Extraction of isomorphic subformulas as a tool for logic ontology construction

T. Kosovskaya
Saint Petersburg State University, University emb., St. Petersburg,199034, Russia
E-mail: kosovtm@gmail.com

Abstract. A way of constructing a logic ontology for complex structured objects is considered in the paper. The descriptions of such objects are given in the predicate calculus language. An algorithm for extracting subsets of objects with the same properties and establishing their relative position is described. To construct such an ontology, an algorithm for extracting the largest subformula isomorphic to subformulas of all descriptions of objects from the class is used.

Introduction
The concept of ontology as a branch of philosophy has existed since ancient times. At the same time, in philosophy, ontology is understood as abstractions that take into account some features of objects (or relations between parts of these objects) and ignore others. With the development of computer technology and informatics, this term was transferred to the construction of a detailed formalization of a certain area of knowledge, taking into account structure of the objects under study and relationships between their parts. The creation of a domain ontology is one of the ways of knowledge representation for their use in artificial intelligence systems and knowledge extraction from databases [1 – 3].

For a mathematician, an ontology is a directed graph, the nodes of which define certain concepts (classes of objects). If the oriented edge \((N_i, N_j)\) connects two concepts \(N_i\) and \(N_j\), then all objects included in the concept \(N_j\) are also included in the concept \(N_i\).

A logical ontology is usually understood as an ontology over complex structured objects. Elements of such objects have predetermined features and are in predetermined relationships. Moreover, for all objects included in one concept, there is a subset of its elements that satisfy the same formula.

A convenient language for description such objects is the predicate calculus language [4]. Earlier [5] the author has introduced the concept of isomorphic elementary conjunctions of literals. This relation differs from that of equivalence. In fact, it means that two elementary conjunctions of literals coincide up to the names of their arguments and the order of the literals in them and define the same property of their arguments.

An algorithm for extracting a formula isomorphic to subformulas of descriptions of a set of objects is described in [5]. This algorithm is the basis for creating a logical ontology.

The following procedure for constructing a logical ontology is considered in this paper. A set of complex structured objects is given. The descriptions of these objects are presented in the form of elementary conjunctions of predicate formulas that specify the properties of the objects' elements and the relations between them. This set defines the initial concept for a logical ontology.
For each set of objects defining the already constructed concept of ontology, it is required to select the maximal formulas that are (up to variable names) subformulas of descriptions of a group of objects from this concept. The sets of objects that satisfy these subformulas define new concepts of the logical ontology that are daughter ones of that from which they are obtained.

The proposed paper is devoted to the description of main definitions and the presentation of the algorithm scheme, which make it possible to implement the just described procedure for constructing a logical ontology.

### Main definitions

Before giving a statement of the problem, it is necessary to formulate the basic definitions.

**Definition 1.** Elementary conjunctions \( P(a_1,\ldots,a_m) \) and \( Q(b_1,\ldots,b_n) \) are called *isomorphic* if there are an elementary conjunction \( R(x_1,\ldots,x_m) \) and substitutions of arguments \( a_1,\ldots,a_m \) and \( b_1,\ldots, b_n \) instead of the variables \( x_1,\ldots,x_m \) such that the results of these substitutions \( R(a_1,\ldots,a_m) \) and \( R(b_1,\ldots, b_n) \) coincide with formulas \( P(a_1,\ldots,a_m) \) and \( Q(b_1,\ldots,b_n) \), respectively, up to the order of literals.

The substitutions \( (a_1,\ldots,a_m) \rightarrow (x_1,\ldots,x_m) \) and \( (b_1,\ldots,b_n) \rightarrow (x_1,\ldots,x_n) \) are called *unifiers* of \( R(x_1,\ldots,x_m) \) with \( P(a_1,\ldots,a_m) \) and \( Q(b_1,\ldots,b_n) \) and are denoted as \( U_{R,P} \) and \( U_{R,Q} \), respectively [5].

For example, formulas

\[
P(a,b,c) = p(a,b) \& \neg p(b,c) \& q(a) \& q(c),
Q(a,b,c) = \neg p(a,b) \& p(c,a) \& q(b) \& q(c)
\]

are isomorphic because they can be unified with formula

\[
R(x,y,z) = p(x,y) \& \neg p(y,z) \& q(x) \& q(z)
\]

by means of the unifiers \( U_{R,P} = (a,b,c) \rightarrow (x,y,z) \) and \( U_{R,Q} = (a,b,c) \rightarrow (y,z,x) \).

Formulas \( a < b \& b < c \& a \neq c \) and \( b < a \& a < c \& b \neq c \) represent a more simple example of isomorphic formulas. They are isomorphic to the relation \( x < y \& y < z \& x \neq z \) “the number \( y \) belongs to the interval \( (x, z) \)”.

It is easy to prove that the isomorphism relation is the equivalence relation.

**Definition 2.** Elementary conjunction \( C(x_1,\ldots,x_n) \) is called a *common* up to the names of arguments *subformula* of two elementary conjunctions \( A(a_1,\ldots,a_m) \) and \( B(b_1,\ldots,b_n) \) if it is isomorphic to some subformulas \( A'(a'_1,\ldots,a'_m) \) and \( B'(b'_1,\ldots,b'_n) \) of \( A(a_1,\ldots,a_m) \) and \( B(b_1,\ldots,b_n) \), respectively.

The unifiers of \( C(x_1,\ldots,x_n) \) with \( A(a_1,\ldots,a_m) \) and \( B(b_1,\ldots,b_n) \) will be denoted as \( U_{C,A}^* \) and \( U_{C,B}^* \), respectively.

**Definition 3.** Elementary conjunction \( C(x_1,\ldots,x_n) \) is called a *maximal* up to the names of arguments *subformula* of two elementary conjunctions \( A(a_1,\ldots,a_m) \) and \( B(b_1,\ldots,b_n) \) if it is a common up to the names of arguments subformula of two elementary conjunctions \( A(a_1,\ldots,a_m) \) and \( B(b_1,\ldots,b_n) \), but it is not isomorphic to any subformula of any of them after adding any literal to it.

For example, let

\[
A(x,y,z) = p(x) \& p(y) \& p(z) \& q(x, y) \& r(x, z),
B(x,y,z) = p(x) \& p(y) \& p(z) \& q(x, z) \& r(x, z).
\]

Every its literal is their common subformula. But only one literal \( r(x, z) \) is their maximal common subformula. The formula

\[
P(u,v) = p(u) \& p(v) \& q(u, v)
\]

is their maximal common up to the names of variables subformula with the unifiers \( U_{P,A}^* = (x, y) \rightarrow (u, v) \) and \( U_{P,B}^* = (x, x) \rightarrow (u, v) \) because

---

1 The words “up to the names of arguments” would be omitted below.
The problem setting
Let \( N_0 \) be a set of complex structured objects in the form \( \omega = \{\omega_1, ..., \omega_k\} \). The predicates \( p_1, ..., p_n \) defined on the elements of \( \omega \) specify the properties of its elements and the relations between them. Description \( S(\omega) \) of an object \( \omega \) is an elementary conjunction of all atomic formulas with predicates \( p_1, ..., p_n \) which are true on \( \omega \).

For every subset \( N \subset N_0 \) the description \( S(N) \) of elements from \( N \) is a disjunction of non-isomorphic elementary conjunctions, every one of which is isomorphic to the description of some object from \( N \). If \( \omega \in N \) then every object \( \omega' \) such that \( S(\omega') \) is isomorphic to \( S(\omega) \) also belongs to \( N \) and is its representative. Below we suppose that \( N \) contains only representatives of such classes of equivalence. That is why we can write

\[
S(N) = \lor_{\omega \in N} S(\omega).
\]

Definition 4. A formula which is a maximal common subformula for every pair of descriptions of objects from \( N \) is called the maximal common property (MCP) of elements from the set \( N \).

Let \( N_{k_1}, ..., N_{k_m} \) (possibly intersecting) be subsets of the set \( N_0 \). It is required to construct an oriented graph whose node with zero indegree is labeled by the set \( N_0 \) and corresponds to the set of non-isomorphic formulas, each of which is isomorphic to some description of an object from \( N_0 \).

Moreover, if there exist oriented edges from a node labeled by \( N_k \) to nodes labeled by \( N_{k_1}, ..., N_{k_r} \), then
- \( N_k \) is the union of the sets \( N_{k_1}, ..., N_{k_r} \);
- for each \( i = 1, ..., r \), each formula in the description of objects from \( N_{k_i} \) is isomorphic to one of the formulas in the description of \( N_k \), i.e. objects from \( N_{k_i} \) have all the properties common to all objects from \( N_k \);
- if \( i \neq j \), then the MCP of objects from \( N_{k_i} \) and \( N_{k_j} \) are different.

In addition, we will label the nodes of the graph (except of \( N_0 \)) not by the names of the sets, but by formulas that define the MCP of the elements of the corresponding set.

To solve this problem, one can use an algorithm for extracting a maximal formula isomorphic to subformulas of two elementary conjunctions described in [5]. The idea of extracting a maximal formula isomorphic to subformulas of two elementary conjunctions is based on the notion of incomplete sequent and is presented in [6]. This idea is as follows.

Let \( \tilde{x} \) and \( \tilde{y} \) be lists of variables of elementary conjunctions \( A (\tilde{x}) \) and \( B (\tilde{y}) \). Sequentially deleting one of the literals of the formula \( B(\tilde{y}) \) and receiving the formula \( B'(\tilde{y}') \), we check the logical sequent \( A(\tilde{x}) \Rightarrow \exists \tilde{y}', B'(\tilde{y}')2 \). If the logical sequent is fulfilled, the resulting formula \( B'(\tilde{y}') \) is the MCP of formulas \( A (\tilde{x}) \) and \( B (\tilde{y}) \). When using constructive methods for checking this sequent (sequential predicate calculus, the resolution method for predicate calculus, or exhaustive search of strings of values from \( \tilde{x} \) for \( \tilde{y}' \)), the found values \( \tilde{x}' \) for \( \tilde{y}' \) will define the unifier \( U^*_{\tilde{x}', \tilde{y}'} \).

It should be noted that, firstly, an inefficient exhaustive search of formulas which may turn out to be the maximal common subformula is performed here, and, secondly, only the MCP of the greatest length is extracted. These disadvantages are eliminated in the algorithm described in [5].

---

2 \( \exists \tilde{y}' \) means «there are pairwise different values for the list of variables \( \tilde{y}' \)».

3 Although these methods give the same results, they differ significantly in their computational complexity. They are all exponentially difficult (the problem is NP-complete), but for logical methods, the exponent contains the number of literals of the formula \( B'(\tilde{y}') \), and for an exhaustive algorithm, the exponent contains the number of its variables.
Algorithm of ontology construction.
1. For every set \( N_i \) extract MCP of its elements \( P_i \) for every pair \( N_i, N_j \).
2. Extract MCP \( (N_i, N_j) \) for every pair \( N_i, N_j \) from \( N_0, N_1, ..., N_m \).
   - If \( MCP(N_i, N_j) = P_i \) then \( N_j \subset N_i \). It means that an oriented edge from \( N_i \) to \( N_j \) is presented in the graph defining the ontology.
   - If \( MCP(N_i, N_j) = P \) with \( P \) not isomorphic to neither of \( P_0, P_1, ..., P_m \) then there is no relation of inclusion between \( N_i \) and \( N_j \).
   - If \( MCP(N_i, N_j) = \emptyset \) then there is no relation of inclusion or intersection between \( N_i \) and \( N_j \).
3. Draw oriented edges from the node labeled by \( N_i \) to all nodes labeled by \( N_j \), for which \( MCP(N_i, N_j) = P_i \).
4. If there is an edge \( (N_i, N_j) \) and an oriented path from \( N_i \) to \( N_j \), then the edge \( (N_i, N_j) \) is removed.

Note that the emergence of a “vicious circle” is impossible, since any oriented path \( (N_{i_1}, ..., N_{i_k}) \) in the graph corresponds to the inclusions \( N_{i_1} \subset ... \subset N_{i_k} \).
In addition, it is possible to construct an ontology in which sets appear that are not represented in the input data. These are the sets MCP of which are formulas \( P \), which appear in item 1 of the algorithm and are not isomorphic to any of the formulas \( P_0, P_1, ..., P_m \).

Model example
Consider a model example of constructing an ontology of convex quadrangles.
In this example, a quadrangle is a sequence of names of four points connected by segments in a cycle.

Predicates \( p, e, r \) are defined on the set of quadrangles:
\[
p(x,y,z,u) \iff \text{“the segment } [x,y]\text{ is parallel to the segment } [z,u]\text{”},
\]
\[
e(x,y,z,u) \iff \text{“the length of the segment } [x,y]\text{ is equal to the length of the segment } [z,u]\text{”},
\]
\[
r(x) \iff \text{“the angle at the corner } x\text{ is right one”}.
\]
For clarity, we will write
\[
a \text{ instead of } [x,y],
\]
\[
b \text{ instead of } [y,z],
\]
\[
c \text{ instead of } [z,u],
\]
\[
d \text{ instead of } [u,x],
\]
\[
A \parallel B \text{ to write that “segment } A\text{ is parallel to segment } B\text{” and}
\]
\[
A = B \text{ to write that “the length of segment } A\text{ is equal to the length of segment } B\text{”}.
\]
Representatives of the sets “Parallelogram”, “Rectangle”, “Rhombus”, “Square”, “Trapezium”, “Isosceles trapezoid”, “Deltoid” are given as representatives of the set of convex quadrangles. Below the following notations will be used: \( P \) for “Parallelogram”, \( RP \) for “Rectangle”, \( R \) for “Rhombus”, \( S \) for “Square”, \( T \) for “Trapezium”, \( IT \) for “Isosceles trapezoid”, \( D \) for “Deltoid”.

![Fig. 1. Denotations of apexes and sides of quadrangles.](image-url)
It is supposed in this example that every apex and every side of quadrangles are denoted in formulas presenting MCP of the mentioned sets as it is shown in Fig. 1. It permits, instead of finding the maximum common up to the names of the arguments subformula, to find the maximum subword in the notations of the corresponding formulas.

Having selected the MCP of each set elements, formulas that determine the corresponding properties of quadrangles are obtained.

\[ P : (x,y,z,u) \iff a \land c \land b \land d \land a = c \land b = d. \]

\[ RP : (x,y,z,u) \iff a \land c \land b \land d \land a = c \land b = d \land R(x) \land R(y) \land R(z) \land R(u). \]

\[ R : (x,y,z,u) \iff a \land c \land b \land d \land a = c \land b = d \land a = b. \]

\[ S : (x,y,z,u) \iff a \land c \land b \land d \land a = c \land b = d \land a = b \land R(x) \land R(y) \land R(z) \land R(u). \]

\[ T : (x,y,z,u) \iff b \land d \land c \land a = c. \]

\[ IT : (x,y,z,u) \iff b \land d \land c \land a = c. \]

\[ D : (x,y,z,u) \iff a = d \land b = c. \]

Let us single out the pairwise MCP of the formulas that define these properties of the quadrangles.

\[ MCP(P,RP) = p(x,y,z,u). \]

\[ MCP(P,R) = p(x,y,z,u). \]

\[ MCP(P,S) = p(x,y,z,u). \]

\[ MCP(P,T) = b \land d. \]

\[ MCP(P,IT) = b \land d \land a = c. \]

\[ MCP(P,D) = \emptyset. \]

\[ MCP(RP,R) = p(x,y,z,u). \]

\[ MCP(RP,S) = p(x,y,z,u). \]

\[ MCP(RP,T) = b \land d. \]

\[ MCP(RP,IT) = b \land d \land a = c. \]

\[ MCP(RP,D) = \emptyset. \]

\[ MCP(R,S) = p(x,y,z,u). \]

\[ MCP(R,T) = b \land d. \]

\[ MCP(R,IT) = b \land d \land a = c. \]

\[ MCP(R,D) = \emptyset. \]

\[ MCP(S,R) = R(x,y,z,u). \]

\[ MCP(S,T) = b \land d. \]

\[ MCP(S,IT) = b \land d \land a = c. \]

\[ MCP(S,D) = D(x,y,z,u). \]

\[ MCP(T,IT) = T(x,y,z,u). \]

\[ MCP(T,D) = \emptyset. \]

The ontology of such quadrangles is shown in Fig. 2. Edges from the node labeled as CONVEX QUADRANGLES to the other nodes are deleted because there are oriented paths from it to every other node.

When constructing this ontology, the sets of quadrangles were not taken into account, which were not specified initially, but appeared as a result of pairwise extraction of MCP. These are the sets \( M_i = \)
\{(x,y,z,u): \ b \ || \ d\} \text{ and } M_2 = \{(x,y,z,u): \ b \ || \ d \ & \ a = c\}. \text{ Obviously, the } \text{MCP}(M_1, \ M_2) = b \ || \ d. \text{ That is, } M_2 \subset M_1.

\[
\text{CONVEX QUADRANGLES}
\]

\[
\begin{align*}
P \\
D \\
R \\
S
\end{align*}
\]

\[
\begin{align*}
T \\
IT
\end{align*}
\]

**Fig. 2.** Ontology of convex quadrangles.

In this case, the ontology obtained in a similar way will look as it is shown in Fig. 3.

\[
\text{CONVEX QUADRANGLES}
\]

\[
\begin{align*}
M_1 \\
M_2 \\
P \\
D \\
R \\
S
\end{align*}
\]

\[
\begin{align*}
T \\
IT
\end{align*}
\]

**Fig. 3.** Ontology of convex quadrangles, taking into account the newly extracted MCP.

**Conclusion**

The proposed algorithm for constructing a logical ontology is exponentially complex. This is due to the fact that the problem of extracting a maximal formula isomorphic to subformulas of two elementary conjunctions is NP-hard. However, its use, like the use of any other ontology, allows you to quickly classify the object under study.

The exponential complexity of the algorithm was eliminated due to the fact that conventions were adopted on the order of the arguments in the formulas defining the MCP of two sets in the example given in the article. The main attention in the example was paid to the process of constructing a graph defining an ontology, based on the fact that for two sets their maximum common up to variable names formula is a formula that defines the MCP of elements of one of the sets. Unfortunately, such agreements are not possible for all sets for which the ontology is being built.

In the example, it was also shown that in the process of extraction MCP of two sets, sets may appear that were not given in advance. This is natural, since an example was taken, the ontology for which has existed for more than 1000 years and the extracted sets \(M_1\) and \(M_2\) were not interesting for researchers. However, the appearance of such sets suggests that it is possible to consider a problem in which subsets of the set of objects under study \(N_0\) are not given, but they should be selected.

The proposed approach allows solving such a problem. But when solving it, a huge number of subsets will be allocated. Calling in a subject matter expert can separate “important” subsets from “irrelevant” ones.
It is interesting to consider the case of multivalued predicates describing the objects under study. Expression of multivalued predicates \( q(\vec{x}) = a \) into binary ones can be done by introducing a binary predicate \( p(\vec{x}; a) \) with an additional argument \( a \). The problem is that the additional argument \( a \), unlike the arguments \( \vec{x} \), is not a variable for an object element.

References

[1] Natalya F Noy and Deborah L McGuinness 2001 Ontology Development 101: A Guide to Creating Your First Ontology Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880

[2] Beniaminov E M 2008 Some problems of widespread introduction of ontologies in IT and directions of their solutions Proceedings of the Symposium "ONTOLOGICAL MODELING". M.: IPI RAS pp.71-82 (In Russian)

[3] Diachenko O O and Zagorulko Yu A 2013 An approach to collective development of ontologies and knowledge bases Knowledge - Ontology - Theories. Materials of the All-Russian conference with international participation Publishing (Institute of Mathematics S.L. Sobolev SB RAS) pp. 141 – 149 (In Russian)

[4] Kosovskaya T 2018 Predicate Calculus as a Tool for AI Problems Solution: Algorithms and Their Complexity pp. 1 – 20. Chapter 3 in: Intelligent System. Open access peer-reviewed Edited volume. https://www.intechopen.com/books/intelligent-system/predicate-calculus-as-a-tool-for-ai-problems-solution-algorithms-and-their-complexity

[5] Kosovskaya T M and Petrov D A 2017 Extraction of a maximal common sub-formula of predicate formulas for the solving of some Artificial Intelligence problems Vestnik of Saint-Petersburg University. Series 10. Applied mathematics. Computer science. Control processes issue 3 pp. 250 – 263 (In Russian)

[6] Kosovskaya T M 2014 An approach to the construction of a level description of classes by means of a predicate calculus language SPIIRAS Proceedings 3(34) pp. 204 – 217 (In Russian)