Journal of Southeast University (English Edition) Vol. 22, No. 3, pp. 319 – 323 Sept. 2006 ISSN 1003—7985

New method for query answering in semantic web

Lin Peiguang[1,2] Liu Hong[3] Fan Xiaozhong[1] Wang Tao[1]

([1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)
([2]School of Computer and Information Engineering, Shandong University of Finance, Jinan 250014, China)
([3]College of Information Science and Engineering, Shandong Normal University, Jinan 250014, China)

Abstract: To promote the efficiency of knowledge base retrieval based on description logic, the concept of assertional graph (AG), which is directed labeled graph, is defined and a new AG-based retrieval method is put forward. This method converts the knowledge base and query clause into knowledge AG and query AG by making use of the given rules and then makes use of graph traversal to carry out knowledge base retrieval. The experiment indicates that the efficiency of this method exceeds, respectively, the popular RACER and KAON2 system by 0.4% and 3.3%. This method can obviously promote the efficiency of knowledge base retrieval.
Key words: description logic; assertional graph; semantic web; information retrieval

Current description logic (DL) systems offer a powerful inference mechanism, e. g., computing the sub-concepts of a given concept, but they usually offer a weak query language[1]. Current DL-reasoners generally support the following queries to access the assertional knowledge of a knowledge base:

Retrieval: Retrieve the instances of a query concept;

Realization: Determine the most specific concept an individual is an instance of;

Instantiation: Boolean query asking if an individual (a pair of individuals) is an instance of a given concept (role).

ABox-inference tasks require both a TBox and an ABox. In this paper we deal with the instance retrieval problem. ABox-reasoning is usually based on the tableau-algorithm. To infer that an individual i is an instance of C, an indirect assumption $\neg C(i)$ is added to the ABox, and the tableau-algorithm is applied. If this reports inconsistency, i is proved to be an instance of C. The main drawback of this approach, e. g., RACER, is that current DL-reasoners require processing the whole ABox[2]. Performance is also a main issue: when dealing with a large number of instances, the traditional approach is slow and inefficient, even when TBox and ABox optimization techniques are applied[3].

In this paper we present a new retrieval method for the ABox of knowledge base. This method converts the knowledge base (KB) and queries into the related assertional graph (AG) and then carries out the retrieval operation based on the graph traversal algorithm.

1 Related Research

Recent results show that DL systems can be enriched by a conjunctive query language, providing a solution to one of the weaknesses of traditional DL systems[4]. These results can be transferred to the semantic web community, where the need for expressive query languages has been witnessed by different proposals. Glimm et al.[1] presented a query answering system, and the system following the DQL and supporting acyclic conjunctive queries used a DL-reasoner to answer the queries, in which the conjunctive queries were transformed into DL retrieval or Boolean retrieval. The OWL-server was a similar system provided by the Knowledge System Laboratory (KSL) of Stanford University. Haarslev et al.[5] introduced a DL query language for retrieving ABox individuals that satisfied specific conditions. The presented language was substantially more expressive than traditional concept-based retrieval languages offered by previous DL reasoning systems and the Racer system was implemented. Calvanese et al.[6] presented a novel approach for determining the instances of DL concepts when huge amounts of underlying data were expected. It transformed a concept description into a Prolog program which represented a query-plan, which was done without any knowledge of the particular data. This method performed very well compared with techniques provided by the traditional DL theorem, while it stored the as-

Received 2006-04-25.
Foundation items: The National Natural Science Foundation of China (No. 69975010, 60374054), the National Research Foundation for the Doctoral Program of Higher Education of China (No. 20050007023).
Biographies: Lin Peiguang (1978—), male, graduate; Fan Xiaozhong (corresponding author), male, professor, fxz@bit.edu.cn.

sertions in related databases. The evaluation shows that this method performs better than KAON2 and worse than Racer when the scale of KB is appropriate.

2 New Retrieval Method

This retrieval method can be implemented in the following steps: ① Transform the knowledge base into a normal knowledge base assertional graph (KAG); ② Transform the query clause into a normal query assertional graph (QAG); ③ Regard the transformed QAG as the pattern and retrieve the individuals by comparing with KAG. It is the first step that can be executed only once and new retrieval operations can directly use the KAG generated the first time.

2.1 Preparation for retrieval

The preparation work for the retrieval is mainly to transform the knowledge base and query clauses into normal KAGs and QAGs. The objectives of the transformation are not only to get the concept that each individual belongs to and the roles between any two individuals, but also to prepare for the next step of matching retrieval.

2.1.1 Preprocessing of ABox

In order to convert the knowledge base into KAG, we must obtain the concept each individual belongs to and the roles between any two individuals. In order to do this, we present the rules to process the quantifiers ∀ and ∃.

Rule 1 The ∃-rule

Conditions: ABox contains $\exists R.C(x)$ and $R(x,z)$ and there is no $C(z)$ in ABox.

Actions: $\text{ABox}' = \text{ABox} \cup \{C(z)\} - \{\exists R.C(x)\}$.

Rule 2 The ∀-rule

Conditions: ABox Contains $\forall R.C(x)$ and $R(x, y)$, and there is no $C(y)$ in ABox.

Actions: $\text{ABox}' = \text{ABox} \cup \{C(y)\} - \{\forall R.C(x)\}$.

Instruction: If there is no $R(x, y)$, then delete $\forall R.C(x)$ directly.

After being processed by using the rules, ABox contains only the syntaxes ¬ , ∪, and ∩. Then using de Morgan's rules and the usual rules for the above syntaxes, we can convert all the concepts of the assertions including the same individual into conjunctive normal form in linear time[6].

Example 1 Suppose that ABox = {Man(Tom), (∃hasChild. Person)(Tom), hasChild(Tom, Charles), Woman(Charles), Man(John), hasChild(Rose, John), (Woman ∩ (∃hasChild. Person))(Rose), hasChild(Rose, Jerry)}. After applying the above rules and the usual rules to ABox, we can get ABox′ = {Man(Tom), hasChild(Tom, Charles), (Person ∩ Woman)(Charles), Woman(Rose), hasChild(Rose, John), (Person ∩ Man)(John), Woman(Rose), hasChild(Rose, Jerry), Person(Jery)}. It is obvious that the transformed ABox does not contain ∀ and ∃. We can also discover each individual's concepts and roles.

2.1.2 Preprocessing of query clause

We can regard the query clause as an assertion and adopt the above method to deal with it. The only difference between the query clause and the assertion is that the former may contain not only individuals but also variables. If it contains variables, we regard them as individuals.

2.1.3 Assertional graph

Definition 1 Atomic assertion is an assertion which contains only atomic concepts or atomic roles.

Definition 2 An assertional graph is a labeled directed graph $G(\alpha) = \langle V, E\rangle$ where V is a set of vertices and E is a set of arcs. The set V consists of all the individuals or variables existing in assertion α. The set E consists of all pairs $\langle v_1, v_2\rangle$, such that $v_1, v_2 \in V$ and $R(v_1, v_2)$ is a role in α. In AG, each vertex v is labeled by the concept of v, and each arc e which links individuals v_1 and v_2 is labeled with a role name R where $R(v_1, v_2)$ is a role in α. If a vertex has no label, then it is labeled with a top concept.

2.1.4 Query assertional graph

After the preprocessing of the query clause, each individual or variable is related to a concept which is expressed in conjunctive normal form and all concept assertions and role assertions are isolated. The preprocessed query clause can be expressed by the following equation:

$$\text{Query}(\boldsymbol{x}) = \text{Query}'(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \text{disj}(\text{conj}(\boldsymbol{w})) = \cup\Big(\bigcap_i((\bigcap_m C_m)(w_i))\bigcap_n R_n(w_i, w_j)\Big) \quad (1)$$

where vector $\boldsymbol{x}$ consists of the must-bind variables, vector $\boldsymbol{y}$ consists of the rest ones vector, $\boldsymbol{z}$ consists of the individuals, and vector $\boldsymbol{w}$ is the union of vectors $\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}$; w_i and w_j denote some individual or variable; disj(*) denotes the disjunctive normal form of all the individuals and variables; while conj(*) denotes the conjunctive normal form; n is equal to the number of elements in vector $\boldsymbol{w}$. C_m is an atomic concept and an R_n is an atomic role.

In Eq. (1), $(\bigcup_m C_m)(w_i)$ denotes the concept of w_i, and $\bigcap_n R_n(w_i, w_j)$ denotes the roles related to w_i. Then QAG can be generated by algorithm 1.

Algorithm 1 Algorithm of QAG generation

Input: The preprocessed query clause.

Procedure:

For each conj($\boldsymbol{w}$) indisj(conj($\boldsymbol{w}$))

Generate the Graph $G = \langle V, E \rangle$ where V consists of the individuals and variables and $E = \Phi$.

For each w_i in conj($\boldsymbol{w}$)

Label w_i with its concept expression

Get its related roles, create the arc fromwitowjand label the arc with role name R, $R(w_i, w_j)$ in Query($\boldsymbol{x}$).

End for

End for

When executing this algorithm, sometimes more than one QAG are generated because we cannot get the disjunctive form of the roles. But we draw the following conclusion that these QAGs are very similar and the only differences are in the roles of each individual. Optimization is pursuant to this.

The main work of this algorithm is to traverse all individuals in ABox, so this algorithm can be completed in linear time.

2.1.5 Knowledge base assertional graph

After the preprocessing of the knowledge base, in the same way as the preprocessed query clause, the KAG can be generated by algorithm 2. It is obvious that this algorithm can be completed in linear time.

Algorithm 2 Algorithm of KAG generation

Input: The preprocessed knowledge base.

Procedure:

Generate the Graph $G = \langle V, E \rangle$, where V consists of the individuals and variables and $E = \Phi$.

For each v in V

Label v with its concept expression

Get its related roles, create the edge from v to w, then label the edge with role name R.

End for

2.2 Retrieving by matching graphs

After transforming the knowledge base and the query clause into assertional graphs, we regard the QAG as a pattern and find the matching sub-graph from the KAG. Then the retrieval problem is converted into the matching one based on the AG. Following the transformation method defined in 2.1.1 and 2.1.2, the final concepts in the KAG and the QAG are conjunctive normal forms and the roles are atomic ones. To implement the matching operations between the QAG and the KAG, in addition to matching the two graphs, we also need to judge if the related concepts and roles are matching or not.

Suppose that the concept in KAG is C_K and C_Q in the QAG, then they can be expressed by the following two equations:

$$C_K = \bigcap_m C_m \tag{2}$$

where C_m is the conjunctive normal form.

$$C_Q = \bigcap_n C_n \tag{3}$$

where C_n is the conjunctive normal form.

To prove the two concepts are matching, we only prove Eq. (4) correctly for any individual $a \in C_K$. Apropos of roles, we only judge if the two roles are the same or not.

$$C_K(a) \Rightarrow C_Q(a) \tag{4}$$

where a is an individual name.

The procedure of concept matching can be completed in linear time by the related research of first order logic and the retrieval operation can be completed in PTIME.

2.2.1 Boolean queries

This is a simple procedure. The only thing we need to do is to enumerate each individual and judge its concept and roles.

2.2.2 Queries with variables

This procedure can be implemented by virtue of 2.2.1 as we can substitute the variables with the individuals in the knowledge base. But this method needs to enumerate all individuals. In order to decrease the time complexity, we can make use of the information of concepts and roles and the arcs in graphs. The individuals can be filtered by concepts and roles (or arcs in graph) and then we can reduce the times of matching.

It is obvious that the above two procedures can be completed in PTIME because ① They spend linear time in traversing the graph; ② They spend linear time in matching the concept; and ③ They spend linear time in filtering the individuals.

2.3 Optimization of query

In order to improve the efficiency of query, we optimize the query when it generates more than one QAG. As introduced in section 2.1.4, the differences in these QAGs lie in the roles of each individual or variable. Then in all these QAGs, the concept part of Eq. (1), $\bigcap_i (\bigcap_m C_m)(w_i)$ is the same and the role part $\bigcap_n R_n(w_i, w_j)$ is different. Eq. (5) can be deduced from Eq. (1) by the equation $(A \cap B) \cup (A \cap C) = A \cap (B \cup C)$.

$$\text{Query}(\boldsymbol{x}) = \text{Query}'(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \text{conj}(\boldsymbol{w}) = \bigcap_i \Big(((\bigcap_m C_m)(w_i)) \cap \Big(\cup \Big(\bigcap_n R_n(w_i, w_j) \Big) \Big) \Big) \tag{5}$$

Compared with Eq. (1), Eq. (5) has the same concept part and its role part is changed into the disjunction of related role assertions in all QAGs. Because the operation of matching roles can be completed in constant time, this optimization reduces the times of matching and the time complexity can at least decrease by one order of magnitude.

3 Experiments and Analysis

We have evaluated our solution by implementing a system named IRS-DL (P4 1.8G, 512M RAM, Windows 2000 Pro). In view of the fact that there is not the standard platform for the knowledge base retrieval, we manually construct a knowledge base which is used to describe the persons in the same family and their things. This KB includes 500 concept assertions and 1 000 role assertions and we divide the KB into three groups for the evaluation. Their numbers of concept assertions and role assertions are (300, 500), (400, 700) and (500, 1 000). Then we define eight query clauses for the experiments and the numbers of variables and constants in every clause are (0, 3), (0, 5), (1, 2), (3, 2), (3, 3), (3, 0), (4, 0) and (5, 0). The test were run under RACER 1.7.16, KAON2 release 2005 - 12 - 08. The results are shown in Tab. 1.

Tab. 1 Comparison of runtime performance ms

Number	The 1st group			The 2nd group			The 3rd group		
	R	*K*	*I*	*R*	*K*	*I*	*R*	*K*	*I*
1	74	78	80	75	85	86	85	90	92
2	60	73	70	72	76	78	80	86	90
3	165	179	180	164	182	186	178	190	192
4	164	179	175	168	185	184	180	194	195
5	168	183	180	172	192	186	182	200	198
6	210	234	231	220	234	239	223	238	242
7	221	239	236	224	242	240	229	251	248
8	220	241	235	228	248	242	234	256	251

In Tab. 1, data in the *R* (RACER) column and the *K* (KAON2) column are averaged and data in the *I* (IRS-AG) column are obtained for the first time after the system runs. The results show that the runtime performance of this system is a little worse than RACER and similar to KAON2. The experiments also indicate that the average runtime after running one more time can decrease 85% or so. Then as in this case, this system performs a bit better than RACER. Fig. 1 shows the comparison of the average runtime when they execute the same retrieval on the same KB ten times. The results prove that the system implemented on our algorithm can carry out faster and more efficient retrieval.

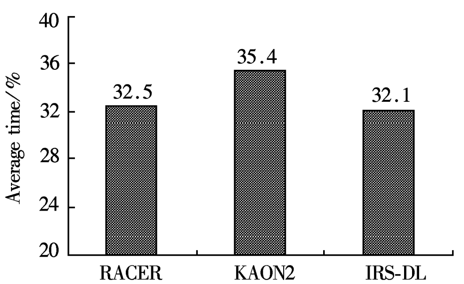

Fig. 1 Comparison of average runtime performance

4 Conclusion

The search engine is the most important method used for people to extract information, and the semantic web is the next generation of the Internet, so query answering in the semantic web is a very important aspect of related researches on the semantic web. This paper presents the concept of the assertional graph and realizes the methods of transforming the knowledge base and query clause into KAG and QAG. The new knowledge base retrieval method is also implemented by extending the graph matching algorithm and the graph traversal algorithm. Neither does this method limit the query clause, nor does it add extra requirements to ABox. The algorithmic analysis and the experiment indicate this method obviously enhances the efficiency of the KB retrieval.

References

[1] Glimm Birte, Horrocks Ian. Query answering systems in the semantic web[A]. In: Bechhofer Sean, Haarslev Volker, Lutz Carsten, eds. *Proc of the KI-*04 *Workshop on Applications of Description Logics* [C]. Edinburgh: CEUR-WS, 2004. 21 - 30.

[2] Nagy Zsolt, Lukacsy Gergely, Szeredi Peter. Translating description logic queries to prolog[A]. In: van Hentenryck P, ed. *Proc of the* 8*th International Symposium on Practical Aspects of Declarative Languages* (*PADL* 2006) [C]. Berlin: Springer, 2006. 168 - 182.

[3] Haarslev Volker, Moller Ralf. Optimization techniques for retrieving resources described in OWL/RDF document: first results[A]. In: Whistler, ed. *Proc of the Ninth International Conference* (*KR*2004) [C]. British Columbia: AAAI Press, 2004. 163 - 173.

[4] Horrocks Ian, Tessaris Sergio. Querying the semantic web: a formal approach[A]. In: Ian Horrocks, Hendler James, eds. *Proc of the* 13*th International Semantic Web Conference* [C]. Berlin: Springer, 2002. 177 - 191.

[5] Haarslev Volker, Moller Ralf. Querying the semantic web with Racer + nRQL[A]. In: Bechhofer Sean, Haarslev Volker, Lutz Carsten, eds. *Proc of the KI-*04 *Workshop on Applications of Description Logics* [C]. Edinburgh: CEUR-WS, 2004. 31 - 40.

[6] Calvanese Diego, de Giacomo Giuseppe, Lembo Domenico, et al. Data complexity of query answering in description logics[A]. In: Horrocks Ian, Sattler Ulrike, Wolter Frank, eds. *Proc of the* 2005 *Description Logic Workshop*[C]. Edinburgh: CEUR-WS, 2005. 49 - 60.

一种新的语义网信息检索方法

林培光[1,2] 刘 弘[3] 樊孝忠[1] 王 涛[1]

([1] 北京理工大学计算机科学与技术学院,北京 100081)
([2] 山东财政学院计算机信息工程学院,济南 250014)
([3] 山东师范大学信息科学与工程学院,济南 250014)

摘要:为了提高基于描述逻辑的知识库检索的效率,提出了断言图的概念和基于断言图匹配的检索方法.断言图是带有标签的有向图,基于断言图匹配的检索方法首先利用给定规则对基于描述逻辑的知识库和查询请求中的量化符号进行处理,然后转换成有向的知识断言图和查询断言图,最后利用图的遍历算法实现基于描述逻辑知识库检索.实验表明,该方法能够明显提高知识库检索的效率,其执行效率比当前流行的 RACER 和 KAON2 分别提高了 0.4% 和 3.3%.

关键词:描述逻辑;断言图;语义网;信息检索

中图分类号:TP391.3