CSIT Conference 2021, Yerevan, Armenia, September 27 - October 1

# Aggregation of Information from Various Agents in Terms of Predicate Formulas

Tatiana Kosovskaya
St. Petersburg State University
St. Petersburg, Russia
email: kosovtm@gmail.com

***Abstract*—Two problems of an object description based on certain incomplete information received by several agents are under consideration. An object in the problems is presented as a set of its elements and is characterized by properties of these elements and relations between them. The description of an object is an elementary conjunction of constant atomic predicate formulas, which are true for this object. It is supposed that all agents in the first problem have the same name for every object element. In the second problem, every agent does not know the true names of the object elements and arbitrarily gives them names. Algorithms solving the set problems are described and the upper bounds of these algorithms run steps are proved. Model examples of an algorithm implementation are given for the problems.**

***Keywords*— Predicate formulas, multi-agent description, aggregation of information, computational complexity.**

## I. Introduction

The use of the predicate calculus language for solving artificial intelligence problems was proposed in the middle of the 20th century (e.g. [1]) and continues to be described in theoretical works (e.g. [2]). It is the language of predicate calculus that allows one to adequately describe complex objects characterized by the properties of their parts and the relations between them.

When describing complex structured objects, it is convenient to use the predicate calculus language, as it actually happens when creating Databases. That is, each object is represented by a set of its elements. A set of predicates, that specify properties of elements and relations between them, is defined on these objects. A description of an object is an elementary conjunction of all literals with these predicates, which are true for the object.

Unfortunately, with such a choice of the object description language, the arising problems are, as a rule, NP-hard problems with computational complexity that exponentially depends on the length of the initial data record. This is not a fundamental reason for rejecting this approach, since when modeling the description of an object in the terms of predicate calculus using a binary string, the length of such a string depends exponentially on the length of this description.

In this paper, two models for the aggregation of information about an object described in terms of predicate calculus language, obtained from various agents are proposed. In the proposed models, each agent has a part of the object description, but the difference in the setting is whether an agent knows the real (or at least common for all agents) names of the parts of the described object or each agent has the right to name each part of the object in its own way.

The solution of the first problem is quite simple. It is presented in the paper mainly to emphasize the complexity of the second problem.

The second problem resembles the parable of the three blind men feeling an elephant. In the formulation of the second problem, it is assumed that each pair of agents has information about some common part of the object, calling these parts differently. The main problem is finding and identifying these common parts.

## II. Required definitions

When solving the set problems, important concepts are such as “isomorphism of predicate formulas”, “maximal common (up to the argument names) subformula”, “consistency of descriptions”.

**Definition 1.** [3] *Elementary conjunctions of predicate formulas $P(a_1,\dots,a_m)$ and $Q(b_1,\dots,b_m)$ are called* **isomorphic**

$$P(a_1,\dots,a_m) \sim Q(b_1,\dots,b_m),$$

*if there are an elementary conjunction $R(x_1,\dots,x_m)$ and substitutions of arguments $a_{i_1},\dots,a_{i_m}$ and $b_{j_1},\dots,b_{j_m}$ instead of the variables $x_1,\dots,x_m$ such that the results of these substitutions $R(a_{i_1},\dots,a_{i_m})$ and $R(b_{j_1},\dots,b_{j_m})$ coincide with formulas $P(a_1,\dots,a_m)$ and $Q(b_1,\dots,b_m)$, respectively, up to the order of literals.*

*The substitutions $(a_{i_1}\to x_1,\dots,a_{i_m}\to x_m)$ and $(b_{j_1}\to x_1,\dots,b_{j_m}\to x_m)$ are called* **unifiers** *of $R(x_1,\dots,x_m)$ with $P(a_1,\dots,a_m)$ and $Q(b_1,\dots,b_m)$ and are denoted as $\lambda_{R,P}$ and $\lambda_{R,Q}$, respectively.*

In fact, the isomorphism relation of elementary conjunctions of literals means their coincidence up to the names of variables and literal order. So, for example, elementary conjunctions $P(x,y,z) = (x<y)\ \&\ (y<z)$ and $Q(x,2,3) = (2<x)\ \&\ (x<3)$ do not have common subformulas, but they both define the ternary relation “one of the numbers lies strictly between the other two”. More precisely, $P(x,y,z)$ defines the relation $y\in(x,z)$, and $Q(x,2,3)$ defines the relation $x\in(2,3)$. Here $R(x,y,z)$ may be defined as $x\in(y,z)$ and $R(x,y,z) = (y<x)\ \&\ (x<z)$ with unifiers $\lambda_{R,P} = (y\to x, x\to y, z\to z)$ and $\lambda_{R,Q} = (x\to x, 2\to y, 3\to z)$, respectively.

It is easy to see that the isomorphism relation of predicate formulas is an equivalence relation. The problem of checking two elementary conjunctions of predicate formulas for isomorphism is polynomially equivalent to the "open" problem *Graph isomorphism* [4], for the solution of which no polynomial algorithm is known and its NP-completeness has not been proved.

**Definition 2.** [5] *Elementary conjunction $C(x_1, \ldots, x_n)$ is called a common up to the names of arguments sub-formula of two elementary conjunctions $A(a_1, \ldots, a_m)$ and $B(b_1, \ldots, b_k)$ if it is isomorphic to some sub-formulas $A'(a'_1, \ldots, a'_{m'})$ and $B'(b'_1, \ldots, b'_{k'})$ of $A(a_1, \ldots, a_m)$ and $B(b_1, \ldots, b_k)$, respectively.*

*The unifiers of $C(x_1, \ldots, x_n)$ with $A'(a'_1, \ldots, a'_{m'})$ and $B'(b'_1, \ldots, b'_{k'})$ will be denoted as $\lambda'_{C,A}$ and $\lambda'_{C,B}$, respectively.*

For example, let

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

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

The formula

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

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

$$P(x, y) = p_1(x)\&p_1(y)\&p_2(x, y)$$

is a sub-formula of $A(x, y, z)$ and

$$P(x, z) = p_1(x)\&p_1(z)\&p_2(x, z)$$

is a sub-formula of $B(x, y, z)$.

**Definition 3.** [5] *Elementary conjunction $M(x_1, \ldots, x_n)$ is called a maximal common (up to the names of arguments) sub-formula of two elementary conjunctions $A(a_1, \ldots, a_m)$ and $B(b_1, \ldots, b_k)$ if it is their common up to the names of arguments sub-formula and after adding any literal to it, it ceases to be one.*

**Definition 4.** [6] *Conjunction of literals from $A(\overline{x})$ which are not in its sub-formula $A'(\overline{x}')$ is called a* **complement** *of $A'(\overline{x}')$ up to $A(\overline{x})$.*

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

**Definition 5.** [6] *Let $M(x_1, \ldots, x_n)$ be a maximal common (up to the names of arguments) sub-formula of two elementary conjunctions $A(a_1, \ldots, a_m)$ and $B(b_1, \ldots, b_k)$ with unifiers $\lambda'_{M,A}$ and $\lambda'_{M,B}$, respectively.*

*Elementary conjunctions $A(a_1, \ldots, a_m)$ and $B(b_1, \ldots, b_k)$ are called* **contradictory** *on $M(x_1, \ldots, x_n)$ if their complements to the formula $M(x_1, \ldots, x_n)$ with substitutions $\lambda'_{M,A}$ and $\lambda'_{M,B}$, respectively, contain contrarian terms (i.e., there is a literal such that after applying the unifiers to the formulas $A(a_1, \ldots, a_m)$ and $B(b_1, \ldots, b_k)$ it enters one of them, and its negation enters the other.)*

Show an example of checking for contradiction. Let

$$A(a, b, c) = p(a, b)\ \&\ p(b, a)\ \&\ q(b, a, c),$$

$$B(a, b, d) = p(b, a)\ \&\ p(b, d)\ \&\ q(a, b, d),$$

$$C(a, b, c) = p(a, b)\ \&\ \neg p(a, c)\ \&\ q(b, a, c)$$

Common up to the names of arguments sub-formula of $A(a, b, c)$ and $B(a, b, d)$ has the form

$$M_{AB}(u, v, w) = p(u, v)\ \&\ q(v, u, w)$$

with unifiers $\lambda'_{M_{AB}A} = (a \to u, b \to v, c \to w)$ and $\lambda'_{M_{AB}B} = (b \to u, a \to v, d \to w)$.

Using the inverse unufiers $\lambda'^{-1}_{M_{AB}A} = (u \to a, v \to b, w \to c)$ and $\lambda'^{-1}_{M_{AB}B} = (u \to b, v \to a, w \to d)$ we have

$$A(u, v, w) = p(u, v)\ \&\ p(v, u)\ \&\ q(v, u, w)$$

$$B(v, u, w) = p(u, v)\ \&\ p(u, w)\ \&\ q(v, u, w).$$

Their complements to the formula $M_{AB}(u, v, w)$

$$C_{A(u,v,w)}M_{AB}(u, v, w) = p(v, u),$$

$$C_{B(v,u,w)}M_{AB}(u, v, w) = p(u, w)$$

do not contain contrarian terms. Formulas of $A(a, b, c)$ and $B(a, b, d)$ are not contradictory on $M_{AB}(u, v, w)$.

Common up to the names of arguments sub-formula of $B(a, b, d)$ and $C(a, b, c)$ has the form

$$M_{BC}(u, v, w) = p(u, v)\ \&\ q(v, u, w)$$

with unifiers $\lambda_{M_{BC}C} = (a \to u, b \to v, c \to w)$ and $\lambda_{M_{BC}B} = (b \to u, a \to v, d \to w)$. In analogous way we have complements of $B(a, b, d)$ and $C(a, b, c)$ to the formula $M_{AB}(u, v, w)$

$$C_{B(v,u,w)}M_{BC}(u, v, w) = p(u, w)$$

$$C_{C(u,v,w)}M_{BC}(u, v, w) = \neg p(u, w).$$

They have contrarian terms. That's why $B(a, b, d)$ and $C(a, b, c)$ are contradictory on $M_{BC}(u, v, w)$.

## III. Setting problems of aggregating information on a complex structured object

Let an object $\omega$ be the set of its elements $\omega = \{\omega_1, \ldots, \omega_t\}$ and be characterized by the predicates $p_1, \ldots, p_n$, specifying the properties of these elements and relations between them.

There are $m$ agents $a_1, \ldots, a_m$, which can measure some predicate values on some elements of an object $\omega$ (that is, determine the properties of some elements of the object under study and some relations between these elements). Each of the agents $a_1, \ldots, a_m$ has information $I_1, \ldots, I_m$, respectively. Information $I_j$ is presented as an elementary conjunction with predicates $p_1, \ldots, p_n$ which is true on $\omega$. The information each agent possesses is absolutely valid.

Full description of an object is a maximal elementary conjunction with predicates $p_1, \ldots, p_n$ which is true on $\omega$. There is a database with full descriptions of all previously known objects.

It is needed to obtain a full description of the object $\omega$ if it is possible.

**Problem 1.** Each agent knows real names of the object elements.

Construct a full description of the object $\omega$ in the form of a conjunction of literals that specify the properties of its elements and relations between these elements.

If there is no object in the database the description of which matches the received one, then find an object from the existing database, the description of which contains the received description (or the obtained description is contained in the description of the object from the database) and analyze which of the descriptions is not complete.

**Problem 2.** Each agent does not know real names of the object elements and gives these elements his own names.

Construct a description of the object $\omega$ in the form of a conjunction of literals that specify the properties of the elements of a given object and the relations between these elements, provided that agent $a_j$ may not know the real number of elements in the object $\omega$ and assume that it is dealing with the object $\omega^j = \{\omega^j_1, ...\omega^j_{t_j}\}$ (element names may differ for each agent).

## IV. Solution of problem 1

Since the information possessed by each agent is assumed to be absolutely valid and the names of the elements of the complex object $\omega = \{\omega_1, \ldots, \omega_t\}$ are known to the agents, then the elementary conjunction $I_1 \ \& \ \ldots \ \& \ I_m$ has no contrarian members, but some literals may be repeated. After removing duplicate literals, we get an elementary conjunction $I$ containing all the information collected by the agents.

If the database contains an object with such a description (more precisely, an object with a description isomorphic to $I$), then we can assume that the agents have collected all available information about the object $\omega$.

If the database contains an object $\tau$ with a description $S(\tau)$, which is isomorphic to a subformula of information $I$, then you should check if the information about $\tau$ is complete.

If the collected information $I$ is isomorphic to a subformula of the description $S(\tau)$ of some object $\tau$ from the base, then you should check whether the collected information is complete.

Otherwise, you should pairwise select maximal common (up to the names of arguments) sub-formula of $I$ and descriptions $S(\tau)$ for all objects $\tau$ from the base. Then check their complements to $I$ and $S(\tau)$ for consistency. If these additions are inconsistent, then $\omega$ is significantly different from $\tau$. If these additions are consistent, then you should check whether the collected information $I$ is complete and whether the information about $\tau$ is complete.

## V. Solution of problem 2

### A. Preliminary considerations

Due to the fact that each agent uses its own designations for the names of object elements, there is no possibility of direct aggregation of all information received from agents. It is required to find all possible maximal common (up to the names of arguments) subformulas of various information and unifiers for them, that is, such changes of object element names that the selected subformulas become graphically equal. After that, "glue" the various parts of the object in accordance with such common subformulas, provided that the information obtained about these parts is not contradictory.

An example of not contradictory to their maximal common (up to the names of arguments) subformula $M_{AB}(u, v, w) = p(u, v) \ \& \ q(v, u, w)$ formulas $A(a, b, c) = p(a, b) \ \& \ p(b, a) \ \& \ q(b, a, c)$ and $B(a, b, d) = p(b, a) \ \& \ p(b, d) \ \& \ q(a, b, d)$ was shown in section 2. We obtained that $A(u, v, w) = p(u, v) \ \& \ p(v, u) \ \& \ q(v, u, w) = M_{AB}(u, v, w) \ \& p(v, u)$ and $B(v, u, w) = p(u, v) \ \& \ p(u, w) \ \& \ q(v, u, w) = M_{AB}(u, v, w) \ \& \ q(v, u, w)$.

The formula $Aggr(A, B) = p(u, v) \ \& \ p(v, u) \ \& \ q(v, u, w) \ \& \ p(u, w)$ is the result of aggregation ("gluing") of these two formulas.

The notion of partial consequence of a formula from a given set of formulas is proposed in [7]. This notion, in particular, allows to extract their maximal common up to the names of variables subformula and construct their unifiers. The notation $A(\overline{x}) \Rightarrow_P B(\overline{y})$ is used to indicate that not a logical consequence $A(\overline{x}) \Rightarrow \exists \overline{y}_{\neq} B(\overline{y})$,[1] is being checked but only that there is such a subformula $B'(\overline{y}')$ of the formula $B(\overline{y})$ and such a sublist $\overline{y}'$ of the list of all variables of the formula $B(\overline{y})$ that $A(\overline{x}) \Rightarrow \exists \overline{y}'_{\neq} B'(\overline{y}')$.

In the process of checking $A(\overline{x}) \Rightarrow_P B(\overline{y})$, the maximal common (up to the names of variables) subformula of the formulas $A(\overline{x})$ and $B(\overline{y})$ and their common unifier are extracted. A detailed description of the algorithm for extracting common subformulas and constructing a common unifier is available, for example, in [3].

### B. Algorithm for aggregating information about an object

1) Replace all constants in $I_1, \ldots, I_m$ with variables so that in each elementary conjunction different constants are replaced with different variables and variable names in the variable lists for $I_i$ and $I_j$ for $i \neq j$ do not coincide. We get a list of elementary conjunctions $I'_1, \ldots, I'_m$.
2) For each pair of formulas $I'_i$ and $I'_j$ ($i \neq j$, $i = 1, \ldots, m-1$, $j = i+1, \ldots, m$) check $I'_i \Rightarrow_P I'_j$ and extract their maximal common (up to the argument names) subformula $C_{ij}$ and the unifiers $\lambda'_{i,ij}$ and $\lambda'_{j,ij}$. Each argument of $C_{ij}$ has a unique name.
3) For each pair $i$, $j$ ($i > j$), check if there are any contradictory formulas in $I'_i$ and $I'_j$ after applying the unifier. That is, either a contradictory pair of atomic formulas, or a contradictory pair of atomic formulas in accordance with the definition of the original predicates. If such a contradiction is revealed, then we remove from $C_{ij}$ atomic formulas in which there are variables corresponding to those included in the contradictory formulas.

[1] Hereinafter $\overline{x}$ is a list of variables. The notation $\exists \overline{y}_{\neq}$ means "there is a list of pairwise different values for the list of variables $\overline{y}$"

We change the unifiers to exclude these variables from them.

4) For each $i$ identify the variables in $C_{ij}$ $(j \neq i)$, which are substituted into $I'_i$ instead of the same initial variable. Replace the names of the identified variables in the substitution part of the unifiers with the same name.
5) Replace the corresponding names of the arguments in $I'_1, \ldots, I'_m$ with the names assigned to the corresponding variables in the subformulas obtained in items 2 - 4 using the obtained unifiers. We get $I''_1, \ldots, I''_m$.
6) The description $I$ is obtained from $I''_1 \; \& \; \ldots \; \& \; I''_m$ by removing duplicate conjunctive terms.
7) Check the presence of the received description in the database. Moreover, if the description $I$ contains the original variables introduced in item 1, then they may need to be identified either with each other or with the newly obtained variables.

## VI. Estimates of the number of steps for solving the problems

Solution of **Problem 1** is reduced to the removal of duplicate conjunctive terms in the elementary conjunction $I_1 \; \& \; \ldots \; \& \; I_m$. If $||I_j||$ is the number of atomic formulas in $I_j$, then the number of "steps" of this procedure does not exceed $\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} ||I_i|| \cdot ||I_j|| \leq \frac{m(m-1)}{2} (\max_{j=1}^{m} ||I_j||)^2$. Here a step is a comparison of two atomic formulas for graphic coincidence.

The total number of steps for solving Problem 1 is $O(m^2 ||I||^2)$, where $||I||$ is the maximal number of atomic formulas in $I_1, \ldots, I_m$.

If it is additionally needed to find the described object in the database, then the number of steps will increase by the number required for searching in the database and depending on the size of the database and the way the data is organized in it.

To estimate the number of steps for solving **Problem 2** let's estimate the number of steps required to complete each item separately.

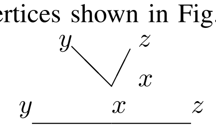
1) Replacing constants with variables requires $\sum_{j=1}^{m} ||I_j||$ "steps".
2) Checking $I_i \Rightarrow_P I'_j$ requires $O({t_i}^{t_j} \cdot 2^{||I_i||})$ "steps" for the exhaustive search algorithm, where $t_i$ is the number of arguments of the elementary conjunction $I_i$, and $O({s_i}^{||I_j||} \cdot ||I_i||^3)$ for an algorithm based on finding an inference in the sequential predicate calculus, or by the resolution method for predicate calculus, where $s_i$ is the maximum number of atomic formulas with the same predicate in $I_i$. (These estimates are obtained in [8].) These estimates should be summed over $i = 1, \ldots, m-1, j = i, \ldots m$.
In total, we get $O(t^t \cdot ||I|| \cdot m^2)$ "steps" for the exhaustive search algorithm, where $t$ and $||I||$ are the maximal numbers of arguments and atomic formulas in $I_i$ $(i = 1, \ldots m)$, respectively. For an algorithm based on inference search in the sequential predicate calculus or the resolution method for predicate calculus, this estimate is $O(s^{||I||} \cdot ||I||^3 \cdot m^2)$, where $s$ is the maximal number of atomic formulas with the same predicate in $I_i$.
3) Checking the consistency of the formula $I'_i$ requires $||I_i||$ "steps". In total, the execution of this point of the algorithm requires no more than $\sum_{i=1}^{m} (m-i)||I_i||$ "steps", which is $O(m^2 ||I||)$ "steps".
4) For each $i$, the identification of variables in $C_{ij}$ $(j > i)$ consists in looking at the substituted part of the unifiers and checking if the substituted part matches. This will take no more than $(m-i)t_i$ "steps".
In total, the execution of this point of the algorithm requires no more than $\sum_{i=1}^{m} (m-i)t_i$ "steps", which is $O(m^2 t)$ "steps".
5) The number of "steps" of changing variable names is linear under $\sum_{j=1}^{m} ||I_j||$, which is $O(m||I||)$ "steps".
6) The number of "steps" for removing repeated conjunctive terms does not exceed $\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} ||I_i|| \cdot ||I_j||$ and is $O(m^2 ||I||^2)$.
7) The number of steps depends on the number of steps required to search the database and depends on the size of the database and the way the data is organized in it. If the description $I$ contains the original variables introduced in section 1, then this number of steps should be multiplied by a value not exceeding $n_1 n_2$, where $n_1$ and $n_2$ are the numbers of original variables and all variables in the resulting description, respectively.

The total number of steps of the algorithm for solving Problem 2 is $O(t^t \cdot ||I|| \cdot m^2)$ steps for the exhaustive search algorithm, where $t$ and $||I||$ are the maximal numbers of arguments and atomic formulas in $I_i$ $(i = 1, \ldots, m)$, respectively, and $O(s^{||I||} \cdot ||I||^3 \cdot m^2)$ for an algorithm based on inference search in sequential predicate calculus or by the resolution method for predicate calculus, where $s$ is the maximal number of atomic formulas with the same predicate in each elementary conjunction $I_i$.

Analysis of the procedure for obtaining this estimate shows that the main contribution to it comes from the number of incomplete derivability checks performed.

## VII. Examples

Consider the solution of Problems 1 and 2 by the example of describing the contour image of a "box" in terms of the predicates $V$ and $L$, which specify the relations between the vertices shown in Fig. 1.




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

$L(x, y, z, ) \Longleftrightarrow x \in (y, z)$

Fig. 1. Initial predicates.

Three agents examine different parts of the image shown in Fig. 2.

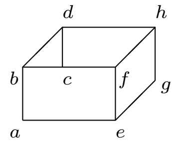



Fig. 2. Image of a “box”.

This image is in the database and has a description

$$V(a,b,e)\&V(e,a,f)\&V(e,f,g)\&V(e,a,g)\&$$
$$V(g,e,h)\&V(b,c,a)\&V(b,f,a)\&V(b,d,c)\&$$
$$V(b,d,f)\&V(b,d,a)\&V(c,b,d)\&V(c,d,f)\&$$
$$V(f,e,c)\&V(f,e,b)\&V(f,c,8)\&V(f,b,h)\&$$
$$V(f,h,e)\&V(d,c,b)\&V(d,h,c)\&V(d,h,b)\&$$
$$V(h,g,f)\&V(h,f,d)\&V(h,g,d)\&L(c,b,f).$$

**Solution of Problem 1.** Each of the agents received, respectively, descriptions of the image fragments shown in Fig. 3.

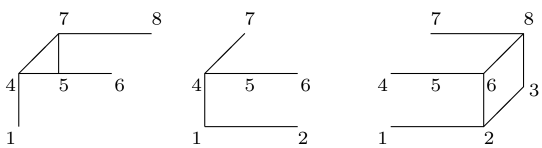



Fig. 3. Fragments of the image obtained by agents when solving problem 1.

Each of these descriptions has a form.[2]

$$I_1 = V(4,5,1)\ \&\ V(4,6,1)\ \&\ V(4,7,5)\ \&\ V(4,7,6)\ \&$$
$$V(4,7,1)\ \&\ V(5,4,7)\ \&\ V(5,7,6)\ \&$$
$$L(5,4,6)\ \&\ V(7,5,4)\ \&V(7,8,5)\ \&\ V(7,8,4),$$
$$I_2 = V(1,4,2)\ \&\ V(4,5,1)\ \&\ V(4,6,1)\ \&\ V(4,7,5)\ \&$$
$$V(4,7,6)\ \&\ V(4,7,1)\ \&\ L(5,4,6),$$
$$I_3 = V(2,1,6)\ \&\ V(2,6,3)\ \&\ V(2,1,3)\ \&\ V(6,2,5)\ \&$$
$$V(6,2,4)\ \&\ V(6,5,8)\ \&\ V(6,4,8)\ \&\ V(6,8,2)\ \&$$
$$V(8,3,6)\ \&\ V(8,6,7)\ \&\ V(8,3,7).$$

The conjunction of these descriptions gives a description isomorphic to that of the object in the database. The unifier is $(1 \to a, 2 \to e, 3 \to g, 4 \to b, 5 \to c, 6 \to f, 7 \to d, 8 \to h)$.

**Solution of Problem 2.**[3] Each of the agents received, respectively, descriptions of the image fragments shown in Fig. 4.

[2]Arguments of $I_j$ will be ommited while it is not important.

[3]In this example, the extraction of the maximal common (up to the argument names) sub-formula of two formulas, the determination of the common unifier and the expression of the formulas by means of the extracted subformulas were carried out using a program written by a student of St. Petersburg State University Petrov D.A.

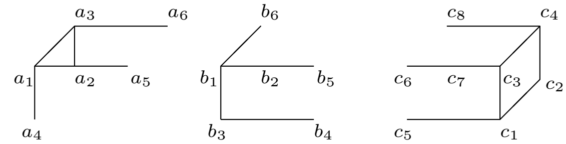



Fig. 4. Fragments of the image obtained by agents for solving problem 2.

In accordance with item 1 of the algorithm, we replace all constants with variables so that different constants are replaced with different variables. Each of these descriptions has a corresponding form.

$$\begin{aligned}
I_1(\overline{x}) = \ & V(x_1,x_2,x_4)\&V(x_1,x_5,x_4)\&V(x_1,x_3,x_2)\& \\
& V(x_1,x_3,x_5)\&V(x_1,x_3,x_4)\&V(x_2,x_1,x_3)\& \\
& V(x_2,x_3,x_5)\&V(x_3,x_2,x_1)\&V(x_3,x_6,x_2)\& \\
& V(x_3,x_6,x_1)\&L(x_2,x_1,x_5), \\
I_2(\overline{y}) = \ & V(y_3,y_1,y_4)\&V(y_1,y_2,y_3)\&V(y_1,y_5,y_3)\& \\
& V(y_1,y_6,y_2)\&V(y_1,y_6,y_5)\&V(y_1,y_6,y_3)\& \\
& L(y_2,y_1,y_5), \\
I_3(\overline{z}) = \ & V(z_1,z_5,z_3)\&V(z_1,z_3,z_2)\&V(z_1,z_5,z_2)\& \\
& V(z_3,z_1,z_7)\&V(z_3,z_1,z_6)\&V(z_3,z_7,z_4)\& \\
& V(z_3,z_6,z_4)\&V(z_3,z_4,z_1)\&V(z_4,z_2,z_3)\& \\
& V(z_4,z_3,z_8)\&L(z_7,z_6,z_3),
\end{aligned}$$

where $(\overline{x}) = (x_1,\dots,x_6)$, $(\overline{y}) = (y_1,\dots,y_6)$, $(\overline{z}) = (z_1,\dots,z_{10})$.

Pairwise verification of incomplete deducibility (item 2 of the algorithm) for these descriptions gives their maximal common (up to the argument names) sub-formulas.

Formula

$$\begin{aligned}
C_{1,2}(\overline{u}) = \ & V(u_0,u_1,u_2)\&V(u_0,u_3,u_2)\&V(u_0,u_4,u_1)\& \\
& V(u_0,u_4,u_3)\&V(u_0,u_4,u_2)\&L(u_1,u_0,u_3)
\end{aligned}$$

has unifiers $\lambda'_{I_1,C_{1,2}} = (u_0 \to x_1, u_1 \to x_2, u_4 \to x_3, u_2 \to x_4, u_3 \to x_5)$ and $\lambda'_{I_2,C_{1,2}} = (u_0 \to y_1, u_1 \to y_2, u_2 \to y_3, u_3 \to y_5, u_4 \to y_6)$ and corresponds to the image in Fig.5.

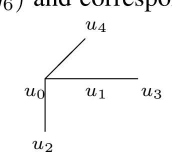



Fig. 5. Image corresponding to the maximal common (up to the argument names) subformula of $I_1$ and $I_2$.

With such denotations

$$\begin{aligned}
I_1(u_0,u_1,u_2,u_3,u_4,x_6) = \ & V(u_1,u_0,u_4)\&V(u_1,u_4,u_3)\& \\
& V(u_4,u_1,u_0)\&V(u_4,x_6,u_1)\& \\
& V(u_4,x_6,u_0)\&C_{1,2}(u_0,\dots,u_4) \\
I_2((u_0,u_1,u_2,y_4,u_3,u_4) = \ & V(u_2,u_0,y_4)\&C_{1,2}(u_0,\dots,u_4).
\end{aligned}$$

Formula

$$\begin{aligned}
C_{2,3}(\overline{v}) = \ & V(v_6,v_2,v_7)\ \&\ V(v_2,v_4,v_6)\ \&\ V(v_2,v_5,v_6) \\
& \&\ V(v_2,v_0,v_4)\ \&\ V(v_2,v_0,v_5)
\end{aligned}$$

has unifiers $\lambda'_{I_2,C_{2,3}} = (v_2 \to y_1, v_4 \to y_2, v_6 \to y_3, v_7 \to y_4, v_5 \to y_5, v_0 \to y_6)$ and $\lambda_{I_3,C'_{2,3}} = (v_0 \to z_1, v_2 \to z_3, v_6 \to z_5, v_5 \to z_6, v_4 \to z_7, v_7 \to z_8)$ and corresponds to the images in Fig. 6.

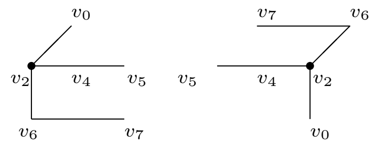



Fig. 6. Images corresponding to the maximal common (up to the argument names) subformula of $I_2$ and $I_3$.

With such denotations

$$\begin{aligned} I_2(v_2, v_4, v_6, v_7, v_5, v_0) = \quad & V(v_2, v_0, v_6)\ \& \\ & L(v_4, v_2, v_5)\ \& \\ & C_{2,3}(v_0, v_2, v_4, v_5, v_6, v_7) \\ I_3((v_0, z_2, v_2, v_6, z_5, v_5, v_4, v_7) = \quad & V(v_2, v_6, v_0)\ \& \\ & V(v_0, z_5, v_2)\ \& \\ & V(v_0, v_2, z_2)\ \& \\ & V(v_0, v_5, z_2)\ \& \\ & V(v_6, z_2, v_2)\ \& \\ & V(v_6, v_2, v_7)\ \& \\ & L(v_4, v_5, v_2)\ \& \\ & C_{2,3}(v_0, v_2, v_4, v_5, v_6, v_7). \end{aligned}$$

Since, by the definition of the predicate $V$, it is true that $V(x, y, z) = \neg V(x, z, y)$, and $I_2(v_2, v_4, v_6, v_7, v_5, v_0)$ contains $V(v_2, v_0, v_6)$, and $I_3((v_0, z_2, v_2, v_6, z_5, v_5, v_4, v_7)$ contains $V(v_2, v_6, v_0)$, $I_2(v_2, v_4, v_6, v_7, v_5, v_0)$ and $I_3((v_0, z_2, v_2, v_6, z_5, v_5, v_4, v_7)$ are inconsistent.

After removing the variables $y_1$ and $z_3$ (instead of which a new variable $v_2$, marked in Fig. 6, was substituted, which led to a contradiction) from $I_2$ and $I_3$, respectively, we obtain their new maximal subformula $C'_{2,3}(v_0, v_1, v_2) = L(v_1, v_0, v_2)$ and unifiers $\lambda'_{I_2,C'_{2,3}} = (v_0 \to y_1, v_1 \to y_2, v_2 \to y_3)$ and $\lambda'_{I_3,C'_{2,3}} = (v_2 \to z_3, v_0 \to z_6, v_1 \to z_7)$. In this case

$$\begin{aligned} I_2(v_0, v_1, v_2, y_4, y_5, y_6) = \quad & V(v_2, v_0, y_4)\ \& \\ & V(v_0, v_1, v_2)\ \& \\ & V(v_0, y_5, v_2)\ \& \\ & V(v_0, y_6, v_1)\ \& \\ & V(v_0, y_6, y_5)\ \& \\ & V(v_0, y_6, v_2)\ \& \\ & C'_{2,3}(v_0, v_1, v_2), \\ I_3(z_1, z_2, v_2, z_4, z_5, v_0, v_1, z_8) = \quad & V(z_1, z_5, v_2)\ \& \\ & V(z_1, v_2, z_2)\ \& \\ & V(z_1, z_5, z_2)\ \& \\ & V(v_2, z_1, v_1)\ \& \\ & V v_2, z_1, v_0)\ \& \\ & V(v_2, v_1, z_4)\ \& \\ & V(v_2, v_0, z_4)\ \& \\ & V(v_2, z_4, z_1)\ \& \\ & V(z_4, z_2, v_2)\ \& \\ & V(z_4, v_2, z_8)\ \& \\ & V(z_4, z_2, z_8)\ \& \\ & C'_{2,3}(v_0, v_1, v_2). \end{aligned}$$

Formula

$$\begin{aligned} C_{1,3}(w_0, \dots, w_6) = \quad & V(w_2, w_4, w_6)\ \&\ V(w_2, w_5, w_6)\ \& \\ & V(w_2, w_0, w_4)\ \&\ V(w_2, w_0, w_5)\ \& \\ & V(w_0, w_1, w_2) \end{aligned}$$

has unifiers $\lambda'_{I_1,C_{1,3}} = (w_2 \to x_1, w_4 \to x_2, w_0 \to x_3, w_6 \to x_4, w_5 \to x_5, w_1 \to x_6)$ and $\lambda'_{I_3,C_{1,3}} = (w_0 \to z_1, w_2 \to z_3, w_6 \to z_4,\ w_1 \to z_5, w_5 \to z_6, w_2 \to z_7)$and corresponds to the images in Fig. 6.


Fig. 7. Images corresponding to the maximal common (up to the argument names) sub-formula of $I_1$ and $I_3$.

Similar to the formula $C_{2,3}$, one can show that if $I_1$ and $I_3$ are expressed in terms of $C_{2,3}$, then they will be contradictory. The variables $x_1$ and $z_3$ (instead of which a new variable $w_2$, marked in Fig. 7, was substituted, which led to a contradiction) should be removed from the unifiers. So, one should take the formula $C'_{1,3}(w_0, w_1, w_2) = L(w_1, w_0, w_2)$ for a common subformula of $I_1$ and $I_3$, and the unifiers $\lambda'_{I_1,C'_{1,3}} = (w_0 \to x_1, w_1 \to x_2, w_2 \to x_5)$ and $\lambda'_{I_3,C'_{1,3}} = (w_2 \to z_3, w_1 \to z_4, w_0 \to z_5)|^{z_3 z_4 z_5}_{w_2 w_1 w_0}$. In this case

$$\begin{aligned} I_1(w_0, w_1, x_3, x_4, w_2, x_6) = \quad & V(w_0, w_1, x_4)\& \\ & V(w_0, w_2, x_4)\& \\ & V(w_0, x_3, w_1)\ \& \\ & V(w_0, x_3, w_2)\ \& \\ & V(w_0, x_3, x_4)\ \& \\ & V(w_1, w_0, x_3)\ \& \\ & V(w_1, x_3, w_2)\ \& \\ & V(x_3, w_1, w_0)\ \& \\ & V(x_3, x_6, w_1)\ \& \\ & V(x_3, x_6, w_0)\ \& \\ & C'_{1,3}(w_0, w_1, w_2), \\ I_3(z_1, z_2, w_2, w_1, w_0, z_6, z_7, z_8) = \quad & V(z_1, w_0, w_2)\ \& \\ & V(z_1, w_2, z_2)\ \& \\ & V(z_1, w_0, z_2)\ \& \\ & V(w_2, z_1, z_7)\ \& \\ & V(w_2, z_1, z_6)\ \& \\ & V(w_2, z_7, w_1)\ \& \\ & V(w_2, z_6, w_1)\ \& \\ & V(w_2, w_1, z_1)\ \& \\ & V(w_1, z_2, w_2)\ \& \\ & V(w_1, w_2, z_8)\ \& \\ & V(w_1, z_2, z_8)\ \& \\ & C'_{1,3}(w_0, w_1, w_2). \end{aligned}$$

In accordance with item 3 of the algorithm in the obtained pairs of unifiers $\lambda'_{I_1,C_{1,2}}$ and $\lambda'_{I_2,C_{1,2}}$, $\lambda'_{I_2,C'_{2,3}}$ and $\lambda'_{I_3,C'_{2,3}}$, $\lambda'_{I_1,C'_{1,3}}$and $\lambda'_{I_3,C'_{1,3}}$ we identify the variables substituted for the same initial variable:

$u_0$ and $w_0$ (are substituted instead of the variable $x_1$),
$u_1$ and $w_1$ (are substituted instead of the variable $x_2$),
$u_2$ and $w_2$ (are substituted instead of the variable $x_4$),
$u_0$ and $v_0$ (are substituted instead of the variable $y_1$),
$u_1$ and $v_1$ (are substituted instead of the variable $y_2$),
$u_2$ and $v_2$ (are substituted instead of the variable $y_3$),

$v_0$ and $w_0$ (are substituted instead of the variable $z_6$),
$v_1$ and $w_1$ (are substituted instead of the variable $z_3$),
$v_2$ and $w_2$ (are substituted instead of the variable $z_7$).
Moreover,we wright
– $\alpha_0$ instead of variables $u_0$, $v_0$ and $w_0$,
– $\alpha_1$ instead of variables $u_1$, $v_1$ and $w_1$,
– $\alpha_2$ instead of variables $u_3$, $v_2$ and $w_2$.
The resulting descriptions correspond to the images shown in Fig. 8.

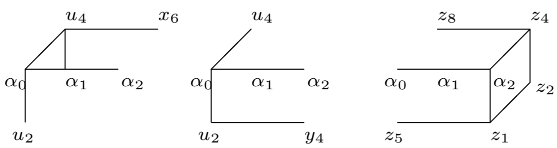



Fig. 8. Images corresponding to $I_1''$, $I_2''$ and $I_3''$.

The result of their "gluing" is shown in Fig. 9.

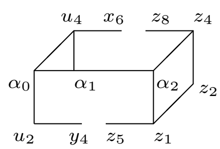



Fig. 9. Image corresponding to the result of "gluing".

It remains to identify some of the original variables ($x_6$, $y_4$, $z_1$, $z_2$, $z_4$, $z_5$, $z_8$,) with other variables to get a description of the object in the database. Such identification can be done by exhaustive search. The description of the object from the base will be obtained by identifying the following pairs of variables:

$$x_6 = z_4,\ y_4 = z_1,\ z_5 = u_2,\ z_8 = u_4.$$

The variables are assigned the following values:

$$z_5 = u_2 = 1,\ y_4 = z_1 = 2,\ z_2 = 3,\ \alpha_0 = 4,$$
$$\alpha_1 = 5,\ \alpha_2 = 6,\ z_8 = u_4 = 7,\ x_6 = z_4 = 8.$$

## VIII. Conclusion

An algorithm for solving a rather complex problem of aggregation non-complete information about a complex structured object described in the predicate calculus language is proposed in the work. Descriptions of such objects by means of binary strings are rather cumbersome and the length of their record is exponential in comparison with the length of the record in the predicate calculus language. This explains the naturalness of the fact that the solution of this problem is exponential under the length of an object description.

The analysis of the complexity estimates of the proposed algorithms obtained in this work makes it possible to impose conditions on the original predicates in order to reduce the real time of the algorithm operation. So, for example, the presence of a large number of initial predicates-features, each of which is quite rare in descriptions, reduces the number of steps of the algorithm.

## Acknowledgment

This research was supported by the St. Petersburg State University, project # 73555239.

## References

[1] R. Duda, P. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

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

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

[4] T. M. Kosovskaya, N. N. Kosovskii, "Polynomial Equivalence of the Problems Predicate Formulas Isomorphism and Graph Isomorphism" *Vestn. St. Petersburg Univ.: Math.*, vol. 52, no. 3, pp. 286 – 292, 2019. (In Russian)

[5] T. Kosovskaya, "Predicate Calculus as a Tool for AI Problems Solution: Algorithms and Their Complexity", 1–20. 2018. Chapter 3 in "Intelligent System", open access peer-reviewed Edited volume. Edited by Chatchawal Wongchoosuk Kasetsart University

[6] T. Kosovskaya . "Implementation of Formula Partial Sequence for Rough Solution of AI Problems in the Framework of the Logic-Predicate Approach", *Proceedings of International Conference Computer Science and Information Technologies*, Yerevan, Armenia, pp. 102-105, 2019.

[7] Kossovskaya T. M. "Partial deduction of predicate formula as an instrument for recognition of an object with incomplete description", *Vestn. St. Petersburg Univ., Series 10: Applied mathematics, Computer science, Control processes*, no. 1, pp. 74–84, 2009. (In Russian)

[8] T.M. Kosovskaya. "Estimating the number of steps that it takes to solve some problems of pattern recognition which admit logical description", *Vestn. St. Petersburg Univ.: Math.*, vol. 40, no. 4, pp. 82–90, 2007. (In Russian)