© A. G. IVASHKO, A. V. GRIGORYEV

ivashco@mail.ru, 107th@mail.ru

UDC 510.5; 510.22; 519-6

REDUCTION OF INTERPRETATION SPACE THROUGH DISCOVERING ISOMORPHIC CONCEPTS

ABSTRACT. The article considers an algorithm for logic inference on knowledge bases described in OWL language which is based on description logic formalism. The considered algorithm has an exponential computational complexity which makes it hard to practice. By now, certain optimizations have been developed to reduce the work time of the algorithm, but nevertheless some real knowledge bases cannot be classified in satisfactory time. The main body of the article is devoted to developing a new method to reduce the number of interpretations under consideration by tableau algorithm. The article demonstrates the proof of correctness for the given method and offers estimation of the number of non-considered interpretations. The authors briefly describe the structures, used for concepts representation, and determine the modification of Edmundson's method for the search of isomorphic substructures in concept descriptions. The developed method is implemented in TReasoner system. At the end of the article testing results are presented. The testing was performed with the help of data of the Description Logic International Workshop DL'98 in comparison with such popular systems of logical analysis of ontologies as JFact and HermiT.

KEY WORDS. Isomorphism, description logic, concept, tableau algorithm.

Introduction

One of the most popular methods to describe knowledge bases is the method of ontologies [1]. Ontologies are described in OWL (Web Ontology Language), the language recommended by World Wide Web Consortium [2] and the one that has become extremely popular due to having formal description logics (DL) as its fundamental principle [3]. Description logic is a decidable fragment of first-order logic used to describe knowledge by classes of objects and binary relations of a set of objects known as roles. The major advantage of description logics in comparison with other formal descriptive methods is their formal semantics, completely defined and assigned by interpretation. Formally defined semantics has been used by researchers to develop an algorithm [4] capable of knowledge base logical analysis which is a test on concept satisfiability, concept classification and the knowledge base consistency.

Subject of studies

For logical analysis of knowledge base there exist the tableau algorithm [4] and the hypertableau method 5] that can test the satisfiability of a complex concept in accord with the set of axioms. The major problem of the logical analysis of knowledge base is the computational complexity of such algorithms [6], which means that the work time of the algorithm may be too long in case when very large knowledge bases have to be analyzed. Therefore, a great number of optimizations to reduce this time have been developed. Most well-known of them are caching [7], global caching [8] and back jumping [9]. However, even when using these latest optimization techniques simultaneously, large knowledge bases cannot be tested. Thus, such knowledge bases as GALEN-module 1 and GALEN-full had not been classified by the tableau algorithm within a 20-minutes period [10].

In this article we attempt to describe another optimization technique for the tableau algorithm which is aimed at reducing the period needed for the knowledge base logical analysis.

Search for automorphisms of concepts. A complex concept is a concept that can have the form: $C \sqcup D$, $C \sqcap D$, $\exists R.C$, $\forall R.C$, A, where concepts C and D are also complex concepts, but concept A is a literal, i.e. a notation for representing a concept.

Let us denote the interpretation of concept C by I as $I \in M_C$, where M_C is a variety of interpretations that can be made up by the tableau algorithm, C_N is a set of all literals (atomic concepts) used in concept C.

Let us also introduce the syntactic equivalence relation $\cong$ for the whole set of concepts, and this relation will be true only for the concepts with the equivalent syntactic notation considering that the operations of disjunction ($\sqcup$) and conjunction ($\sqcap$) are commutative. For short, we will use the infix notation $C \cong D$. Let $C_{\{A//B\}}$ denote a concept in which literal A's name is changed to literal B's name. By automorphism of a concept we will denote bijective representation f: $C_N \longrightarrow C_N$ of a set of literals C_N of concept C into itself such that $C \cong C_{\{CNi//f(CNi)\}}$ for any C_{Ni} denoting the literal used in the notation of concept C. Let the substitute $\{C_{Ni} // f(C_{Ni})\}$ in the interpretation of I denote the substitution of all literals C_{Ni} by $f(C_{Ni})$ in the set L(x) for any individual x of the interpretation. The substitution of literals in the interpretation in accord with automorphism f will be denoted by I^f.

Lemma 1. If for the set of literals C_N of concept C there exists the automorphism f and I is the interpretation of this concept ($I \in M_C$), then the automorphic interpretation I^f belongs to the set M_C.

Proving.

Let M_C^f be the set of all interpretations of concept $C_{\{CNi//f(CNi)\}}$, and let us prove that the sets M_C and M_C^f are equal. Any interpretation that can be made up with the help of the tableau algorithm has such individual that is used to start making a graph-model. For concept C let this individual be denoted by the symbol x, for concept $C_{\{CNi//f(CNi)\}}$ let it be denoted by the symbol y. Suppose that there exists the interpretation $I \in M_C$ and $I \notin M_C^f$. As soon as making interpretations begins with L(x) = {C}, and L(y) = $\{C_{\{CNi//f(CNi)\}}\}$, then such interpretation can only exist if $C \not\cong C_{\{CNi//f(CNi)\}}$, and this contradicts the definition of automorphism of concepts. By analogy we can prove non-existence of the interpretation of I such that $I \notin M_C$ and $I \in M_C^f$.

Lemma 2. If for the set of literals C_N of concept C there exists the automorphism f and I is the interpretation of this concept, then concept C is satisfiable under the interpretation if and only if it is satisfiable under the interpretation I.

Proving.

Let us observe the first assertion. If C is satisfiable under the interpretation I, then C is satisfiable under the interpretation I^f. Concept C is satisfiable under the interpretation I^f, if and only if for any individual $x \in \Delta^{I^f}$: $\{A, \neg A\} \not\subseteq L(x)$ – according to the definition of the concept satisfiability—however, since concept C is satisfiable under the interpretation I, then $\{A, \neg A\} \not\subseteq L(y)$ for any $y \in \Delta^I$ and some $A \in C_N$. Suppose that $\{B, \neg A\} \subseteq L(x)$ given $f(B) = A$, but since automorphism is a bijective representation, then $f(A) \neq A$, therefore $\{A, \neg A\} \not\subseteq L(x)$. The assertion that if C is satisfiable under the interpretation I^f, then C is satisfiable under the interpretation I, is proved by analogy, given that f^{-1} is equal to the interpretation without automorphism.

We have modified the tableau algorithm, i.e. when using Lemma 1 we exclude the interpretations of concepts, or subconcepts, the automorphic concepts of which have been analyzed before. As it follows from Lemma 2, the concepts will be satisfiable or non-satisfiable and there will be no need to analyze the whole interpretation.

Let us now estimate the efficiency of the developed optimization. If we analyze a concept $(A \sqcup C) \sqcap (B \sqcup D)$ using the tableau algorithm, we will have to analyze the following interpretations:

$L(x) = \{A, B\}$;
$L(x) = \{A, D\}$;
$L(x) = \{C, B\}$;
$L(x) = \{C, D\}$.

and, if there exists the automorphism $f(C) = A$, $f(A) = C$, then according to Lemma 2 there is no need to analyze interpretation 3 (or 4) as it is isomorphic to interpretation 1 (in compliance with interpretation 2).

Let us estimate the number of the interpretations to analyze. If there exists the automorphism:

$C_{a1} \rightarrow C_{a2}$
$C_{a2} \rightarrow C_{a1}$
...
$C_{a2n-1} \rightarrow C_{a2n}$
$C_{a2n} \rightarrow C_{a2n-1}$

then when using the tableau algorithm we must make the non-determined choice in all disjunctions like $(C_{ai} \sqcup C_{aj})$, the maximum number of such disjunctions is equal to $n * (n - 1)$, and the number of the disjunctions left is equal to $m * (m - 1) + m * n$, m is the number of the disjunctions in which there are no concepts C_{ai}. The interpretations $2^{n*(n-1)}$ make it possible not to analyze one interpretation, i.e. the number of the interpretations to analyze is equal to $2^{(n * (n - 1) - 1) * (m * (m - 1) + m * n)}$. Thus, a single automorphism can reduce the number of the analyzed interpretations by quantity equal to $2^{(m * (m - 1) + m * n)}$, and the smaller n is, the greater m is ($m = K - n$, where K is the total number of disjunctions).

Search for isomorphic structures of concepts. To represent concepts a graph-like structure is used [11]. In this case the concept is represented as a directed acyclic graph with just one distinguished vertex with the in-degree equal to 0 and from which all the other vertices can be reached, their in-degree is > 0. To find concepts with isomorphic structures we must find out whether the graphs representing the given concepts are isomorphic. To state the isormorpic nature of graphs Edmundson's algorithm is employed [12], however, it can be employed only when analyzing tree graphs. To be able to use this method in search for automorphisms in a directed acyclic graph we will have to modify it. When the label is formed at the stage of going down, it is made up by all vertices from which the graph edges go to this vertex. After using Edmundson's algorithm to go through a graph we get classes of automorphisms of concepts which will then be used to test the satisfiability of the concepts.

Estimation of the method's efficiency

The presented optimization was realized in the system of logical analysis of ontologies TReasoner (the source code as well as the compiled library of the classes are available at https://code.google.com/p/treasoner/). To test the method we used the testing data set from the Description Logic International Workshop DL'98 [13]. To estimate the efficiency of the developed method we tested the TReasoner system along with and in comparison with most modern and popular systems of logical analysis of ontologies such as HermiT, version 1.3.7, [10] and JFact, version 1.0.0, that is ported into Java language by the system FaCT++ [14]. HermiT uses the hypertableau method while JFact and TReasoner use the tableau algorithm. As both HermiT and JFact work with ontologies in the owl format, the testing files were translated into the owl language. All systems were tested on ASUS Notebook VX7SX Series Intel Core i7-2630QM CPU@2.00 GHz 2.00 GHz; 6.00 GB RAM under the operation system Windows 7. The testing results are presented in the table.

Table

File name	HermiT	JFact	TReasoner
k_branch_n.owl	0	13	21
k_branch_p.owl	10	13	21
k_d4_n.owl	0	15	21
k_d4_p.owl	14	21	21
k_dum_n.owl	0	21	21
k_dum_p.owl	0	21	21
k_grz_n.owl	0	21	21
k_grz_p.owl	21	21	21
k_lin_n.owl	0	21	21
k_lin_p.owl	21	21	21
k_path_n.owl	1	21	21
k_path_p.owl	2	21	21
k_ph_n.owl	5	21	14
k_ph_p.owl	4	7	9
k_poly_n.owl	14	21	21
k_poly_p.owl	21	21	21
k_t4p_n.owl	0	21	21
k_t4p_p.owl	0	21	21

The first column of the table gives the name of the testing. Each file comprises 21 concepts, columns 2–4 of the table give the number of the performed tests that indicates the number of concepts which were tested by the system within the time limit of 7 seconds. File k_ph_n.owl contains only satisfiable concepts; besides, among the whole set of interpretations only one of them is satisfiable. The developed system tested a fewer number of interpretations due to the satisfiable interpretation having been found after going through a number of non-satisfiable interpretations. The modification presented in this article did not affect the results of testing the concepts as the concepts in the file do not contain automorphisms.

Conclusion. The method of reduction of interpretation space through discovering isomorphic interpretations has been developed. The correctness of the method has been proved and the number of interpretations excluded from the analysis has been estimated. The test results of the method show that it can be employed to significantly reduce the time limit needed for the satisfiability testing of concepts.

REFERENCES

1. Ivashko, A.G., Ivanova, E.I., Ovsjannikova, E.O., Kolomiec, S.I. Application of Description Logic for the Description of Information System Architecture. *Vestnik Tjumenskogo gosudarstvennogo universiteta — Tyumen State University Herald.* № 4. 2012. Pp. 137-142. (in Russian).

2. Jie Bao, Elisa F. Kendall, Deborah L. McGuinness, Peter F. Patel-Schneider. OWL 2 Web Ontology Language Quick Reference Guide (Second Edition) // http://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/

3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. The Description Logic Handbook: Theory, Implementation and Applications. CUP, 2003, p. 601.

4. Schmidt-Schaus, M. andSmolka, G. Attributive concept descriptions with complements. *Artificial Intelligence.* 48(1):1-26, 1991.

5. Motik, B., Shearer, R., Horrocks, I. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research* 36(1):165-228, 2009.

6. Donini, F.M., Massacci, F. EXPTIME tableaux for ALC. *Artificial Intelligence.* 124(1): 87-138, 2000.

7. Ding, Y. and Haarslev, V. Tableau caching for description logics with inverse and transitive roles. In Proc. DL-2006: International Workshop on Description Logics. 2006. Pp. 143-149.

8. Linh Anh Nguyen. An Efficient Tableau Prover using Global Caching for the Description Logic ALC. *Artificial Intelligenc.* 93(1):273-288, 2009.

9. Prosser, P. Hybrid Algorithms for the Constraint Satisfaction Problem. *Computational Intelligence.* 9(3). 1993. Pp. 268-299.

10. Horrocks, I., Boris Motik, and Zhe Wang. The HermiT OWL Reasoner // OWL Reasoner Evaluation Workshop. 2012.

11. Grigor'ev, A.V. Algorithm of ALC concepts simplification. Sovremennye problemy matematicheskogo i informacionnogo modelirovanija. *Perspektivy razrabotki i vnedrenija innovacionnyh IT-reshenij — Topical Problems of Mathematics of Information Modeling. Prospects of Innovative IT-Methods Development and Implementation. 2012.* (in Russian).

12. Busacker, G., Saaty, T.L. Finite Graphs and Networks. McGraw Hill, 1965.

13. Franconi, E. DL'98 Systems Comparison: official test data. //http://dl.kr.org/dl98/comparison/data.html

14. Tsarkov, D. and Horrocks, I. FaCT++ Description Logic Reasoner: System Description. In Proc. IJCAR 2006. Pp. 292-297.