관계형 데이터베이스 기반 ABox Reasoning

Ankesh Khandelwal*, 오정정**, 김주리**, 이현창***, 한성국**
*인도 Delhi연구소, **원광대학교 컴퓨터공학부, ***원광대학교 정보전자상거래학부

ABox Reasoning with Relational Databases

Ankesh Khandelwal, Summit Bisai, Kim Ju-Ri, Lee Hyun-Chang, Han Sung-Kook
Indian Institute of Technology Delhi, Wonkwang University
E-mail :{cyanic, hclglory, skhan}@wonkwang.ac.kr

요 약

OWL 온톨로지의 확장 가능한(scalable) 추론(reasoning)에 대한 접근 방법으로 SQL로 구축된 논리 규칙을 관계형 데이터베이스에 저장되어있는 개체(individual)에 대한 사실(facts)과 공리(axioms)들에 적용하는 것이다. 예로서 미네르바(Minerva)는 서술 논리 프로그램(Description Logic Program, DLP)을 적용함으로써 ABox 추론을 수행한다. 본 연구에서는 관계형 데이터베이스를 기반으로 추론을 시도하며, 대규모 논리 규칙 집합을 사용한 추론을 시도한다. 뿐만 아니라, 특정 클래스에 속한 익명(anonymous)의 개체들과 개체들의 묵시적(implicit)인 관계성 추론을 시도하며, 필요한 경우 새로운 개체를 생성함으로써 명시화하여 추론을 시도한다. 더욱이, 추론의 논리 패러다임(paradigm)에서부터 데이터베이스 패러다임에 이르기까지 변화 시켜가면서 카디널리티(cardinality) 제약을 만족하는 개체들에 대한 제약적인 추정 추론을 시도하며, 벤치마크 테스트 결과 향상된 추론 능력을 얻을 수 있음을 보인다.

1. Introduction

The logical foundation of Web Ontology Language(OWL)[1] is Description Logic(DL)[2]. A DL knowledge base consists of two components, a TBox that describes terminology and an ABox that contains assertions about individuals. Correspondingly, the OWL-DL reasoning includes TBox reasoning (i.e., reasoning with concepts) and ABox reasoning(i.e., reasoning with individuals).

Racer [3] and Pellet [4] are complete and sound reasoners for OWL ontologies. However, for inferences about individuals they do not scale up to the amount of data that forms the semantic web. KAON2 [6] uses a datalog engine to make sound and almost complete reasoning (no support for nominals and certain DL axioms with cardinality restrictions). It draws quicker inferences about individuals and is scalable compared to the tableaux-based reasoners, Racer and Pellet.

Alternatively, relational databases have been used to achieve scalable reasoning. Relational databases offer efficient indexing techniques for data retrieval and they can be used for persistent storage of large amounts of data. In our work we use relational database to store OWL ontologies and perform ABox reasoning.

Rest of the paper is organized as follows. In section 2 describes the related work. In the 3rdsection we give the details of our approach. In the next section we discuss implementation and in the following section we report evaluation results. Section 7 concludes the paper.

2. Related Work

We can distinguish four approaches to reasoning of ontologies in OWL-DL or its subsets.

Tableaux based provers for DL implement conceptually sound and complete approach. Pellet[PELLET] and RacerPro[RACER] reasoners are excellent implementations based on this approach. Elaborate optimization and reduction methods, like [PSEUDOMODELS], have also been designed. However these systems are not scalable. Alternatively OWL ontology can be transformed to a disjunctive datalog program and reasoning can be performed using disjunctive datalog engine. KAON2 is a sound implementation based on this approach and it utilizes well-known optimization techniques for disjunctive databases such as magic set transformation. However it does not support nominals and has problems with axioms containing cardinality restrictions. Third approach is to use a standard rule entailment engine to reason with OWL. OWLIM and OWLJessKB are based on this approach. However this approach is known to be incomplete. Fourth approach is a hybrid approach of combining an external reasoner with a Database system. Instance Store [Instance Store], Large Abox Store (LAS) [LAS] and Minerva [Minerva] are hybrid approach based systems, and they combine reasoner and databases differently. Instance store has limited capabilities because it deals only with role free Aboxes. LAS is complete since it uses RACER to perform reasoning but is for same reason not scalable even when it enhances query response time by storing relevant information in databases.

Minerva combines a DL reasoner and a rule engine for ontology inference. It materializes all the inferred results into a database. It performs no reasoning at the query answering time and answers the queries by direct retrieval of instances from the database. It decomposes complex class descriptions into instantiations of class constructors, assigned to them new ids. It obtains concept and role taxonomies using DL reasoner and executes DLP rules in SQL for Abox reasoning. Description Logic Programs (DLP), intersection of Logic (DL) and Logic Programs (LP), is the LP-correspondent of Description Horn Logic (DHL), which is a subset of OWL DL.

The whole algorithm revolves around precise understanding of the RDF/XML document or the RDF graph. Each line in the RDF/XML document is parsed well by interpreting the meaning of rdf:about, rdf:id, rdf:Description, rdfs:Class and etc.

3. Approach

We would now consider some of additional axioms, $C \subseteq \neg D$, $C \equiv \neg D$, $x=y$, $T \subseteq \leq 1\ R$, $T \subseteq \leq 1\ R-$, $C \subseteq \exists R.\{a\}$, $C \subseteq \exists R.D$, $C \subseteq \beta 1R$, $C \subseteq \rho 1R$, $C \subseteq \rho n\ R$, $C \subseteq \beta n\ R$, $\rho nR \subseteq C$, $\beta n\ R \subseteq C$, compared to the axioms used for Abox inferencing in Minerva [Minerva].

Some of these axioms can be mapped either to logic rules, some with disjunction in the head, or to integrity constraints [KAON2]. In few cases we can create new individuals for better inferencing [Tableaux]. For the remaining axioms we suggest an approximate algorithm based on an intuitive deduction. All the logic rules except those with disjunction in the head can be implemented using SQL.

We have had distinctly two paradigms of OWL reasoning, the Tableaux based approach and the Logic based approach. Even as we implement logic rules using SQL we continue to be in the logic paradigm. As we extend the OWL axioms that are handled by Minerva through mapping to rules we first try to apply a technique, of creation and addition of new individuals and their facts, from Tableaux based algorithm [Tableaux]. It has been shown in [KAON2 defHornLogic] that this technique can be realized in logic paradigm through introduction of new predicates. Secondly, we take the distinct advantage of working in database domain. We use the aggregation function to count the results that satisfy our query and easy backtracking by deletion of entries in the tables.

It may not be possible to map this axiom to a single rule, and for cardinality values greater than 1 may require disjunction in the head. Let's consider the case when cardinality value is 2. It is mapped to the following rule.

sameAs(y, z), sameAs(z, w), sameAs(w, y) :- Related(x, R, y), Related(x, R, z), Related(x, R, w), TypeOf(x, C), SubClass(C, RX), MaxCardinality(RX, R, n).

As we do not have convenient way to map $C \subseteq \beta n\ R$ to logic rule we cannot handle the following axiom $C \subseteq =n\ R$ (C subsumes a DL cardinality restriction on property R).

For the next two axioms we use techniques to deduce approximate facts, i.e. the facts that are possibly but not necessarily true.

$\rho n\ R \subseteq C$ (a DL maximum cardinality restriction on property R subsumes C)

Say, we have MinCardinality(RN, R, n). We count the number of distinct R-successors for an individual, say x, and if it is more than n-1 we assume that TypeOf(x, RN), and therefore TypeOf(x, C). Here by distinct we mean individuals that have not been deduced to be the same. In addition to the assumption TypeOf(x, RN) we also assume that the distinct n-1 R-successors of x are pair-wise different from each other. We add these axioms along with the facts, TypeOf(x, RN) and TypeOf(x, C), to our database.

For the axiom $\beta n\ R \subseteq C$ (a DL maximum cardinality restriction on property R subsumes C), we make similar assumptions if an individual has less than n+1 distinct R-successors and the individual belongs to the domain of R (this is relevant when an individual x has no R-successors). However, in this case we do not assume the R-successors to be pair-wise different. We add the assumed facts to our database.

Now we deduce facts and axioms based on the assumptions. When we encounter an inconsistency, either because an individual belongs to two disjoint classes or two individuals are inferred to be both same and different from each other, we delete the original assumptions along with the deductions made based on them. In case we do not encounter an inconsistency we persist with the approximations and the inferred facts and axioms. When we do not find an inconsistency with our assumption, we have found a model [REFERENCE] of our knowledge base where our assumptions are correct. However, an individual x is an instance of concept C w.r.t knowledge base K iff for every model I of K x is an instance of concept C [REFERENCE]. Therefore our assumptions are only approximate.

Moreover our assumptions are made by counting of individuals, an action which we cannot map to a set of logic rules that can perform deductions similarly. We cannot claim to have deduced these facts directly from the knowledge base. The only other way to infer a fact $x \in C$ is by inferring $A \cap \{\neg C(x)\}$ to be inconsistent [HORROCKS, some]. However we do not perform this check.

4. Evaluation

Evaluation is conducted on University Ontology Benchmark (UOB) [UOB]. The UOB consists of university domain ontologies, customizable and repeatable synthetic data, a set of test queries and corresponding answers. OWL-DL ontology in UOB covers almost the complete set of OWL-DL constructors. For our experiments we consider the 3 test sets: DL-1, DL-5 and DL-10. The numbers indicate the number of universities. Each university contains about 20 departments and over 210,000 statements. DL-10, the largest data set, contains over 2,200,000 statements. There are 15 queries in the benchmark which cover most of the features of OWL-DL. The details of all the queries can be found in [UOB].

Figure 1 gives load times of our systems and that of Minerva (Load times have been interpreted from graphs in [UOB Paper]). Our system takes comparable time to load ontologies, even though it performs expensive extra reasoning tasks like creating new individuals and adding their facts and deducing approximate cardinality-restriction memberships.

	Our System (Time in sec)	Minerva (Time in sec)
DL-1	679	800
DL-5	6896	6250

Fig.1. Load Times (time in secs)

5. Conclude

This paper presented a scalable reasoner for large scale OWL ontologies based on an approach discussed in [MINERVA]. We extended its capability to perform Abox reasoning, and suggested intuitive approximate technique to handle certain axioms containing cardinality restrictions. We implemented and tested our algorithms, and enhanced Abox inferencing is evident from results of the University Ontology Benchmark tests. With approximate inferences we are able to make exact deductions for the query that requires deduction of facts from axioms containing minimum cardinality restriction. (Even with additional inferencing tasks that we perform our system takes comparable (same/ similar) time to Minerva to complete Abox reasoning).

For RDBMS to be seriously used for complex OWL-DL reasoning tasks we should be able to handle negations. Therefore we believe our next challenge is to achieve develop techniques to handle axioms with negation.

[참고문헌]

[1] Bechhofer, S., van Harmelen, Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., eds.:OWL Web Ontology Language Reference. W3C Recommendation.(2004)

[2] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.:The Description Logic Handbook:Theory, Implementation, and Applications., Cambridge University Press(2003)

[3] Haarslev, V., M”oller, R.:RACER System Description. In:Automated Reasoning, First International Joint Conference, IJCAR 2001.(2001)

[4] Sirin, E., Parsia, B.:Pellet: An OWL DL Reasoner. In:DL.(2004)

[5] Volker Haarslev and Ralf M”oller. Optimizing TBox and ABox reasoning with pseudo models. In Proceedings of the International Workshop in Description Logics 2000(DL2000)(2000)

[6]Motik, B., Studer, R.:KAON2-A Scalable Reasoning Tool for the Semantic Web. In: Proceedings of the 2nd European Semantic Web Conference (ESWC’05), Heraklin, Greece(2005)

[7] Boris Motik: Reasoning in Description Logics using Resolution and Deductive Databases. In: Dissertation, Forschungszentrum Informatik an der Univ. Karlsruhe(2006)