbers of occurrences of the predicate $p_i$ in the description $S(\omega)$ and in the formula $A(\overline{x})$ respectively.

One can see that these upper bounds of number of steps of the algorithms have different parameters in the exponent of the power. So a researcher may choose the method in applications in dependence of the structure of the initial predicates and the goal conditions.

It is evident that the algorithms solving the problem Conjunctive Boolean Query have the same estimates.

## Level description of classes

To decrease the obtained step number estimates a level description of goal formulas was offered in Kosovskaya, [2008]. The procedure of a level description construction uses the following definition.

**Definition.** *Two elementary conjunctions of atomic predicate formulas $A$ and $B$ are called isomorphic if there exists such an elementary conjunction $C$ and such substitutions $\lambda_{A,C}$ and $\lambda_{B,C}$ of objective variables from $C$ instead of objective variables from $A$ and $B$ respectively that the results of these substitutions $A\lambda_{A,C}$ and $B\lambda_{B,C}$ coincide with the formula $C$ up to the order of literals.*

*These substitutions $\lambda_{A,C}$ and $\lambda_{B,C}$ are called the unifiers of formulas $A$ and $B$ respectively with the formula $C$.*

Let the set of all possible objects $\omega$ is divided to some classes and these classes have descriptions consisting of conjunctions of atomic predicate formulas $A_1(\overline{x}_1)$, ..., $A_K(\overline{x}_K)$.

Find all sub-formulas $P_i^1(\overline{y}_i^1)$ with a "small complexity" such that sub-formulas isomorphic to them "frequently" appear in goal formulas $A_1(\overline{x}_1)$, ..., $A_K(\overline{x}_K)$ and denote them by atomic formulas with new first-level 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

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

Let $A_k^1(\overline{x}_k^1)$ be a formula received from $A_k(\overline{x}_k)$ by substitution of $p_i^1(z_{i,j}^1)$ instead of the $j$th appearance of the formula isomorphic to $P_i^1(\overline{y}_i^1)$. 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 $\omega_i^1$ denotes some ordered list $\overline{\tau}_i^1$ of elements from $\omega$ 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.

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

**Procedure of a level description use** for the identification problem consists in the following Kosovskaya, [2014].

1. For every $i$ check $S(\omega) \Rightarrow \exists\overline{y}_{i\neq}^1 P_1^1(\overline{y}_i^1)$ and find all unifiers of $P_1^1(\overline{y}_i^1)$ with true first-level predicates. Add these first-level true atomic formulas to the object description and form $S^1(\omega)$. $l := 1$.

2. If an $l$-level ($l = 1, \ldots, L-1$) object description $S^l(\omega)$ is formed then for every $i$ check $S^l(\omega) \Rightarrow \exists\overline{y}_{i\neq}^l P_1^l(\overline{y}_i^l)$ and find all values for true $(l+1)$-level predicate arguments.

3. Add these $(l+1)$-level true atomic formulas to the object description $S^l(\omega)$ and receive $S^{l+1}(\omega)$.

4. Substitute $p_i^l(y_{i,j}^l)$ instead of the $j$th appearance of $P_i^l(\overline{y}_i^l)$ into $A_k^l(\overline{y}_k^l)$.

5. Repeat the previous steps for $l = 1, \ldots, L$.

6. Check $S^L(\omega) \Rightarrow \exists\overline{y}_{k\neq}^L A_k^L(\overline{y}_k^L)$.

Such $L$-level goal conditions may be used for efficiency of an algorithm solving a problem formalized in the form of logical sequent. 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

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

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 is

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

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, $\rho_j^1$ is the number of atomic formulas in $P_i^1(\overline{y}_i^1)$.

Extraction of such sub-formulas $P_i^1(\overline{y}_i^1)$ with a "small complexity" which "frequently" appear in goal formulas $A_1(\overline{x}_1)$, ..., $A_K(\overline{x}_K)$ is described in Kosovskaya, [2014]. The procedure of the extraction of the maximal sub-formula which is isomorphic to some sub-formulas of two elementary conjunctions is described in Petrov, [2016].

**Level description of classes construction.**

1) For every $i = 1, \ldots, n-1$, $j = i+1, \ldots, n$ extract maximal sub-formulas $Q_{i,j}$ which are isomorphic to some sub-formulas of elementary conjunctions $A_i(\overline{x}_i)$ and $A_j(\overline{x}_j)$ ($i \neq j$) and find common unifiers $\lambda_{i,ij}$ and $\lambda_{j,ij}$ of these conjunctions with $Q_{i,j}$.

$l$) The procedure of maximal sub-formulas extraction and obtaining of common unifiers is repeated with the formulas received earlier.

The process will end for some $l = L$, because the lengths of the extracted formulas decrease from step to step. Sub-formulas consisting of one literal are not under consideration. That's why the item $l$ is repeated for $l = 2, \ldots, L$.

$L+1$) The extracted sub-formulas that haven't another common sub-formulas are denoted by $P_i^1(\overline{y}_i^1)$ $(i = 1, \ldots, n_1)$.

$L+2$) For every $l = 1, \ldots, L-1$ sub-formulas having common sub-formulas $P_i^l(\overline{y}_i^l)$ are denoted by $P_i^{l+1}(\overline{y}_i^{l+1})$ $(i = 1, \ldots, n_{l+1})$. At the same time instead of the $j$th occurrence of $P_i^l(\overline{y}_i^l)$ substitute $p_i^l(y_{i,j}^l)$ where $y_{i,j}^l$ is an $l$-level variable for a list of variables $\overline{y}_{i,j}^l$ taking into account the corresponding unifier.

## An approach to the construction of a level data base for solving the Conjunctive Boolean Query problem

While creation a data base we are not sure what queries may make a user. But the data base itself remains practically fixed. That's why patterns (formulas isomorphic to some conjunctions of atomic formulas from the data base) must be searched in the data base itself (in the set of constant atomic formulas $S(D)$).

In order to receive complexity estimates regard the case when the query has the form of one elementary conjunction, as well as it was done in the problem of logic-objective recognition.

Remind, that the estimates of the number of steps needed for checking the logical consequence of the form $C(\overline{x}) \Rightarrow \exists y_1, y_2, \ldots, y_l A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r)$, where $C(\overline{x})$ is a set of atomic formulas or their conjunction, are the following:

— $O(t^l)$, where $t$ is the number of elements in $\overline{x}$, $l$ is the number of variables in $A(\overline{y})$, while using an exhaustive algorithm;

— $O(\sum_{i:p_i\ is\ in\ A(\overline{y})} s_i^{a_i})$, where $s_i$ and $a_i$ are the numbers of occurrences of the predicate $p_i$ in the $C(\overline{x})$ and in the formula $A(\overline{y})$ respectively, while using a logic algorithm.

**Two-level data base construction**

1. Let for $i = 1, \ldots, n_1$ groups of mutually disjoint sub-sets of $S(D)$ such that all conjunctions of elements of a sub-set from the $i$th group are isomorphic to each other and to some formula $P_i^1(\overline{y}_i^1)$ are extracted. For every group unifiers of the conjunctions of a sub-set elements with $P_i^1(\overline{y}_i^1)$ are found.

2. Introduce first-level predicates $p_i^1$ $(i = 1, \ldots, n_1)$ defined by the equivalence $p_i^1(y_i^1) \Leftrightarrow P_i^1(\overline{y}_i^1)$, where $y_i^1$ are the first-level variables for the lists of initial variables $\overline{y}_i^1$.

3. Supplement the set $S(D)$ by the set of atomic first-level formulas in the form $p_i^1(d_i^{1,j})$, where $d_i^{1,j}$ is the notation of a list of constants from $D$ which is included into unifier of some conjunction of a sub-set with $P_i^1(\overline{y}_i^1)$.

A two-level data base $S^1(D)$ is constructed.

Note that the construction of a two-level data base is an NP-hard problem with huge input data, because in the item 1 it is solved an NP-hard problem of extraction of groups of mutually disjoint sub-sets with small capacity of $S(D)$ such that all conjunctions of elements of a sub-set from the a group are isomorphic to each other from all formulas of the data base.

But this NP-hard problem with huge input data is solved only once.

**Two-level data base implementation**

Let we have a conjunctive Boolean query $\exists y_1, y_2, \ldots, y_l A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r)$, where $d_1, \ldots, d_r$ are constants from $D$.

1. Checking the logical sequent

$$A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r) \Rightarrow \exists \overline{y}_i^1 P_i^1(\overline{y}_i^1)$$

for $i = 1, \ldots, n_1$ allows (if it is fulfilled) to find all sub-formulas of the query, which are isomorphic to $P_i^1(\overline{y}_i^1)$, and their unifiers with the corresponding lists of constants.

In spite of the fact that the problem in this item is NP-hard, its input data have not large length and the estimates of number of steps have not large parameters of the formula $P_i^1(\overline{y}_i^1)$ in the exponent (the number of variables or the number of atomic formulas with the same predicate). These parameters are less then the corresponding parameters of the formula $A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r)$.

These estimates have the form

— $O(\sum_{i=1}^{n_1}(l+r)^{\|\overline{y}_i^1\|})$, $\|\overline{y}_i^1\|$ be the number of arguments in $P_i^1(\overline{y}_i^1)$, for an exhaustive algorithm;

— $O(\sum_{i=1}^{n_1}\sum_{j:p_j\ is\ in\ P_i^1(\overline{y}_i^1)} a_j^{\alpha_j}))$, $a_j$ and $\alpha_j$ be the number of atomic formulas with the predicate $p_j$ in $A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r)$ and $P_i^1(\overline{y}_i^1)$ respectively, for a logical algorithm.

2. Substitute into $A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r)$ instead of every sub-formula,which is isomorphic to $P_i^1(\overline{y}_i^1)$, atomic formula $p_i^1(y_i^{1,j})$, where $y_i^{1,j}$ is a variable for a list of variables and constants of $\overline{y}_i^1$ (index $j$ changes from 1 to the number of occurrences sub-formulas isomorphic to $P_i^1(\overline{y}_i^1)$). It is possible because we have unifiers of every such sub-formulas with $P_i^1(\overline{y}_i^1)$). An elementary conjunction $A^1(\overline{x}^1)$ is received. Here $\overline{x}^1$ is a list of the initial variables, 1-level variables and constants that remains explicitly in the formula as arguments.

This item is fulfilled in linear under the notation length of $A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r)$ number of steps. Note, that the notation length of $A^1(\overline{x}^1)$ is not greater than the notation length of $A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r)$ and is strictly less if at least one sub-formula has been changed by an atomic one of the first level.

3. While checking $S^1(D) \Rightarrow \exists\overline{x}^1 A^1(\overline{x}^1)$ first of all check atomic formulas of $A^1(\overline{x}^1)$ with first-level predicates and find possible values for first-level variables. After that check atomic formulas of $A^1(\overline{x}^1)$ with initial predicates taking into account the values of initial variables that have occurrences into lists defining first-level variables.

The number steps estimates for checking atomic formulas with first-level predicates and finding values for first-level variables are

— $O(t_1^{l_1})$ ($t_1$ be the number of first-level constants in $S^1(D)$, $l_1$ be the number of first-level variables in $A^1(\overline{x}^1)$) for an exhaustive algorithm;

— $O(\sum_{i:p_i^1\ is\ in\ A^1(\overline{x}^1)} {s_i^1}^{a_i^1})$ ($s_i^1$ and $a_i^1$ be the numbers of occurrences of the predicate $p_i^1$ in the description $S^1(\omega)$ and in the formula $A^1(\overline{x}^1)$ respectively) for a logical algorithm.

Note that if at least one sub-formula has been changed by an atomic first-level one then the number of initial variables in $\overline{x}^1$ and the number of atomic formulas with initial predicates in $A^1(\overline{x}^1)$ decreases in comparison with the Boolean query $A(y_1, y_2, \ldots, y_l, d_1, \ldots, d_r)$.

The number steps estimates for checking checking atomic formulas with initial predicates and finding values for initial variables are

— $O(t^{l-l_1})$ ($t_1$ be the number of first-level constants in $S^1(D)$, $l_1$ be the number of first-level variables in $A^1(\overline{x}^1)$) for an exhaustive algorithm;

— $O(\sum_{i:p_i\ is\ in\ A^1(\overline{x}^1)} {s_i}^{a_i-a_i^1})$ ($s_i^1$ and $a_i^1$ be the numbers of occurrences of the predicate $p_i^1$ in the description $S^1(\omega)$ and in the formula $A^1(\overline{x}^1)$ respectively) for a logical algorithm.

If we sum the estimates for items 1 – 3 we obtain complete estimates

— $O(\sum_{i=1}^{n_1}(l+r)^{\|\overline{y}_i^1\|}) + O(t_1^{l_1}) + O(t^{l-l_1}) = O(t_1^{l_1} + t^{l-l_1})$ for an exhaustive algorithm;

— $O(\sum_{i=1}^{n_1}\sum_{j:p_j\ is\ in\ P_i^1(\overline{y}_i^1)} a_j^{\alpha_j}) + O(\sum_{i:p_i^1\ is\ in\ A^1(\overline{x}^1)} {s_i^1}^{a_i^1}) + O(\sum_{i:p_i\ is\ in\ A^1(\overline{x}^1)} {s_i}^{a_i-a_i^1}) = O(\sum_{i:p_i^1\ is\ in\ A^1(\overline{x}^1)} {s_i^1}^{a_i^1}) + O(\sum_{i:p_i\ is\ in\ A^1(\overline{x}^1)} {s_i}^{a_i-a_i^1})$ for a logical algorithm.

It is obvious that if at least one sub-formula has been changed by an atomic first-level one then the number steps estimates for checking the Boolean query sequent from the two-level data base is less than its sequent from the initial data base estimates.

## Conclusion

Essential differences of the well-known NP-complete problem Conjunctive Boolean Query while its implementation as a problem of pattern recognition in the frameworks of logic-objective approach and as a problem of the data base use are shown.

In the both implementations it is possible to construct a level description of a fixed input data which increases the time complexity of multiple implementation. The estimates of number of steps while using a two-level data base are proved in the paper.

Algorithms of level description of classes are yet developed in the previous papers of the author, but there is only an approach to developing of algorithms of level data base construction.

## Acknowledgements

The paper is published with partial support by the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua)

## Bibliography

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

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

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

Kosovskaya T.M. Level system of formulas for decreasing the number of proof steps of formulas simulating some Artificial Intelligence problems // CLMPS. 15-th Congress of Logic, Methodology and Philosopfy of Science. Book of abstracts. 3 - 8 August 2015, University of Helsinki. P. 283.

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

## 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 Computational Complexity of Algorithms*