Hypergraphs for Describing Complex Structured Objects

Tatiana Kosovskaya
St. Petersburg State University
St. Petersburg, Russia
e-mail: kosovtm@gmail.com

***Abstract*—When studying the properties of objects, which consist of many elements that have some of the specified properties and are in specified relationships, a convenient tool to describe them is the use of predicate calculus formulas.**

In such a case, the relations between elements of such an object are set by atomic formulas in which the order of arguments is strictly defined. But if the relation does not depend on the order of the arguments (for example, it is commutative, as the relation "be friends"), then it becomes necessary to write out atomic formulas with all possible argument orders. It is even more difficult to write down such a property of a group of elements in which the number of elements for different groups of the same type is different. For example, a group of "family" elements can include from two to a sufficiently large number of elements.

The use of special-type hypergraphs for the description and analysis of such objects, named hypergraphs of relations, is proposed below. The notion of such a hypergraph's isomrphism is defined. The above-mentioned concept is illustrated by model examples.

***Keywords*— Predicate formulas, complex structured object, hypergraph of relations.**

I. Introduction

At the end of the 20th century, many works based on the use of predicate calculus appeared to solve image recognition problems in which recognizable objects are complex, that is, they consist of elements with specified properties and are in specified relationships with a fixed number and a fixed order of arguments [1]–[3].

Such problems are NP-complete or NP-hard [4]. Objections to the use of predicate formulas due to the exponential complexity of the algorithms involved can be answered as follows.

Information recorded using predicate formulas can be simulated by binary strings. In this case, the problem of recognizing whether one of the binary strings is a substring of the other is solved in several steps that do not exceed a polynomial in the lengths of these strings records. That is, instead of an NP-hard problem, you can solve a polynomial one. But after all, the length of the record of the binary string modeling the predicate formula is exponentillay greater than the length of the record of the original predicate formula. Therefore, objections regarding the NP-difficulty of problems are untenable in this case.

Within the framework of the logic-predicate approach to solving Artificial Intelligence problems, the following problems were solved:

– constructing a level description of objects that significantly reduces the computational complexity [5];

–building a fuzzy logic-predicate network that allows recognizing objects with a given degree of confidence [6];

– building a logic-predicate network that changes its configuration in the process of further retraining [6];

– creation of the ontology [7],

and some others.

The main notions that allow us to solve theese problems are the notions of *elementary conjunctions isomorphism* [8] and *maximal common subformula*.

A significant disadvantage of the logical-predicate approach is that

– the relations between the elements can be commutative,

– in predicate calculus, all predicate symbols have a fixed dimension.

In the first case, for the l-ary relation p, if $p(x_1,\dots,x_l)$ is true, then for every permutation of the arguments $x_{i_1},\dots,x_{i_l}$ true $p(x_{i_1},\dots,x_{i_l})$, that is, instead of a single atomic formula, $p(x_1,\dots,x_l)$ in the descriptions, you need to write down a disjunction of $l!$ formulas of the form $p(x_{i_1},\dots,x_{i_l})$. If there are t occurrences of commutative predicates in the description, the number of arguments in which is $l_1,\dots,l_t$, then you will have to write down $l_1!\cdot\dots\cdot l_t!$ formulas of the same type.

In the second case, instead of a single predicate p of fixed arity, it is needed to introduce predicates $p^2,\dots,p^k$ with the number of arguments $2,\dots,k$, where k is the largest number of elements that can be in this relation.

To overcome these disadvantages, the use of a special type of hypergraphs, called hypergraphs of relations, is proposed below.

II. Necessary Definitions

Definition 1. *A complex structured object (CSO) is an object $\omega=\{\omega_1\dots\omega_t\}$ the elements of which have specified properties (satisfy unary predicates) and are in the specified relationships (satisfy multi-place predicates) $p_1\dots p_n$.*

Definition 2. *Description of a CSO $S(\omega)$ is an elementary conjunction of atomic formulas with predicates $p_1\dots p_n$, which is the maximum in the number of literals, and is true for ω.*

Definition 3. *Two elementary conjunctions of atomic predicate formulas* $P(a_1, \ldots, a_m)$ *and* $Q(b_1, \ldots, b_m)$ *are called isomorphic*

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

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

The resulting substitutions $|^{x_1}_{a_{i_1}} \cdots {}^{x_m}_{a_{i_m}}$ [1] *and* $|^{x_1}_{b_{j_1}} \cdots {}^{x_m}_{b_{j_m}}$ *are called unifiers of formulas* $P(a_1, \ldots, a_m)$ *and* $Q(b_1, \ldots, b_m)$ *with the formula* $R(x_1, \ldots, x_m)$ *respectively.*

In the above definition, one could do without introducing the formula $R(x_1, \ldots, x_m)$ into it, as it is done in the definition of isomorphic graphs. But, firstly, my mathematical education does not allow me to substitute anything in the formula instead of constants. Secondly, in the future, this formula will act precisely as a formula with variables that sets the common property of two CSOs.

Let Ω_1 be a subset of all elements.

Definition 4. *A disjunction of elementary conjunctions with variables for arguments*

$$S(\Omega_1) = A_1^1(\overline{x}_1^1) \vee \cdots \vee A_1^{k_1}(\overline{x}_1^{k_1}),$$

which is true for objects of the set Ω_1 *and only for them is called a description of* $S(\Omega_1)$ *of the set* Ω_1*.*

In such a case, the problem of verifying if $\omega \in \Omega_1$ is reduced to checking the logical sequence of

$$S(\omega) \Rightarrow \exists \overline{x}_{\neq} S(\Omega_1).$$ [2]

More precisely, to checking for each $i = 1, \ldots, k_j$ logical sequence

$$S(\omega) \Rightarrow \exists \overline{x}_{j\neq} A_j^i(\overline{x}_j^i). \qquad (1)$$

In fact, the logical sequence (1) means isomorphism of an elementary conjunction $A_j^i(\overline{x}_j^i)$ to some subformula of a conjunction from $S(\Omega_1)$ [9].

With this formulation of the problem, each complex structured object can be defined by a graph. The elements of the object $\omega_1, \ldots, \omega_n$ correspond to vertices with the same names. The predicate symbols $p_1, \ldots, p_k$ correspond to vertices with the same names.

An atomic formula $p_i(a_{n_1}, ...a_{n_k})$ with a k-ary predicate symbol p_i corresponds to an oriented path from vertex p_i through vertices $a_{n_1}, ...a_{n_k}$.

[1] The notation $P|^{x_1}_{a_{i_1}} \cdots {}^{x_m}_{a_{i_m}}$ is used to replace all free occurrences in the formula P of variables $x_1, \cdots, x_m$ with constants $a_1, \cdots, a_m$, respectively.

[2] The notation $\overline{x}_{\neq}$ is used to mark that the variables have various values.

III. Hypergraphs of Relations

To overcome the difficulties described in the Introduction, it is proposed to use special-type hypergraphs, which we will call hypergraphs of relations.

Let the properties and relations of $p_1, \ldots, p_n$ are defined on the set $\omega = \{\omega_1, \ldots, \omega_t\}$. Moreover, relations can be commutative (sometimes only for a part of arguments) and have an arbitrary number of arguments.

For every element of the set ω, it is known what properties are fulfilled, and in what relationships these elements are.

The vertices of a hypergraph of relations are

– vertices with the names of properties and relations, which will be called **pedicate vertices**;

– vertices with the names of the elements of ω, which will be called **objective vertices**.

If p is the **property** name of an element ω_i (atomic formula $p(\omega_i)$), then the vertex with the name p is connected by an oriented edge with the vertex named ω_i for every name of an element that has this property

$$p \to \omega_i \ .$$

If p is the name of a **noncommutative** k**-ary relation** (atomic formula $p(\omega_{i_1}, \ldots, \omega_{i_k})$), then a predicate vertex p is connected by an oriented edge with an oriented sequence of objective vertices, the elements of which are in the relation p

$$p \to (\omega_{i_1} \to \cdots \to \omega_{i_k}) \ .$$

If p is the name of a **commutative relation of arbitrary dimension**, then a vertex named p is connected by an oriented edge with the set of vertices, containing all names of the object elements that are in this relation

$$p \to \{\omega_{i_1}, \cdots, \omega_{i_k}\} \ .$$

If p is the name of a **noncommutative** k**-ary relation, which can have one of the elements of a set specific to that position**, then the vertex named p is connected by an oriented edge with a sequence of sets whose elements are in this relation

$$p \to (\{X_1\} \to \cdots \to \{X_k\}) \ .$$

The last of the hyperedges described corresponds to the formula

$$\forall x_1 \ldots x_k (x_1 \in \{X_1\} \& \ldots \& x_k \in \{X_k\} \Rightarrow p(x_1 \ldots x_k)).$$

Similarly, we can define the case when p is the name of a **commutative** k**-ary relation, in which at each position there may be one of the elements of a set specific to that position**, then the vertex named p is connected by an edge to a sequence of sets, the elements of which are in this relation

$$p \to \{\{X_1\}, \cdots, \{X_k\}\} \ .$$

The following definition of isomorphism for hypergraphs of relations differs from traditional ones for graphs and hypergraphs. But it is equivalent to that for hypergraphs. The nessesity of introducion a new hypergraph with variables as vertex names is due to the fact that it is this hypergraph that will determine the general properties of objects.

Definition 5. *Two hypergraphs of relations G_1 and G_2 are called isomorphic if there exists such a hypergraph of relations H with variables as vertex names and such substitutions λ_1 and λ_2 of the names of the object elements in G_1 and G_2 instead of the variables of the graph H, the results of applying these substitutions are graphs G_1 and G_2.*

Such substitutions will be called the unifiers of the hypergraphs G_1 and G_2 with the hypergraph H.

It is easy to prove that checking the isomorphism of graphs of relations is polynomially equivalent to the ¡¡open¿¿ problem *Graph isomorphism.*

An example of isomorphic hypergraphs of relations

G_1	G_2	H
$p_1 \to a$	$p_1 \to \beta$	$p_1 \to x$
$p_1 \to c$	$p_1 \to \gamma$	$p_1 \to y$
$p_2 \to b$	$p_2 \to \alpha$	$p_2 \to z$
$p_2 \to d$	$p_2 \to \delta$	$p_2 \to u$
$p_3 \to (a, \{c, b\})$	$p_3 \to (\gamma, \{\beta, \delta\})$	$p_3 \to (\{X\}, \{Y\})$

The following substitutions are the unifiers of the hypergraph H with the hypergraphs G_1 and G_2.

$$\lambda_1 = |^x_a, |^y_b, |^z_c, |^u_d, |^{\{X\}}_{\{a\}}, |^{\{Y\}}_{\{c,b\}},$$

$$\lambda_2 = |^x_\beta, |^y_\gamma, |^z_\alpha, |^u_\delta, , |^{\{X\}}_{\{\gamma\}}, |^{\{Y\}}_{\{\beta,\delta\}}.$$

IV. Setting the Problem of Recognition in the Language of Hypergraphs of Relations

Definition 6. *A set $\omega = \{\omega_1, \dots, \omega_n\}$, on which the properties of these objects and the relations between groups of objects are defined, is called a complex object (CO).*

Definition 7. *A hypergraph of relations $S(\omega)$, the set of predicate vertices of which contains the names of all properties of objects and the relations between them, and the set of objective vertices contains the names of all elements of this object, is called a description of CO ω.*

Let Ω_1 be a subset of all elements.

Definition 8. *A description $S(\Omega_1)$ of the set Ω_1 is such a collection of hypergraphs of relations with variables for object names that for each object from the class in this collection there is a hypergraph isomorphic to the graph defining the description of an object.*

The problem of checking whether a recognized object belongs to a given class is to check whether there is a hypergraph in the class description that is isomorphic to a subgraph that defines the object description.

V. Maximal Common Property of Two Hypergraphs of Relations

Definition 9. *A hypergraph of relations H with variables for the names of objective vertices defines a common property of two hypergraphs of relations G_1 and G_2 if there exist such subgraphs G'_1 and G'_2 of G_1 and G_2, respectively, that are isomorphic to H.*

Definition 10. *A hypergraph of relations H with variables for the names of objective vertices defines a maximal common property of two hypergraphs of relations G_1 and G_2 if it defines their common property, but after the addition of any vertex or any edge to H, the obtained hypergraph of relations is not isomorphic to G_1 or G_2.*

Introduction of the notion of a maximal common property of two hypergraphs of relations allows us to construct a description of a class according to a training set. It is sufficient to pairwise extract the maximal common properties of the training set descriptions.

According to such a class description, it is possible to solve the problem of whether a concrete object described in the terms of such hypergraphs belongs to a described class.

VI. Model Example of Description and Recognition by Means of Hypergraphs of Relations

Let an object be the family of Johnson. It contains 6 members: father John, mother Mary, two sons, Peter and Basil, daughter Katherine.

Two properties and one relation are defined as
$m(x)$ – "x is a man",
$w(x)$ – "x is a woman",
$p(\{X\}, \{Y\})$ – "the element of the set $\{X\}$ is a parent of the element of the set {Y}".

In such a case, the description of the Johnson family can be represented by a hypergraph of relations. The predicate vertices of such a hypergraph of relations are m, w, p. The names of family members J, M, P, B and K are in the set of objective vertices.

The edges have the form

$$m \to J, \quad m \to P, \quad m \to B,$$

$$w \to M, \quad w \to K,$$

$$p \to (\{J, M\}, \{P, B, K\})).$$

The last edge means that every element of the set $\{J, M\}$ is the parent of every element of the set $\{P, B, K\}$.

Description of the class "families with brothers"[3] corresponds to the fact that the formula $\exists x x_1 x_2 (p(x, \{x_1, x_2\}) \& m(x_1) \& m(x_2)$ follows from the description of a particular family.

This formula (without quantifiers of existence) corresponds to a hypergraph with variables x, x_1, x_2 as vertex names and with edges

$$m \to x_1, \quad m \to x_2,$$

$$p \to (x, \{x_1, x_2\}).$$

[3]Below, the letters x, y and z are used for the names of variables for objects (perhaps with indexes), and $\{X\}$, $\{Y\}$ and $\{Z\}$ (perhaps with indexes) are used for the names of sets of objects.

It is obvious that it is a subgraph of a hypergraph defining a description of the Johnson family with the following unifiers (values of variables) $x = J$, $x_1 = P$, $x_2 = B$.

There are other unifiers $x = M$, $x_1 = P$, $x_2 = B$.

ACKNOWLEDGMENT

The author acknowledges Saint-Petersburg State University for a research project 121061000159-6.

The author is grateful to Hasmik Sahakyan for valuable comments on the definitions of hypergraphs.

REFERENCES

[1] R.O. Duda, O.E. Hart, D.G. Stork, *Pattern Classification*, Second Edition. Wiley, New York, 2000.

[2] N.J. Nilson, *Problem-solving methods in Artificial Intelligence*, McGraw-Hill, New York, 1971.

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

[4] T.M. Kosovskaya, “Proofs of estimates of the number of steps for solving some image recognition problems with logical descriptions”, *Vestnik of St. Petersburg university. Mathematics.*, issue 4, pp. 82–90, 2007. (In Russian)

[5] T.M. Kosovskaya, “Level descriptions of classes for decreasing step number of pattern recognition problem solving described by predicate calculus formulas”, *Vestnik of St. Petersburg university. Applied mathematics. Computer science. Control processes*, issue. 1, pp. 64–72, 2008. (in Russian)

[6] Tatiana Kosovskaya. Fuzzy Neural Networks that Change their Configuration, ISSN 0361-7688, Programming and Computer Software, Springer Nature, vol. 50, Suppl.1, pp. S10 – S17, 2024.

[7] T. M. Kosovskaya, and N. N. Kosovskii, “Extraction common properties of objects for creating logical ontologies”, *Vestnik of St. Petersburg State University. Applied mathematics. Computer science. Control processes*, vol. 18, no. 1, pp. 37–51 2022. (in Russian) https://doi.org/10.21638/11701/spbu10.2022.103

[8] T. M. Kosovskaya and J. Zhou, Algorithms of Isomorphism of Elementary Conjunctions Checking, Pattern Recognition and Image Analysis, vol. 34, no. 1, pp. 102–109, 2024. DOI: 10.1134/S1054661824010103

[9] T.M. Kosovskaya, J. Zhou, Algorithm for Extraction Common Properties of Objects Described in the Predicate Calculus Language with Several Predicate Symbols, ISSN 0361-7688, Programming and Computer Software, Springer Nature, vol. 50, Suppl.1, pp. S1 – S9, 2024.