Conjunctive Query Answering in Distributed Ontology Systems for Ontologies with Large OWL ABoxes

Xueying Chen[1] and Michel Dumontier[1,2]

[1] School of Computer Science,
[2] Department of Biology,
Carleton University, 1125 Colonel By Drive, K1S 5B6, Ottawa, Canada
xychen@scs.carleton.ca, michel_dumontier@carleton.ca

Abstract. We present a query processing procedure for conjunctive queries in distributed ontology systems where a large ontology is divided into ontology fragments that are later distributed over a set of autonomous nodes. We focus on ontologies with large ABoxes. The query processing procedure determines and retrieves the facts that are relevant to answering a given query from other nodes, then construct a new fragment that includes the set of relevant facts, the local TBox and RBox. The given query is evaluated against the new fragment and answers are returned to the user. We prove that our technique returns sound answers for queries over OWL ontologies.

1 Introduction

In recent years, we have witnessed both the number of ontologies appearing on the web growing rapidly and the size of ontologies expanding dramatically. While Ontology integration systems(OIS) [2] provide tools to semantically combine different ontologies, existing reasoners show poor performance in reasoning with large ontologies. An example of ontology with large ABox is yOWL[3] which contains estimated $582,800$ instances. [3] shows Racer[4] is unable to execute instance realization(which classifies a given instance into their most specific concept) in yOWL using an Intel Pentium 4 computer with 3GB RAM.

In our earlier work [1], we define a Distributed Ontology System (DOS) which is motivated to solve query processing issues resulting from large ontologies. This paper, as a continuation of [1], proposes a query processing procedure for conjunctive queries posed over ontologies with large ABoxes in DOS. The main idea is that given a query, the master node(the node that receives the given query) determines the set of facts that are relevant to answering the query by unfolding the query with its subconcepts and subroles. Then the master node collects the relevant facts from other nodes and constructs a new ABox which

[3] http://ontology.dumontierlab.com/yowl-jbi.owl
[4] http://www.racer-systems.com/products/index.phtml

along with the local TBox and RBox form a new ontology fragment. Finally, the given query is evaluated against the new fragment and the answers are returned to the user. We prove that the procedure can return sound answers for OWL ontologies.

This paper is organized as follows. Section 2 briefly introduces DOS. The querying processing procedure is described in Section 3. Conclusions and future work are in Section 4.

2 Distributed Ontology Systems

In our previous work [1], we propose Distributed Ontology System(DOS) which allow both query processing and collaborative ontology development.

In a DOS, there are a set of autonomous node across the network and each of them is equipped with a DL reasoner. A large ontology is divided into smaller pieces(called ontology fragments) that are distributed over the nodes. These fragments are logically interrelated, and the relationship between them should be recorded so that it can be used to coordinate the nodes to complete tasks such as query processing, collaborative development of the ontology and so on.

Formally, *A distributed ontology system $\mathcal{I}$ is a triple $< \mathcal{O}, \mathcal{F}, \mathcal{D} >$ where*

1. *$\mathcal{O}$ is an ontology, expressed in a language $\mathcal{L}_{\mathcal{O}}$ over an signature $Sig(\mathcal{O})$ containing terms of $\mathcal{O}$.*
2. *$\mathcal{F}$ is a set of ontology fragments $F_1, \cdots, F_{\mathrm{n}}$ into which $\mathcal{O}$ is divided. We denote with $Sig(F_i)$ the signature containing terms of ontology fragment F_{i}. Here, $Sig(F_i) \subseteq Sig(\mathcal{O})$, and $Sig(\mathcal{O}) = \bigcup_{\mathrm{i}=1,\cdots,\mathrm{n}} Sig(F_i)$.*
3. *$\mathcal{D}$ is a dictionary which contains the semantic information of each fragment and the relationship between fragments.*

There exists a special case of ontology fragments: modular fragment where for a query q, the answers to q can be returned by evaluating q against ontology fragments instead of the whole ontology. Formally, a *modular* fragment is defined as follows.

Let $\mathcal{O}$ be an ontology and $Sig(\mathcal{O})$ its signature, F a fragment of $\mathcal{O}$, and q a query posed over $\mathcal{O}$. We say F is modular with respect to a signature $Sig(K)$ if it holds that

$$\mathcal{O} \models q^{\mathcal{O}} \Longleftrightarrow F \models q^{\mathcal{O}}$$

for $Sig(q) \subseteq Sig(K)$ where $Sig(q)$ is the signature of q.

3 A Query Processing Procedure for Conjunctive Queries

In this section, we describe a query processing procedure for conjunctive queries in DOS. We assume the case of non-modular fragments and focus on ontologies with large ABoxes, therefore we distribute only the ABoxes, i.e., each ontology fragment in DOS contains a complete TBox and RBox, but an incomplete ABox of the original ontology.

We say that a conjunctive query is of the following form:

$$A(\bar{x}) \leftarrow B_1(\bar{y_1}), \cdots, B_n(\bar{y_n})$$

where $A(\bar{x})$ and $B_i(\bar{y_i})(i = 1, \cdots, n)$ are terms in a DL language, A and B_i are DL predicates, $\bar{x}$ and $\bar{y_i}$ are lists of variables and constants that match the arity, i.e., number of attributes of x, y_i respectively.

The idea of our approach is to extract a relevant ABox for a given query. Consider a query q posed over an ontology $\mathcal{O}$, the master node(the node that receives the q) runs the querying processing procedure. The procedure first constructs a new query q' by appending subconcepts(subroles) to the non-atomic concepts(roles) in q with disjunction. q and q' are logically equivalent with respect to $\mathcal{O}'s$ TBox and RBox. In addition to q', the procedure also generates a set of instance or role retrieval queries Q that is used to retrieve the facts which are relevant to answering q(or q'). Then the master node forwards Q to other nodes. All the nodes, including the master node, runs the queries in Q and returns the answers to the master node. Upon receiving the relevant facts, the master node constructs a new ontology fragment(the set of relevant facts +TBox+RBox). Finally, the master node evaluates q' against the new fragment and returns the answers to the user.

We divide the query processing procedure into three steps:

1. Normalize TBox and RBox.
   Let C, D be concepts and R, P roles.
   (a) Replace equivalence with subsumption, that is, replace each axiom of the form $C \equiv D(r \equiv p)$ with $C \sqsubseteq D(r \sqsubseteq p)$ and $D \sqsubseteq C(p \sqsubseteq r)$.
   (b) Transform each general inclusion axiom into its normal form, that is, replace each axiom of the form $C \sqsubseteq D(r \sqsubseteq p)$ with $\top \sqsubseteq \neg C \sqcup D(\top \sqsubseteq \neg r \sqcup p)$.
   (c) Transform the right hand side of each general inclusion axiom into disjunctive normal form.
2. Unfold q using axioms in the normalized TBox and RBox to get q' and a set of queries Q that retrieves relevant facts .
   q' is constructed by doing the following recursively. For each non-atomic concept C or role r in q, search the normalized TBox(RBox). If C or r appears as a conjunctive clause on the right hand side of the inclusion axioms, take the rest of the formula on the right hand side, negate it(we get the subconcept/subrole of C/r), then append it to C or r with disjunction. Note that q' and q are logically equivalent. The recursion ends when all the concepts or roles in the appended clause are atomic. Q contains all the concepts and roles in q' and is used to retrieve facts.
3. Forward Q to other nodes. Each node runs the queries in Q and sends the results to the master node. The master node collects all the instances and combine them with its TBox and RBox to form a new ontology fragment F'. Finally, the master node evaluates q' against F' and returns the answers to the user.

Proposition 1. *(Soundness) Let $\mathcal{O}$ be an OWL ontology in DOS, q a conjunctive query posed over $\mathcal{O}$, q' a conjunctive query after applying rules in the procedure to q, F' an ontology fragment constructed according the rules in the procedure. Then:*

$$F' \models \bar{a} \text{ is an answer to } q' \Longrightarrow \mathcal{O} \models \bar{a} \text{ is an answer to } q$$

Proof.

$F' \models \bar{a}$ is an answer to q'
$\Rightarrow \mathcal{O} \models \bar{a}$ is an answer to q'(by monotonicity of DL reasoning and $F' \subseteq \mathcal{O}$)
$\Rightarrow \mathcal{O} \models \bar{a}$ is an answer to q(q and q' are logically equivalent)

Corollary 1. *F' is a modular fragment with respect to $Sig(q)$ where $Sig(q)$ is the signature of q.*

Proof. Since q and q' are logically equivalent, evaluate them against F' would return the same set of answers. From proposition 1, we get that evaluating q against F' returns the same answer set as evaluating q against $\mathcal{O}$.

We now discuss the computational complexity of the procedure step by step. Step 1 is a syntactic transformation of axioms in TBox and RBox, which can be done in polynomial time in the size of TBox and RBox. Step 2 is recursive with the purpose to unfold concepts and roles where the size of results can reach exponential in the size of the inputs, that is, step 2 in the worst case is in EXPTIME. Step 3 is basically standard OWL-DL reasoning, which is intractable in the worse case. Therefore, the lower bound of complexity of the procedure is EXPTIME.

4 Conclusions and Future Work

In this paper, we propose a query processing procedure for conjunctive queries over ontologies with large ABoxes in DOS. Our approach aims at reducing the size of large ABoxes by filtering out those facts that are not relevant to answering queries. We prove that our approach returns sound answers for OWL ontologies. Our future work is to identify the expressivity of OWL for which our technique returns complete query answers.

References

1. X. Chen and M. Dumontier. A framework for distributed ontology systems. In *proc. of the Second Canadian SemanticWebWorking Symposium(CSWWS09)*, page 137-148.
2. N. F. Noy. Semantic integration: a survey Of ontology-based approaches. *SIGMOD Record, Special Issue on Semantic Integration*, 33, 4. Published in 2004.
3. N. Villanueva-Rosales and M. Dumontier. YOWL: an Ontology-Driven Knowledge Base for Yeast Biologists. *Journal of Biomedical Informatics*. Available online May 11, 2008.