A New Operator for ABox Revision in DL-Lite

Sibei Gao[1] **Guilin Qi**[1] **Haofen Wang**[2]
[1] Key Laboratory of Computer Network and Information Integration of State Education Ministry
School of Computer Science and Engineering, Southeast University, China, gqi@seu.edu.cn
Postal Address:SiPaiLou #2, Nanjing, 210096, P.R. China
Phone: 86-25-52090910
[2] Shanghai Jiao Tong University, Shanghai, China

Abstract

In this paper, we propose a new operator for revising ABoxes in DL-Lite ontologies. We present a graph-based algorithm for ABox revision in DL-Lite, which implements the revision operator and we show it runs in polynomial time

Introduction

Revising ontologies in Description Logics (DLs) deals with the problem of incorporating a new ontology into an old one consistently. This problem is important as DLs underpin W3C standard Web ontology language OWL and ontologies may evolve during their construction. Existing revision operators in DLs are mostly generalizations of belief revision operators in propositional logic.

Recently, there has been some work on revising ontologies in DL-Lite family, which is a family of DLs that provides tractable reasoning services. In (Qi and Du 2009; Kharlamov and Zheleznyakov 2011), some model-based revision operators in DL-Lite are proposed. These operators suffer from the inexpressibility problem, i.e., the result of revision may not be expressed in DL-Lite anymore. In (Calvanese et al. 2010), the authors propose algorithms for TBox revision and ABox revision respectively. The algorithm for TBox revision suffers from the problem of non-determinism, i.e., the output of the algorithm may change if we run it several times. In contrast, the result of ABox revision is uniquely defined. The algorithms for ABox revision run in polynomial time. However, they need to compute the ABox closure w.r.t. the TBox, which will hinder their applicability for ontologies with large ABoxes, as it needs much space and time (even if done off-line), especially when the ABox is frequently changed. Here we restrict our attention to the problem of ABox revision, which captures many scenarios, such as database, ontology-based data management, etc.

In this paper, we first define a new operator for ABox revision in DL-Lite. We then present a graph-based algorithm for ABox revision in DL-Lite. Our algorithm has potential to scale to ontologies with large ABoxes as it leverages a graph structure and some efficient graph operators on top of it. Details of our work can be found in the technical report, which is available at http://gqi.limewebs.com/aaaist12.pdf.

Preliminaries

DL-Lite

In our work, we consider DL-Lite$_{FR}$, which is an important fragment of DL-Lite. We start with the introduction of DL-Lite$_{core}$, which is the core language for the DL-Lite family. The complex concepts and roles of DL-Lite$_{core}$ are defined as follows: (1) $B \longrightarrow A \mid \exists R$, (2) $R \longrightarrow P \mid P^-$, (3) $C \longrightarrow B \mid \neg B$, (4) $S \longrightarrow R \mid \neg R$, where A denotes an atomic concept, P an atomic role, B a basic concept, and C a general concept. A basic concept which can be either an atomic concept or a concept of the form $\exists R$, where R denotes a basic role which can be either an atomic role or the inverse of an atomic role.

In DL-Lite$_{core}$, an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, where $\mathcal{T}$ is a finite set of *concept inclusion assertions* of the form: $B \sqsubseteq C$; and $\mathcal{A}$ is a finite set of *membership assertions* of the form: $A(a)$, $P(a, b)$. DL-Lite$_{FR}$ extends DL-Lite$_{core}$ with inclusion assertions between roles of the form $R \sqsubseteq E$ and functionality on roles or on their inverses of the form $(functR)$. To keep the logic tractable, whenever a role inclusion $R_1 \sqsubseteq R_2$ appears in $\mathcal{T}$, neither $(functR_2)$ nor $(functR_2^-)$ can appear in it. We call assertions of the form $B_1 \sqsubseteq \neg B_2$ or $R_1 \sqsubseteq \neg R_2$ as negative inclusions (NIs).

The semantics of DL-Lite is defined in a standard way. Given an interpretation $\mathcal{I}$ and an assertion $\alpha, \mathcal{I} \vDash \alpha$ denotes that $\mathcal{I}$ is a *model* of α. An interpretation is called a *model* of an ontology $\mathcal{O}$, iff it satisfies each assertion in $\mathcal{O}$. An ontology is satisfiable if it has at least one model. An ontology $\mathcal{O}$ logically implies an assertion α, written $\mathcal{O} \models \alpha$, if all models of $\mathcal{O}$ are also models of α. The deductive closure of an ABox $\mathcal{A}$ (of a TBox $\mathcal{T}$), denoted $cl_{\mathcal{T}}(\mathcal{A})$ (resp., $cl(\mathcal{T})$), is the set of all ABox (resp., TBox) assertions α such that $(\mathcal{T} \cup \mathcal{A}) \models \alpha$ $(\mathcal{T} \models \alpha)$.

A New Operator for ABox Revision in DL-Lite

We consider the problem of ABox revision in DL-Lite. Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a new ABox $\mathcal{A}'$, suppose $\mathcal{O}$ is consistent and $\mathcal{T} \cup \mathcal{A}'$ is consistent, but $\mathcal{T} \cup \mathcal{A}' \cup \mathcal{A}$ may be inconsistent. The problem of ABox revision is, how do we modify (by deletion or insertion of assertions) A such that $\mathcal{T} \cup \mathcal{A}'$ is consistent with the modified ABox? We present a revision operator that removes one assertion from each *minimal inconsistent subset* of $\mathcal{A}$ w.r.t. $\mathcal{A}'$ and $\mathcal{T}$.

Definition 1. *Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and an ABox $\mathcal{A}'$. A minimal inconsistent subset (MIS) D of $\mathcal{A}$ w.r.t.$\mathcal{A}'$ and $\mathcal{T}$ is a sub-ABox of $\mathcal{A}$ which satisfies (1) $D \cup \mathcal{T} \cup \mathcal{A}'$ is inconsistent; (2) $\forall D' \subset D$, $D' \cup \mathcal{T} \cup \mathcal{A}'$ is consistent. We denote the set of all the MISs of $\mathcal{A}$ w.r.t. $\mathcal{A}'$ by MIS$_{\mathcal{A}'}(\mathcal{A})$ (we omit $\mathcal{T}$ to simplify the notation).*

Example 1. *(originally from (Giacomo et al. 2009)) Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a new ABox $\mathcal{A}'$, where $\mathcal{T} = \{\exists WillPlay \sqsubseteq AvailablePlayer,\ AvailablePlayer \sqsubseteq Player\ Injured \sqsubseteq \neg AvailablePlayer\}$. $\mathcal{A} = \{WillPlay(Peter, game06)\}$, $\mathcal{A}' = \{Injured(Peter)\}$. It is easy to check that there exists one MIS of $\mathcal{A}$ w.r.t. $\mathcal{A}' : \{WillPlay(Peter, game06)\}$.*

Lemma 1. *(Calvanese et al. 2010) Let $\mathcal{T} \cup \mathcal{A}$ be a DL-Lite ontology. If $\mathcal{T} \cup \mathcal{A}$ is unsatisfiable, then there is a subset $\mathcal{A}_0 \subset \mathcal{A}$ with at most two elements, such that $\mathcal{T} \cup \mathcal{A}_0$ is unsatisfiable.*

The lemma implies that every MIS of MIS$_{\mathcal{A}'}(\mathcal{A})$ contains only one assertion. Thus, to restore consistency, we can simply remove $\cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i$. However, this may delete much more information than necessary. Consider Example 1 again, we can find that Peter is injured implies that he is not an available player anymore, but he remains a player, and this would not be captured by simply removing $\cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i$. Consequently, we will add $Player(Peter)$ to the result of revision as it does not contradict $\mathcal{A}' \cup \mathcal{T}$ and it can be inferred from $\cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i \cup \mathcal{T}$.

Definition 2. *Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and an ABox $\mathcal{A}'$, a maximal consistent set S of $cl_{\mathcal{T}}(\cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i)$ w.r.t. $\mathcal{A}'$ is a sub-ABox of $cl_{\mathcal{T}}(\cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i)$ which satisfies (1) $S \cup \mathcal{T} \cup \mathcal{A}'$ is consistent; (2) $\forall \alpha \in cl_{\mathcal{T}}(\cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i)$ and $\alpha \notin S$, $S \cup \{\alpha\} \cup \mathcal{T} \cup \mathcal{A}'$ is inconsistent.*

Theorem 1. *Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and an ABox $\mathcal{A}'$, the maximal consistent set S of $cl_{\mathcal{T}}(\cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i)$ w.r.t. $\mathcal{A}'$ is uniquely defined.*

Definition 3. *Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$. The revision operator $\circ$ for $\mathcal{O}$ is defined as follows: for each ABox $\mathcal{A}'$*

$$(\mathcal{T} \cup \mathcal{A}) \circ \mathcal{A}' = \mathcal{T} \cup (\mathcal{A} \setminus \cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i) \cup S \cup \mathcal{A}'$$

We can show that the deductive closure of the resulting ABox of our operator is the same as the ABox obtained by the algorithm *FastEvo* given in (Calvanese et al. 2010).

Consider Example 1 again. We have $(\mathcal{T} \cup \mathcal{A}) \circ \mathcal{A}' = \mathcal{T} \cup \{Injured(Peter), Player(Peter))\}$.

Theorem 2. *The result of ABox revision $(\mathcal{T} \cup \mathcal{A}) \circ \mathcal{A}'$ is uniquely defined.*

A Graph-based Algorithm

In order to compute $MIS_{\mathcal{A}'}(\mathcal{A})$, we need to compute all the NIs axioms implied by $\mathcal{T}$. We define a hierarchical graph to compactly represent $cl(\mathcal{T})$.

Definition 4. *(HG) Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, a hierarchical graph (HG for short) corresponding to $\mathcal{O}$, denoted as $G_{\langle \mathcal{T}, \mathcal{A} \rangle} = \langle V, E \rangle$, is defined by the following rules (functionality axioms are excluded and will be treated separately):*

1. *$\langle C, B \rangle \in E$ and $C, B \in V$ if $B \sqsubseteq C$ is in $\mathcal{T}$;*
2. *$\langle \neg C, B \rangle \in E$ and $\neg C, B \in V$ if $B \sqsubseteq \neg C$ is in $\mathcal{T}$;*
3. *$\langle \exists R_2, \exists R_1 \rangle \in E$, $\langle \exists R_2^-, \exists R_1^- \rangle \in E$ and $\exists R_2, \exists R_1, \exists R_2^-, \exists R_1^- \in V$ if $R_1 \sqsubseteq R_2$ is in $\mathcal{T}$;*
4. *$\langle A, a \rangle \in E$ and $A, a \in V$ if $A(a)$ is in $\mathcal{A}$;*
5. *$\langle \exists P, a \rangle \in E$, $\langle \exists P^-, b \rangle \in E$ and $\exists P, \exists P^-, a, b \in V$ if $P(a, b)$ is in $\mathcal{A}$.*

Our algorithm *GraphRevi* takes as input $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and $\mathcal{A}'$. Let $\mathcal{A}_{all} = \mathcal{A} \cup \mathcal{A}'$. Then, it computes the set D of all the membership assertions in $\mathcal{A}$ that conflict with functionality axioms and $\mathcal{A}'$. It constructs a HG from $\langle \mathcal{T}, \mathcal{A}_{all} \setminus D \rangle$ and uses a function *Search* to compute the set of all the membership assertions in $\mathcal{A}$ that conflict with some NIs assertions and some assertions in $\mathcal{A}'$, and use this set to update D. Thus, D is actually $\cup_{A_i \in MIS_{\mathcal{A}'}(\mathcal{A})} A_i)$. Let $M = cl_{\mathcal{T}}(D) \setminus D$. The algorithm deletes all the membership assertions in M that conflict with NIs assertions and $\mathcal{A}'$. Finally, $\mathcal{T} \cup (\mathcal{A} \setminus D) \cup M \cup \mathcal{A}'$ is the result of revision.

Theorem 3. *Algorithm GraphRevi runs in polynomial time and $(\mathcal{T} \cup \mathcal{A}) \circ \mathcal{A}'$=GraphRevi($\mathcal{T}$, $\mathcal{A}$, $\mathcal{A}'$).*

Conclusion

In this paper, we proposed a new operator for ABox revision in DL-Lite$_{FR}$ and a graph-based algorithm to implement this revision operator. As a future work, we will discuss the logical properties of our revision operator. We will also carry out comprehensive experiments to verify the efficiency and effectiveness of our approach.

Acknowledgement

Guilin Qi is partially supported by NSFC (61003157), Jiangsu Science Foundation (BK2010412), Excellent Youth Scholars Program of Southeast University, and Doctoral Discipline Foundation for Young Teachers in the Higher Education Institutions of Ministry of Education (No. 20100092120029).

References

Calvanese, D.; Kharlamov, E.; Nutt, W.; and Zheleznyakov, D. 2010. Evolution of dl-lite knowledge bases. In *Proc. of ISWC*, 112–128.

Giacomo, G. D.; Lenzerini, M.; Poggi, A.; and Rosati, R. 2009. On instance-level update and erasure in description logic ontologies. *J. Log. Comput.* 19(5):745–770.

Kharlamov, E., and Zheleznyakov, D. 2011. Capturing instance level ontology evolution for dl-lite. In *Proc. of ISWC*, 321–337.

Qi, G., and Du, J. 2009. Model-based revision operators for terminologies in description logics. In *Proc. of IJCAI*, 891–897.