OntoMerge: A System for Merging DL-Lite Ontologies

Zhe Wang[1], Kewen Wang[2], Yifan Jin[2], and Guilin Qi[3,4]

[1] University of Oxford, United Kingdom
[2] Griffith University, Australia
[3] Southeast University, China
[4] State Key Laboratory for Novel Software Technology
Nanjing University, Nanjing, China

Abstract. Merging multi-sourced ontologies in a consistent manner is an important and challenging research topic. In this paper, we propose a novel approach for merging DL-Lite$^{\mathcal{N}}_{bool}$ ontologies by adapting the classical model-based belief merging approach, where the minimality of changes is realised via a semantic notion, *model distance*. Instead of using classical DL models, which may be infinite structures in general, we define our merging operator based on a new semantic characterisation for DL-Lite. We show that subclass relation *w.r.t.* the result of merging can be checked efficiently via a QBF reduction. We present our system OntoMerge, which effectively answers subclass queries on the resulting ontology of merging, without first computing the merging results. Our system can be used for answering subclass queries on multiple ontologies.

1 Introduction

Ontologies are widely used for sharing and reasoning over domain knowledge, and their underlying formalisms are often description logics (DLs). To effectively answer queries, ontologies from heterogeneous sources and contributed by various authors are often needed. However, ontologies developed by multiple authors under different settings may contain overlapping, conflicting and incoherent domain knowledge. The ultimate goal of ontology merging is to obtain a single consistent ontology that preserves as much knowledge as possible from two or more heterogeneous ontologies. This is in contrast to ontology matching [5], whose goal is to align entities (with different name) between ontologies, and which is often a pre-stage of ontology merging.

Existing merging systems often adopt formula-based approaches to deal with logical inconsistencies [10; 9; 14]. Most of such approaches can be described as follows: the system first combine the ontologies by taking their union; then, if any inconsistency is detected (through a standard reasoning), it pinpoints the axioms which (may) cause inconsistency; and finally, remove certain axioms to retain consistency. However, such an approach is sometimes unsatisfactory because it is not fine-grained either in the way it measures the minimality of changes, and thus it is often unclear how close the result of merging is to the source ontologies semantically; or in the way it resolve inconsistency. In [12], an attempt is made to provide some semantic justification for the minimality of changes, however, the result of merging is still syntax-dependant and is often a set of ontologies.

On the other hand, model-based merging operators have been intensively studied in propositional logic, which are syntax-independent and usually satisfy more rationality postulates than formula-based ones. However, a major challenge in adapting model-based merging techniques to DLs is that DL models are generally infinite structures and the number of models of a DL ontology is infinite. Several notions of model distance are defined on classical DL models for ontology revision [13]. Mathematically, it is possible to define a distance o classical DL models. Such a distance is computationally limited as it is unclear how to develop an algorithm for the resulting merging operator. A desirable solution is to define ontology merging operators based on a suitable finite semantic characterisation instead of classical DL models.

In this paper, we focus on merging ontologies expressed as DL-Lite TBoxes, which can be also accompanied with the ontology-based data access (OBDA) framework for data integration [2]. We propose a novel approach for merging ontologies by adapting a classical model-based belief merging approach, where the minimality of changes is realised via a semantic notion, model distance. Instead of using classical DL models, which may be infinite structures in general, we define our merging operator based on the notion of *types*. We show that subclass relation *w.r.t.* the result of merging can be checked efficiently via a QBF reduction, which allows us to make use of the off-the-shelf QBF solvers [8]. We present our system OntoMerge, which effectively answers subclass queries on merging results, without first computing the merging results. Our system can be used to answer subclass queries on multiple ontologies.

2 A New Semantic Characterisation

In our approach, it is sufficient to consider a finite yet large enough signature. A *signature* $\mathcal{S}$ is a union of four disjoint finite sets $\mathcal{S}_C$, $\mathcal{S}_R$, $\mathcal{S}_I$ and $\mathcal{S}_N$, where $\mathcal{S}_C$ is the set of atomic concepts, $\mathcal{S}_R$ is the set of atomic roles, $\mathcal{S}_I$ is the set of individual names and $\mathcal{S}_N$ is the set of natural numbers in $\mathcal{S}$. We assume 1 is always in $\mathcal{S}_N$.

Formally, given a signature $\mathcal{S}$, a DL-Lite$^{\mathcal{N}}_{bool}$ language has the following syntax [1]:

$$R \leftarrow P \mid P^- \qquad\qquad S \leftarrow P \mid \neg P$$

$$B \leftarrow \top \mid A \mid \geqslant n\ R \qquad C \leftarrow B \mid \neg C \mid C_1 \sqcap C_2$$

where $n \in \mathcal{S}_N$, $A \in \mathcal{S}_C$ and $P \in \mathcal{S}_R$. B is called a *basic concept* and C is called a *general concept*. $\mathcal{B}_\mathcal{S}$ denotes the set of basic concepts on $\mathcal{S}$. We write $\bot$ for $\neg\top$, $\exists R$ for $\geq 1\ R$, and $C_1 \sqcup C_2$ for $\neg(\neg C_1 \sqcap \neg C_2)$. Let $R^+ = P$, where $P \in \mathcal{S}_R$, whenever $R = P$ or $R = P^-$. A *TBox* $\mathcal{T}$ is a finite set of concept *axioms* of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are general concepts. An *ABox* $\mathcal{A}$ is a finite set of membership *assertions* of the form $C(a)$ or $S(a, b)$, where a, b are individual names. In this paper, an ontology is represented as a DL TBox.

The classical DL semantics are given by models. A TBox $\mathcal{T}$ is *consistent* with an ABox $\mathcal{A}$ if $\mathcal{T} \cup \mathcal{A}$ has at least one model. A concept or role is *satisfiable* in $\mathcal{T}$ if it has a non-empty interpretation in some model of $\mathcal{T}$. A TBox $\mathcal{T}$ is *coherent* if all atomic concepts and atomic roles in $\mathcal{T}$ are satisfiable. Note that a coherent TBox must be consistent. TBox $\mathcal{T}$ *entails* an axiom $C \sqsubseteq D$, written $\mathcal{T} \models C \sqsubseteq D$, if all models of $\mathcal{T}$ satisfy $C \sqsubseteq D$. Two TBoxes $\mathcal{T}_1, \mathcal{T}_2$ are *equivalent*, written $\mathcal{T}_1 \equiv \mathcal{T}_2$, if they have the same models.

Now, we introduce a semantic characterisation for DL-Lite TBoxes in terms of *types*. A *type* $\tau \subseteq \mathcal{B}_\mathcal{S}$ is a set of basic concepts over $\mathcal{S}$, such that $\top \in \tau$, and $\geqslant n\ R \in \tau$ implies $\geqslant m\ R \in \tau$ for each pair $m, n \in \mathcal{S}_N$ with $m < n$ and each (inverse) role $R \in \mathcal{S}_R \cup \{\, P^- \mid P \in \mathcal{S}_R \,\}$. Type τ *satisfies* basic concept B if $B \in \tau$, $\neg C$ if τ does not satisfy C, and $C_1 \sqcap C_2$ if τ satisfies both C_1 and C_2. Given a TBox $\mathcal{T}$, type τ satisfies $\mathcal{T}$ if τ satisfies concept $\neg C_1 \sqcup C_2$ for each axiom $C_1 \sqsubseteq C_2$ in $\mathcal{T}$.

For a TBox $\mathcal{T}$, define $\mathsf{TM}(\mathcal{T})$ to be the maximal set of types satisfying the following conditions: (1) all the types in $\mathsf{TM}(\mathcal{T})$ satisfy $\mathcal{T}$; (2) for each type $\tau \in \mathsf{TM}(\mathcal{T})$ and each $\exists R$ in τ, there exists a type $\tau' \in \mathsf{TM}(\mathcal{T})$ (possibly $\tau' = \tau$) containing $\exists R^-$. A type τ is called a *type model* (T-model) of $\mathcal{T}$ if $\tau \in \mathsf{TM}(\mathcal{T})$. Note that $\mathsf{TM}(\mathcal{T})$ is uniquely defined for each TBox $\mathcal{T}$. Note that for a coherent TBox $\mathcal{T}$, $\mathsf{TM}(\mathcal{T})$ is exactly the set of all types satisfying $\mathcal{T}$. Let $\mathbf{TM}(\Pi) = \mathsf{TM}(\mathcal{T}_1) \times \cdots \times \mathsf{TM}(\mathcal{T}_n)$ for $\Pi = \langle \mathcal{T}_1, \ldots, \mathcal{T}_n \rangle$.

Proposition 1. *Given a TBox $\mathcal{T}$, we have the following results:*

- *$\mathcal{T}$ is consistent iff $\mathsf{TM}(\mathcal{T}) \neq \emptyset$.*
- *For a general concept C, C is satisfiable wrt $\mathcal{T}$ iff there exists a T-model in $\mathsf{TM}(\mathcal{T})$ satisfying C.*
- *For two general concepts C, D, $\mathcal{T} \models C \sqsubseteq D$ iff either $\mathsf{TM}(\mathcal{T}) = \emptyset$ or all T-models in $\mathsf{TM}(\mathcal{T})$ satisfy $C \sqsubseteq D$.*
- *$\mathcal{T} \equiv \mathcal{T}'$ iff $\mathsf{TM}(\mathcal{T}) = \mathsf{TM}(\mathcal{T}')$, for any TBox $\mathcal{T}'$.*

Given a type τ, an individual a and an ABox $\mathcal{A}$, we say τ is a *type of a w.r.t. $\mathcal{A}$* if there is a model $\mathcal{I}$ of $\mathcal{A}$ such that $\tau = \{B \mid a^\mathcal{I} \in B^\mathcal{I}, B \in \mathcal{B}_\mathcal{S}\}$. For example, given $\mathcal{A} = \{A(a), \neg B(b), C(c)\}$, type $\tau = \{A, B\}$ is a type of a, but not a type of either b or c in $\mathcal{A}$. For convenience, we will say a *type of a* when the ABox $\mathcal{A}$ is clear from the context. Let $\mathsf{TM}_a(\mathcal{A})$ be the set of all the types of a in $\mathcal{A}$ if a occurs in $\mathcal{A}$; and otherwise, $\mathsf{TM}_a(\mathcal{A})$ be the set of all the types. A set M of T-models *satisfies* an ABox $\mathcal{A}$ if there is a type of a in M, *i.e.*, $M \cap \mathsf{TM}_a(\mathcal{A}) \neq \emptyset$, for each individual a in $\mathcal{A}$.

Proposition 2. *Given a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, $\mathcal{T} \cup \mathcal{A}$ is consistent iff $\mathsf{TM}(\mathcal{T}) \cap \mathsf{TM}_a(\mathcal{A}) \neq \emptyset$ for each a in $\mathcal{A}$.*

3 Merging Operator

In this section, we introduce an approach to merging DL-Lite ontologies to obtain a coherent unified ontology.

An ontology *profile* is of the form $\Pi = \langle \mathcal{T}_1, \ldots, \mathcal{T}_n \rangle$, where $\mathcal{T}_i$ is the ontology from the source n.o. i $(1 \leq i \leq n)$. There are two standard definitions of integrity constraints (ICs) in the classical belief change literature [3], the consistency- and entailment-based definitions. We also allow two types of ICs for merging, namely the *consistency constraint* (CC), expressed as a set $\mathcal{A}_c$ of data, and the *entailment constraint* (EC), expressed as a TBox $\mathcal{T}_e$. We assume the IC is self-consistent, that is, $\mathcal{T}_e \cup \mathcal{A}_c$ is always consistent. For an ontology profile Π, a CC $\mathcal{A}_c$ and a EC $\mathcal{T}_e$, an ontology *merging* operator is a mapping $(\Pi, \mathcal{T}_e, \mathcal{A}_c) \mapsto \nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c)$, where $\nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c)$ is a TBox, *s.t.* $\nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c) \cup \mathcal{A}_c$ is consistent, and $\nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c) \models \mathcal{T}_e$.

In classical model-based merging approaches, merging operators are often defined by certain notions of *model distances* [11; 6]. We use $S \triangle S'$ to denote the symmetric difference between two sets S and S', *i.e.*, $S \triangle S' = (S \setminus S') \cup (S' \setminus S)$. Given a set S and a tuple $\boldsymbol{S} = \langle S_1, \ldots, S_n \rangle$ of sets, the *distance* between S and $\boldsymbol{S}$ is defined to be a tuple $\boldsymbol{d}(S, \boldsymbol{S}) = \langle S \triangle S_1, \ldots, S \triangle S_n \rangle$. For two n-element distances $\boldsymbol{d}$ and $\boldsymbol{d}'$, $\boldsymbol{d} \preceq \boldsymbol{d}'$ if $d_i \subseteq d'_i$ for each $1 \leq i \leq n$, where d_i is the i-th element in $\boldsymbol{d}$. Given two sets S and S', define $\sigma(S, S') = S$ if S' is empty, and otherwise, $\sigma(S, S') = \{ e_0 \in S \mid \exists e'_0 \in S' \; s.t. \; \forall e \in S, \forall e' \in S', d(e, e') \not\prec d(e_0, e'_0) \}$. In [6], given a collection $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ of propositional formulas, and some ECs expressed as a propositional theory μ, the result of merging $\varphi_1, \ldots, \varphi_n$ *w.r.t.* μ is the theory whose models are exactly $\sigma(\mathsf{mod}(\mu), \mathsf{mod}(\Psi))$, *i.e.*, those models satisfying μ and having minimal distance to Ψ.

Inspired by classical model-based merging, we introduce a merging operator in terms of T-models. For an ontology profile Π and an EC $\mathcal{T}_e$, we could define the T-models of the merging to be a subset of $\mathsf{TM}(\mathcal{T}_e)$ (so that $\mathcal{T}_e$ is entailed) consisting of those T-models which have minimal distance to Π, *i.e.*, $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi))$. However, this straightforward adoption does not take the CC into consideration, and the merging result obtained in this way may not be coherent. For example, let $\mathcal{T}_1 = \{A \sqsubseteq \neg B\}$, $\mathcal{T}_2 = \{\top \sqsubseteq B\}$, $\mathcal{T}_e = \emptyset$, and $\mathcal{A}_c = \{A(a), B(a)\}$. Then, $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\langle \mathcal{T}_1, \mathcal{T}_2 \rangle))$ consists of only one type $\{B\}$. Clearly, the corresponding TBox $\{A \sqsubseteq \bot, \top \sqsubseteq B\}$ does not satisfy the CC, and it is not coherent.

Note that in the above example, once the merging result satisfies the CC, then it is also coherent, because both concepts A and B are satisfiable. In general, it is also the case that coherency can be achieved by applying certain CC to merging. We introduce an auxiliary ABox $\mathcal{A}^\dagger$ in addition to the initial CC $\mathcal{A}_c$, in which each concept and each role is explicitly asserted with a member. That is, $\mathcal{A}^\dagger = \{A(a) \mid A \in \mathcal{S}_C, a \in \mathcal{S}_I \text{ is a fresh individual for } A\} \cup \{P(b,c) \mid P \in \mathcal{S}_R, b, c \in \mathcal{S}_I \text{ are fresh individuals for } P\}$. As assumed, $\mathcal{S}_I$ is large enough for us to take these auxiliary individuals. From the definition of CCs, the merged TBox $\mathcal{T}$ must be consistent with all the assertions in $\mathcal{A}^\dagger$, which assures all the concepts and roles in $\mathcal{T}$ to be satisfiable. Based on this observation, we have the following lemma.

Lemma 1. *$\mathcal{T}$ is coherent iff $\mathcal{T} \cup \mathcal{A}^\dagger$ is consistent for any TBox $\mathcal{T}$.*

To ensure the coherence of merging, we only need to include $\mathcal{A}^\dagger$ into the CC.

For the merging to be consistent with the CC $\mathcal{A}_c$, from Proposition 2, the T-model set M of the merging needs to satisfy $\mathcal{A}_c$. That is, M needs to contain a type of a for each individual a in $\mathcal{A}_c$. However, $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi))$ does not necessarily satisfy this condition, as can be seen from the above example: $\mathsf{TM}_a(\mathcal{A}_c)$ consists of a single type $\{A, B\}$ and $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi)) \cap \mathsf{TM}_a(\mathcal{A}_c) = \emptyset$. Intuitively, for the merging to satisfy the CC, type $\{A, B\}$ need to be added to the T-models of merging. In general, the T-models of merging can be obtained by extending (if necessary) the set $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi))$ with at least one type of a *w.r.t.* $\mathcal{A}_c$ for each individual a in $\mathcal{A}_c$, and if there are multiple such types, choose those with minimal distances.

Based on the above intuitions, the definition of TBox merging is presented as follows.

Definition 1. *Let Π be an ontology profile, $\mathcal{T}_e$ be a TBox, and $\mathcal{A}_c$ be an ABox. Denote $\mathcal{A}^* = \mathcal{A}_c \cup \mathcal{A}^\dagger$. The result of merging Π w.r.t. the EC $\mathcal{T}_e$ and the CC $\mathcal{A}_c$, denoted $\nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c)$, is defined as follows*

$$\begin{aligned}\mathsf{TM}(\nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c)) = {} & \sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi)) \cup \\ & \bigcup_{a \text{ occurs in } \mathcal{A}^*} \sigma(\mathsf{TM}(\mathcal{T}_e) \cap \mathsf{TM}_a(\mathcal{A}^*), \mathbf{TM}(\Pi)).\end{aligned}$$

From the definition, the T-models of the merging are constituted with two parts. The first part consists of those T-models of $\mathcal{T}_e$ (for the satisfaction of the EC) with minimal distances to Π. The second part consists of types of a, for each individual a in $\mathcal{A}^*$, which are added to the first part for the satisfaction of the CC. These types are also required to be T-models of $\mathcal{T}_e$ and have minimal distances to Π. It is clear from Proposition 1 that the result of merging is unique up to TBox equivalence.

4 QBF Reduction

In this section, we consider a standard reasoning problem for ontology merging, namely the *subclass queries*: whether or not the result of merging entails a subclass relation $C \sqsubseteq D$. We present a QBF reduction for this problem, which allows us to make use of the off-the-shelf QBF solvers [8]. We assume that every TBox in the ontology profile is coherent, and in this case, the T-models of a TBox $\mathcal{T}$ are exactly those satisfying $\mathcal{T}$.

We achieve the reduction in three steps. Firstly, we introduce a novel propositional transformation for DL-Lite TBoxes. The transformation is inspired by [1], which contains a transformation from a DL-Lite TBox into a theory in the one variable fragment of first order logic. Considering T-models instead of classical DL models allows us to obtain a simpler transformation to propositional logic than theirs to first order logic.

Function $\phi(\cdot)$ maps a basic concept to a propositional variable, and a general concept (resp., a TBox axiom) to a propositional formula.

$$\begin{aligned}
&\phi(\bot) = \bot, \quad \phi(A) = p_A, \quad \phi(\geqslant n\ R) = p_{nR}, \\
&\phi(\neg C) = \neg\phi(C), \qquad\qquad\quad \phi(C_1 \sqcap C_2) = \phi(C_1) \wedge \phi(C_2), \\
&\phi(C_1 \sqsubseteq C_2) = \phi(C_1) \to \phi(C_2).
\end{aligned}$$

Here, p_A and p_{nR} are propositional variables. We use $V_{\mathcal{S}}$ to denote the set of propositional variables corresponding to the basic concepts over $\mathcal{S}$, and we omit the subscript $\mathcal{S}$ in what follows for simplicity. We can see that $\phi(\cdot)$ is a bijective mapping between the set of DL-Lite$^{\mathcal{N}}_{bool}$ general concepts and the set of propositional formulas only referring to symbols in $V_{\mathcal{S}}$ and boolean operators $\neg$, $\wedge$ and $\to$.

Naturally, given the mapping $\phi(\cdot)$, an arbitrary propositional model may not correspond to a type. We define a formula η whose models are exactly the set of types. Let

$$\eta = \bigwedge_{R^+ \in \mathcal{S}_R} \bigwedge_{\substack{m,n \in \mathcal{S}_N \text{ with } m<n \\ \text{and } m<k<n \text{ for no } k \in \mathcal{S}_N}} p_{nR} \to p_{mR}.$$

Then, $\mathsf{mod}(\eta) = \{\phi(\tau) \mid \tau \text{ is a type}\}$ where $\phi(S)$ stands for for $\{\phi(B_1) \mid B \in S\}$ for a set S of basic concepts.

Given a coherent DL-Lite TBox $\mathcal{T}$, let $\phi(\mathcal{T}) = \bigwedge_{\alpha \in \mathcal{T}} \phi(\alpha) \wedge \eta$. The models of $\phi(\mathcal{T})$ correspond to the T-models of $\mathcal{T}$. For a DL-Lite ABox $\mathcal{A}$ and an individual name a in $\mathcal{A}$, let

$$\phi_a(\mathcal{A}) = \bigwedge_{C(a) \in \mathcal{A}} \phi(C) \wedge \bigwedge_{P \in \mathcal{S}_R} (p_{uP} \wedge p_{vP^-})$$

where u and v are respectively, the maximal number in $\mathcal{S}_N$ *s.t.* $u \leq |\{b_i \mid P(a, b_i) \in \mathcal{A}\}|$ and $v \leq |\{b_i \mid P(b_i, a) \in \mathcal{A}\}|$. Note that we are not transforming an ABox into a propositional theory, but using the encodings $\phi_a(\mathcal{A})$ as constraints over the models.

It is worth noting that the sizes of $\phi(\mathcal{T})$ and $\phi(a, \mathcal{A})$ are both polynomial in the size of $\mathcal{T} \cup \mathcal{A}$. The intuition behind $\phi(\mathcal{T})$ and $\phi(a, \mathcal{A})$ can be shown by the following lemma.

Lemma 2. *Given a coherent TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, then,*

1. $\mathsf{mod}(\phi(\mathcal{T})) = \{\phi(\tau) \mid \tau \in \mathsf{TM}(\mathcal{T})\}$*;*
2. $\mathsf{mod}(\phi_a(\mathcal{A}) \wedge \eta) = \{\phi(\tau) \mid \tau \in \mathsf{TM}_a(\mathcal{A})\}$*.*

This transformation essentially allows us to build a connection between our merging operator and propositional belief merging.

Secondly, as we have a transformation from T-models to propositional models, we can encode (minimal) distances between them using QBFs, by extending the encoding in [4], which was introduced for a different purpose. In particular, we need to encode the distances between the models of ϕ and the models of

$\varphi_1, \ldots, \varphi_n$ $(n \geq 1)$, where ϕ and φ_i's are propositional formulas in signature V. We make n-fresh copies of V to, informally, encode the models of $\varphi_1, \ldots, \varphi_n$, respectively; let $V^i = \{p^i \mid p \in V\}$ for $1 \leq i \leq n$ where each p^i is a fresh variable for p, and $V^N = \bigcup_{1 \leq i \leq n} V^i$. For a propositional formula φ and $1 \leq i \leq n$, φ^i denotes the formula obtained from φ by replacing each occurrence of p with p^i. We also need another n-fresh copies of V, $V_d^i = \{p_d^i \mid p \in V\}$ to represent the distances. An assignment to V_d^i is expected to capture the difference between a model of ϕ and a model of φ_i, and in particular, p_d^i is assigned true if the truth values of p and p^i are different. Let $V_d^N = \bigcup_{1 \leq i \leq n} V_d^i$. Define

$$F(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle) = \phi \wedge \bigwedge_{1 \leq i \leq n} \Big(\varphi_i^i \wedge \bigwedge_{p \in V} \big((p \leftrightarrow \neg p^i) \rightarrow p_d^i \big) \Big).$$

A model M of $F(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)$ consists of the assignments to three sets of variables V, V^N and V_d^N. For a set $S \subseteq V \cup V^N \cup V_d^N$, $m(S)$ is the set obtained from S by eliminating the super- and subscripts. Then, $M \cap V$ is a model of ϕ, and $m(M \cap V^i)$ is a model of φ_i. From $(p \leftrightarrow \neg p^i) \rightarrow p_d^i$, $m(M \cap V_d^i)$ *subsumes* the symmetric difference between the former two models. We use $\rightarrow$ instead of $\leftrightarrow$ here, as we will further constraint the assignments of V_d^N to minimal distances.

Furthermore, define a QBF

$$\begin{aligned} D(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle) = & (\exists V \exists V^N F(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)) \wedge \\ & \bigwedge_{\substack{p \in V \\ 1 \leq i \leq n}} \Big(p_d^i \rightarrow \neg \exists V \exists V^N \big(F(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle) \wedge (p \leftrightarrow p^i) \big) \Big), \end{aligned}$$

where $\exists V$ with $V = \{p_1, \ldots, p_k\}$ is an abbreviation for $\exists p_1 \ldots \exists p_k$. A model M_d of $D(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)$ is an assignment to V_d^N representing a minimal distance between the models of ϕ and the model tuples of $\langle \varphi_1, \ldots, \varphi_n \rangle$. The first conjunct of $D(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)$ says that there is a model of ϕ and a model tuple of $\langle \varphi_1, \ldots, \varphi_n \rangle$ such that the distance between them is subsumed by $m(M_d)$. The second conjunct checks that $m(M_d)$ is minimal, *i.e.*, there is no distance properly subsumed by $m(M_d)$.

Lemma 3. *Given propositional formulas ϕ and $\varphi_1, \ldots, \varphi_n$, let $MD(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)$ be the set of minimal distances (w.r.t. $\preceq$) between ϕ and $\langle \varphi_1, \ldots, \varphi_n \rangle$ (of the form $\langle M \triangle M_1, \ldots, M \triangle M_n \rangle$ with $M \in \mathsf{mod}(\phi)$ and $M_i \in \mathsf{mod}(\varphi_i)$). Then,*

$$\begin{aligned} \mathsf{mod}(D(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)) = \{ M_d \subseteq V_d^N \mid \exists \boldsymbol{d} \in MD(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle) \ s.t. \\ m(M_d \cap V_d^i) = \boldsymbol{d}_i \ for\ each\ 1 \leq i \leq n \}. \end{aligned}$$

Finally, with the encoding of minimal distances, we can encode the T-models of the merging and we are ready to encode the entailment relation. Given an ontology profile $\mathit{\Pi} = \langle \mathcal{T}_1, \ldots, \mathcal{T}_n \rangle$ and a TBox $\mathcal{T}_e$, all the types in $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\mathit{\Pi}))$ satisfy TBox axiom α if and only if QBF $\neg \exists V_d^N E(\mathit{\Pi}, \mathcal{T}_e, \alpha)$ evaluates to true, with

$$\begin{aligned} E(\mathit{\Pi}, \mathcal{T}_e, \alpha) = D(\phi(\mathcal{T}_e), \langle \phi(\mathcal{T}_1), \ldots, \phi(\mathcal{T}_n)\rangle) \wedge \\ \neg\forall V\Big(\big(\exists V^N F(\phi(\mathcal{T}_e), \langle \phi(\mathcal{T}_1), \ldots, \phi(\mathcal{T}_n)\rangle)\big) \rightarrow \phi(\alpha)\Big). \end{aligned}$$

This QBF can be understood as follows. A model M of $E(\mathit{\Pi}, \mathcal{T}_e, \alpha)$ is an assignment to V_d^N, and represents, by the first conjunct of $E(\mathit{\Pi}, \mathcal{T}_e, \alpha)$, a minimal distance between the T-models of $\mathcal{T}_e$ and the T-model tuples of $\mathit{\Pi}$. The second conjunct states the non-entailment, that is, not all of the T-models of $\mathcal{T}_e$ having such a distance satisfy α. The QBF as a whole essentially says that there does not exists a minimal distance $\boldsymbol{d}$ such that a type (in $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\mathit{\Pi}))$) selected with $\boldsymbol{d}$ fails to satisfy α. Similarly, given an ABox $\mathcal{A}$ and an individual a in $\mathcal{A}$, all the types in $\sigma(\mathsf{TM}(\mathcal{T}_e) \cap \mathsf{TM}_a(\mathcal{A}), \mathbf{TM}(\mathit{\Pi}))$ satisfy α iff QBF $\neg\exists V_d^N E_a(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}, \alpha)$ evaluates to true, where $E_a(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}, \alpha)$ is obtained from $E(\mathit{\Pi}, \mathcal{T}_e, \alpha)$ by replacing $\phi(\mathcal{T}_e)$ with $\phi(\mathcal{T}_e) \wedge \phi_a(\mathcal{A})$.

Now, we can reduce the subclass query answering for TBox merging to QBF as follows.

Theorem 1. *Let $\mathit{\Pi}$ be an ontology profile with coherent source TBoxes, $\mathcal{T}_e$ be a TBox and $\mathcal{A}_c$ be an ABox in DL-Lite$_{bool}^{\mathcal{N}}$. Let $\mathcal{A}^* = \mathcal{A}_c \cup \mathcal{A}^\dagger$. Given a TBox axiom α, we have $\nabla(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}_c) \models \alpha$ iff the following QBF evaluates to true*

$$\neg\exists V_d^N \big(E(\mathit{\Pi}, \mathcal{T}_e, \alpha) \vee \bigvee_{a \textit{ occurs in } \mathcal{A}^*} E_a(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}^*, \alpha)\big). \qquad (1)$$

5 System Architecture

We have implemented the algorithm for answering subclass queries in ontology merging, called *OntoMerge*, and it is publicly available for test at `http://www.ict.griffith.edu.au/~kewen/OntoMerge/`. The ultimate goal of our system OntoMerge is to transform the entailment over merged ontologies to the validity of QBFs as given in Eq.(1). If the input ontologies are $\mathcal{T}_1, \ldots, \mathcal{T}_n$ and the query is α, then the corresponding QBF can be split into two parts: one is the formula $E(\mathit{\Pi}, \mathcal{T}_e, \alpha)$, and the other is a disjunction of formulas of the form $E_a(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}^*, \alpha)$. The size of the first part is only determined by the sizes of the input ontologies, the input EC and the query α, and the size of α is often small compared to the ontologies. In the second part of the resulting QBF, the number of disjuncts is essentially determined by the number of unsatisfiable concepts in the union of input ontologies (as will be explained later) and the input CC. Thus, the number of unsatisfiable concepts plays a crucial role in the complexity of the reduction algorithm.

In OntoMerge, we first check whether the given ontologies can be simply jointed without causing any incoherency using an off-the-shelf DL reasoner (HermiT [7] is used in the current program). The set of unsatisfiable concepts will be stored for being used later. If the union of input ontologies is coherent, the query answering can be done by the DL reasoner; otherwise, the system generates the QBF specified in Eq.(1). For this purpose, the system will first scan all input ontologies to obtain the set of basic concepts occurring in these ontologies and assign a propositional variable with each basic concept. Then the QBF is generated in QBF 1.0 format[5]. The structure of OntoMerge is depicted in Figure 1.

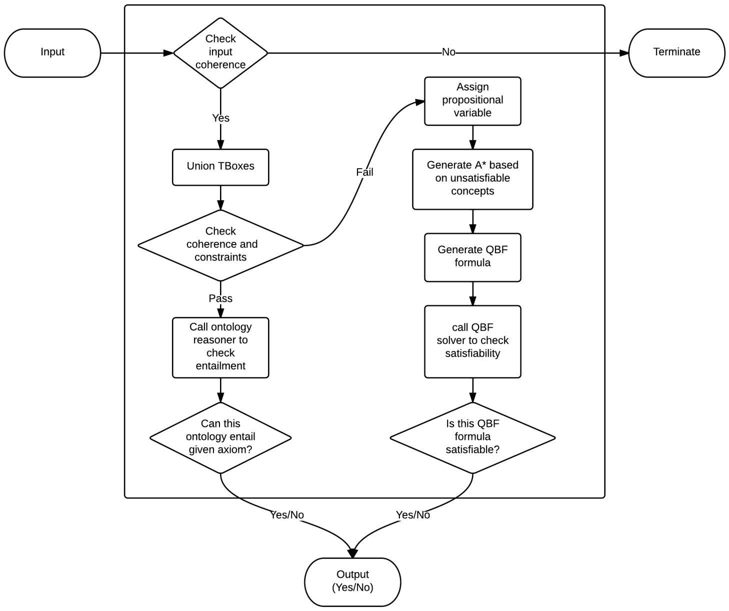
Input
Check input coherence
No
Terminate
Yes
Union TBoxes
Check coherence and constraints
Fail
Assign propositional variable
Generate A* based on unsatisfiable concepts
Generate QBF formula
Pass
Call ontology reasoner to check entailment
call QBF solver to check satisfiability
Can this ontology entail given axiom?
Is this QBF formula satisfiable?
Yes/No
Yes/No
Output (Yes/No)

Fig. 1. System Structure of OntoMerge

However, as most of efficient QBF solvers accept only input QBFs in the prenex normal form, the QBF generated in this way cannot be directly fed to a QBF solver. So we need to convert the QBF into the prenex normal form first. Unfortunately, the standard translation from a given QBF into its prenex normal form is very inefficient and thus several heuristics based on the specific QBF are used to optimize the efficiency in our implementation. For instance, from Eq.(1), we can see that the major source for introducing a large number of new variables is from the construction of the ABox $\mathcal{A}^*$. So we introduce new variables for only those concepts that are unsatisfiable and add new assertions for such new variables to $\mathcal{A}^*$ in the reduction algorithm. This optimization significantly reduces the number of new variables.

[5] `http://qbflib.org/boole.html`

Once the QBF is generated, a publicly available program is used to convert it from QBF 1.0 format to QDIMACS format[6] or ISCAS format[7] and then an efficient QBF solver is used to decide the validity of the QBF.

6 Experimental Results

To test the efficiency of OntoMerge, we used the DL-Lite$_{bool}^{\mathcal{N}}$ fragment of the medical ontology Galen[8], which is of medium size. Our experimental results show that the system is relatively efficient, while further optimization is still under way. Specifically, various randomly modified fragments of Galen were merged using OntoMerge and the following three types of experiments were conducted:

- Fixed total number of axioms in the input ontologies but varied number of unsatisfiable concepts.
- Fixed total number of unsatisfiable concepts but varied total number of axioms in the input ontologies.
- Fixed number of unsatisfiable concepts but varied total number of input ontology axioms.

A PC with Intel Core 2 Duo E8400, 4GB RAM, running Linux Mint 13 64bit, and CirQit QBF solver [8] were used in our tests. For each test, the time is limited to one hour (i. e. the program will be terminated after one hour no matter a result is returned or not).

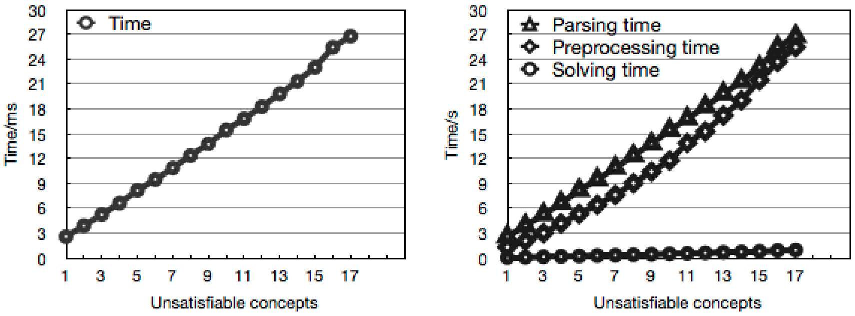

Fig. 2. Total axioms:42, unsatisfiable concepts from 1 to 18

Figure 2 shows the experimental results in the first set of tests. In each test, 42 axioms were randomly selected from Galen ontology and they were separated into two sub-ontologies for merging. Then assertions were inserted into one of them so that some concepts became unsatisfiable in the union of these two ontologies. The number of unsatisfiable concepts varied from 1 to 18. From this

[6] `http://www.qbflib.org/qdimacs.html`

[7] `http://logic.pdmi.ras.ru/~basolver/rtl.html`

[8] `http://www.co-ode.org/galen/`

figure we can see that the program is quite fast according to the time used for generating the QBF and the time used for deciding the validity of the QBF. This is because when the number of unsatisfiable concepts increases, the size of the QBF generated increases linearly. However, when the number of unsatisfiable concepts is over 18, OntoMerge can still generate the QBF but the QBF solver is unable to decide the validity of such a QBF.

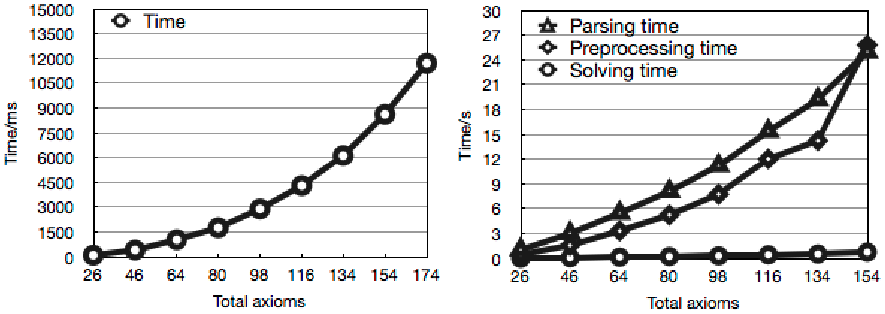

Fig. 3. Total number of unsatisfiable concept:1, total axioms from 26 to 174

In the second set of tests, we fixed the number of unsatisfiable concepts to 1 but increased the total number of axioms from 26 to 174. Figure 3 shows that when the number of axioms is increased, the time cost for generating QBF increases faster than in the first set of tests. This is partly due to the fact that with the increase of the total axioms, the size of the QBF significantly increases too. Similar to the case for the first set of tests, when the total number of axioms is over 174, the QBF solver fails again.

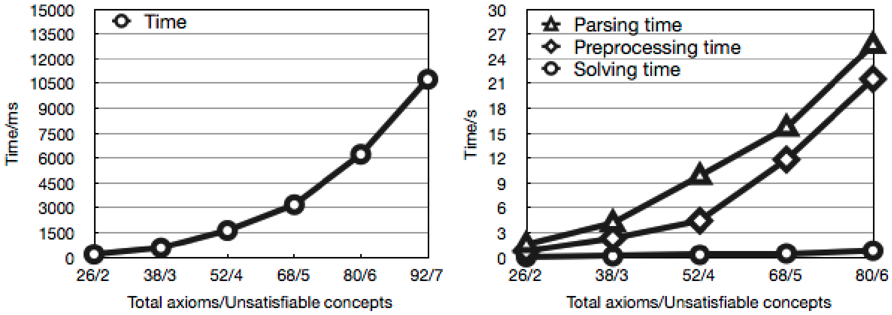

Fig. 4. Ratio of unsatisfiable concepts to total axioms = 0.15

In the third set of tests, we fix the ratio of total number of axioms to the number of unsatisfiable concepts to around 0.15 but let the total number of axioms varied from 26 to 92 as well as the number of unsatisfiable concepts. Figure 4 shows a similar pattern to Figure 3. When the total number of axioms is over 92, the QBF solver failed to return an answer.

7 Conclusion

We have developed a novel approach for merging ontologies in DL-Lite, in terms of types instead of classical DL models. We have also presented algorithms to reduce subclass queries of DL-Lite merging to the evaluation of QBFs and thus provided a novel way of reasoning with the result of ontology merging using efficient QBF solvers. We have implemented a preliminary merging system OntoMerge, and reported some experimental results in the paper. Currently we are extending the approach in two directions: (1) merging DL-Lite ontologies with both TBoxes and ABoxes, and (2) merging ontologies in expressive DLs.

Acknowledgement: We would like to thank the three anonymous referees for their helpful comments. Special thanks to Rodney Topor for various discussions with him. Kewen Wang was partially supported by the ARC grants DP1093652 and DP110101042. Guilin Qi was partially supported by the NSFC grant 61272378.

References

1. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *J. Artif. Intell. Res.*, 36:1–69, 2009.
2. D. Calvanese and G. De Giacomo. Data integration: A logic-based perspective. *AI Magazine*, 26(1):59–70, 2005.
3. J. P. Delgrande and T. Schaub. A consistency-based approach for belief change. *Artif. Intell.*, 151(1-2):1–41, 2003.
4. J. P. Delgrande, T. Schaub, H. Tompits, and S. Woltran. On computing belief change operations using quantified boolean formulas. *J. Log. Comput.*, 14(6):801–826, 2004.
5. J. Euzenat and P. Shvaiko. *Ontology matching.* Springer-Verlag, 2007.
6. P. Everaere, S. Konieczny, and P. Marquis. Conflict-based merging operators. In *Proc. KR*, 348–357, 2008.
7. I. Horrocks, B. Motik and Z. Wang. The HermiT OWL Reasoner. *Proc. IJCAR-ORE workshop*, 2012.
8. R. Goultiaeva, V. Iverson, and F. Bacchus. Beyond CNF: A circuitbased QBF solver. In *Proc. SAT*, 412–426, 2009.
9. A. Kalyanpur, B. Parsia, E. Sirin, and B. Cuenca Grau. Repairing unsatisfiable concepts in OWL ontologies. In *Proc. of ESWC*, pages 170–184, 2006.
10. A. Kalyanpur, B. Parsia, E. Sirin, and J. A. Hendler. Debugging unsatisfiable classes in owl ontologies. *J. Web Semantics*, 3(4):268–293, 2005.
11. S. Konieczny and R. Pino Pérez. Merging information under constraints: A logical framework. *J. Log. Comput.*, 12(5):773–808, 2002.
12. T. Meyer, K. Lee, and R. Booth. Knowledge integration for description logics. In *Proc. AAAI*, 645–650, 2005.
13. G. Qi and J. Du. Model-based revision operators for terminologies in description logics. In *Proc. IJCAI*, 891-897, 2009.
14. S. Schlobach, Z. Huang, R. Cornet, and F. van Harmelen. Debugging incoherent terminologies. *J. Autom. Reasoning*, 39(3):317–349, 2007.
15. Z. Wang, K. Wang, and R. Topor. A new approach to knowledge base revision in DL-Lite. In *Proc. AAAI*, 369–374, 2010.