

Instance-driven TBox Revision in DL-Lite

Zhe Wang¹, Kewen Wang¹, Guilin Qi², Zhiqiang Zhuang¹, and Yuefeng Li³

¹ School of ICT, Griffith University, Australia

² School of CSE, Southeast University, China

³ School of EECS, Queensland University of Technology, Australia

Abstract. The development and maintenance of large and complex ontologies are often time-consuming and error-prone. Thus, automated ontology learning and revision have attracted intensive research interest. In data-centric applications where ontologies are designed or automatically learnt from the data, when new data instances are added that contradict to the ontology, it is often desirable to incrementally revise the ontology according to the added data. This problem can be intuitively formulated as the problem of revising a TBox by an ABox. In this paper we introduce a model-theoretic approach to such an ontology revision problem by using a novel alternative semantic characterisation of DL-Lite ontologies. We show some desired properties for our ontology revision. We have also developed an algorithm for reasoning with the ontology revision without computing the revision result. The algorithm is efficient as its computational complexity is in coNP in the worst case and in PTIME when the size of the new data is bounded.

1 Introduction

A large number of professional and comprehensive ontologies have been developed for accessing data, including the SNOMED CT [23] and the NCI ontologies [9]. The Web Ontology Language (OWL), with its latest version, OWL 2, is accepted as the W3C Consortium (W3C) recommendation for ontology languages. OWL and OWL 2 are based on description logics (DLs) [2]. Three profiles of OWL 2 are proposed, namely EL, QL and RL, which are based on fine-tuned DLs for efficient reasoning.

Existing large ontologies like SNOMED CT are mostly hand-crafted by human experts. The development and maintenance of such large ontologies are often time-consuming and error-prone. Recently, automated ontology learning [3, 15, 13] and revision [19, 18, 20, 17, 25, 26] have attracted intensive interest. Although it is often conceived that ontologies, once established, are more stable and reliable than data, automated ontology crafting and revision are especially useful in data-centric applications and for pre-mature ontologies. A typical scenario is ontology learning, where terminological relations are learnt from example data provided by domain experts, which in DL terms is learning TBox axioms from ABox assertions. Several TBox learning approaches have been proposed in the literature and effective learning tools have been developed [15]. An obvious requirement in ontology learning is that the learnt ontologies must be consistent with the example data. However, when new data instances become available, it is possible that the new data may contradict some previously learnt axioms.

Obviously, when the size(s) of the ontology and/or the data is large, it is undesirable to re-learn the ontology from the scratch. Thus, a challenge is how to revise the ontology according to the new data. In DLs, it amounts to the question of how to effectively revise a TBox by an ABox.

TBox debugging and repair approaches [11, 22, 10] and interactive revision approaches [16] can be used to pinpoint the problematic axioms, and most of TBox repair approaches seek to eliminate a minimal number of axioms to restore consistency. Similar syntax-based approach was adopted in revision [8, 18, 20]. However, most syntax-based approaches to TBox revision lack of a suitable semantic justification, which is partially reflected in their inability of preserving implicit knowledge. For example, given $\mathcal{T} = \{\text{Bird} \sqsubseteq \text{CanFly}, \text{Mammal} \sqsubseteq \neg\text{CanFly}\}$, the revision of \mathcal{T} by $\mathcal{A} = \{\text{CanFly}(\text{bat}), \text{Mammal}(\text{bat})\}$ using syntax-based approaches in [8, 20] will discard implicit information $\text{Bird} \sqsubseteq \neg\text{Mammal}$, though it does not involve in the contradiction. To resolve this issue, some more recent works, such as [5, 6], advocate to define syntax-based ontology revision on the deductive closure $cl(\mathcal{T})$ of the initial TBox \mathcal{T} instead of \mathcal{T} itself. This of course relies on the finiteness of the deductive closure (with the exception of some special cases), and there may not exist a unique optimal solution to syntax-based revision [5], since often multiple minimal sets of axioms in \mathcal{T} (or $cl(\mathcal{T})$) exist that are responsible for the contradictions.

Seeing the TBox as a formalisation of the conceptual model of the data, which is a common perspective in ontology learning, the TBox essentially represents a collection of possible DL models. Taking this perspective, one can argue that in TBox revision, it is critical to ensure minimal changes to the prospective models. Extant model-based TBox revision literature only cover the revision by TBox axioms [17, 26]. In this paper, we develop a novel approach to revise a TBox by ABox assertions, and to full-fill the minimal change principle in a model-theoretic manner. To the best of our knowledge, it is the first work on this direction. However, as witnessed by previous model-based TBox revision approaches, working directly with classical DL models is difficult. Thus, we introduce the type semantics for DL-Lite_{core}, which is the core logic underlying OWL 2 QL. We provide a concrete TBox revision operator based on types. We show that our operator possess desired properties. In particular, our operator supports revision of axioms, and the result of revision is uniquely defined which eliminates non-determinism. For the computational aspects, we transform the entailment problem of TBox revision in DL-Lite into one in propositional logic. We show that entailment checking and revision computation can be done efficiently, and in particular, do not shift the corresponding complexity in propositional logic.

2 Preliminaries

A *signature* \mathcal{S} is a union of three disjoint (possibly infinite) sets \mathcal{S}_C , \mathcal{S}_R , and \mathcal{S}_I , where \mathcal{S}_C is the set of atomic concepts, \mathcal{S}_R is the set of atomic roles, and \mathcal{S}_I is the set of individual names. Concepts and roles in DL-Lite_{core} and DL-Lite_R are built on atomic ones as follows [4]:

$$B \rightarrow A \mid \exists R \qquad C \rightarrow B \mid \neg B \mid \perp \qquad R \rightarrow P \mid P^-$$

where $A \in \mathcal{S}_C$ and $P \in \mathcal{S}_R$. B is a *basic concept*, C is a *general concept*, and R is a *basic role*. In what follows, we often use A for an atomic concept, B for a basic concept, C for a general concepts, P for an atomic role, and R for a basic role. \mathcal{B} and \mathcal{R} denotes the sets of basic concepts and basic roles over \mathcal{S} , respectively. We write R^- for P if $R = P^-$; and R^+ for P , whenever $R = P$ or $R = P^-$. A DL-Lite_{core} TBox is a finite set of *concept inclusions* or *axiom* of the form $B \sqsubseteq C$. An ABox is a finite set of *concept assertions* of the form $A(a)$ and *role assertions* of the form $P(a, b)$, where $a, b \in \mathcal{S}_I$. For the convenience of presentation, we sometimes write $P^-(a, b) \in \mathcal{A}$ meaning $P(b, a) \in \mathcal{A}$. A knowledge base (KB) $\mathcal{K} = \mathcal{T} \cup \mathcal{A}$ consists of a TBox \mathcal{T} and an ABox \mathcal{A} . In this paper, an ontology is a DL-Lite TBox.

The semantics of DL-Lite_{core} is defined as usual using interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, and we refer to [4] for the details. An interpretation \mathcal{I} *satisfies* inclusion $B \sqsubseteq C$ if $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$; \mathcal{I} *satisfies* assertion $A(a)$ if $a^{\mathcal{I}} \in A^{\mathcal{I}}$; and \mathcal{I} *satisfies* assertion $P(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$. \mathcal{I} *satisfies* TBox \mathcal{T} (or ABox \mathcal{A}) if \mathcal{I} satisfies each inclusion in \mathcal{T} (resp., each assertion in \mathcal{A}). \mathcal{I} is a *model* of \mathcal{T} (\mathcal{A} , or $\mathcal{T} \cup \mathcal{A}$) if \mathcal{I} satisfies \mathcal{T} (resp., \mathcal{A} , or both \mathcal{T} and \mathcal{A}). $\mathcal{T} \cup \mathcal{A}$ is *consistent* if it has at least one model. We also say that \mathcal{T} is consistent with \mathcal{A} . A concept or role E is *satisfiable w.r.t. \mathcal{T}* if a model \mathcal{I} of \mathcal{T} exists such that $E^{\mathcal{I}} \neq \emptyset$; otherwise, E is *unsatisfiable*. \mathcal{T} is *coherent* if all atomic concepts and atomic roles in \mathcal{T} are satisfiable. $\mathcal{T} \cup \mathcal{A}$ *entails* an inclusion or assertion α , written $\mathcal{T} \cup \mathcal{A} \models \alpha$, if all models of $\mathcal{T} \cup \mathcal{A}$ satisfy α . Two KBs $\mathcal{K}_1, \mathcal{K}_2$ are *equivalent*, written $\mathcal{K}_1 \equiv \mathcal{K}_2$, if they have the same models.

3 Type Semantics

To revise a TBox \mathcal{T} by an ABox \mathcal{A} , the result of revision should be a new TBox \mathcal{T}' , such that \mathcal{T}' is consistent with \mathcal{A} and that \mathcal{T}' preserves as much terminological knowledge as possible from \mathcal{T} . We address TBox revision and minimal change from a model-theoretic perspective, that is, the models of the result of revision can be obtained from the models of the initial TBox with minimal change. To achieve this, we adapt classical model-based belief revision to DL-Lite, which measures change through a notion of model distance. However, DL models may have complex structures and the number of models is always infinite. This makes it hard to handle TBox revision directly through classical DL models. Hence, we introduce a new relation of satisfaction for DL-Lite_{core} TBoxes based on the notion of types [14]. Such an alternative semantic characterisation use structures that are finite and much simpler than classical DL models.

In TBox revision, it suffices to consider a finite signature \mathcal{S} that covers the initial TBox and the new data. In this case, \mathcal{B} and \mathcal{R} are both finite. Formally, a *type* $\tau \subseteq \mathcal{B}$ is a (possibly empty) set of basic concepts. The satisfaction relation between a type τ and a TBox \mathcal{T} (or concept C) can be defined in the propositional manner, seeing τ as a propositional interpretation (with basic concepts seen as propositional atoms) and \mathcal{T} (or C) as a propositional theory. $PM(\mathcal{T})$ ($PM(C)$) denotes the set of types corresponding to the propositional models of \mathcal{T} (C). When $\mathcal{T} = \{B \sqsubseteq C\}$ is a singleton, we also write $PM(B \sqsubseteq C)$. However, classical relation of satisfaction is insufficient and we need to introduce a refined definition of satisfaction. For example, let $\mathcal{T} = \{\exists P \sqsubseteq \perp\}$, then $\mathcal{T} \models \exists P^- \sqsubseteq \perp$. Yet $PM(\mathcal{T}) \not\subseteq PM(\exists P^- \sqsubseteq \perp)$, witnessed by the type $\tau = \{\exists P\}$.

For a well defined semantics, one would expect that the models of \mathcal{T} are also models of $\exists P^- \sqsubseteq \perp$. Hence, the refined satisfaction relation needs to take care of the non-propositional inference that $\exists R \sqsubseteq \perp$ implies $\exists R^- \sqsubseteq \perp$ for any $R \in \mathcal{R}$, and we will show that this is the only non-propositional inference needing special care in DL-Lite_{core}. In [14, 25], this problem is remedied using a set of types (instead of a single type) in a model. However, the cost of using arbitrary sets of types as in [25] is an exponential blow up on the sizes of models. Although simpler (in particular polynomial-sized) models are defined in [14], the model structures are still rather complex. In what follows, we present a more succinct characterisation, the type semantics, and we will extend the type semantics to ABoxes.

Definition 1. *A type τ satisfies a DL-Lite_{core} TBox \mathcal{T} if the following conditions hold:*

1. $\tau \in PM(\mathcal{T})$.
2. if $\mathcal{T} \models \exists R \sqsubseteq \perp$ then $\exists R^- \notin \tau$.

In this case, τ is a type model (T-model) of \mathcal{T} . The set of T-models of \mathcal{T} is denoted as $TM(\mathcal{T})$.

Again, we write $TM(B \sqsubseteq C)$ for a singleton TBox $\mathcal{T} = \{B \sqsubseteq C\}$.

T-models can be used to characterise the semantics of DL-Lite_{core} TBoxes, and major reasoning problems such as concept satisfiability and subsumption *w.r.t.* TBoxes can be characterised using T-models.

Proposition 1. *Given a DL-Lite_{core} TBox \mathcal{T} , the following results hold:*

1. \mathcal{T} is consistent iff there exists a T-model of \mathcal{T} .
2. For a concept C , C is satisfiable *w.r.t.* \mathcal{T} iff $TM(\mathcal{T}) \cap PM(C) \neq \emptyset$.
3. For an inclusion $B \sqsubseteq C$, $\mathcal{T} \models B \sqsubseteq C$ iff $TM(\mathcal{T}) \subseteq TM(B \sqsubseteq C)$.
4. For a DL-Lite_{core} TBox \mathcal{T}' , $\mathcal{T} \equiv \mathcal{T}'$ iff $TM(\mathcal{T}) = TM(\mathcal{T}')$.

To prove this proposition, we first show a connection between the classical DL models and the type models of a TBox. Each DL interpretation \mathcal{I} and each element d in the domain of \mathcal{I} determines a unique type, *i.e.*, $\tau_d^{\mathcal{I}} = \{B \in \mathcal{B} \mid d \in B^{\mathcal{I}}\}$. We call $\tau_d^{\mathcal{I}}$ the type of d in \mathcal{I} .

Lemma 1. *Given a DL-Lite_{core} TBox \mathcal{T} and a DL interpretation \mathcal{I} , \mathcal{I} is a model of \mathcal{T} iff $\tau_d^{\mathcal{I}} \in TM(\mathcal{T})$ for each $d \in \Delta^{\mathcal{I}}$.*

Lemma 1 shows for each DL interpretation \mathcal{I} , a set M of types can be determined by \mathcal{I} , and \mathcal{I} is a model of some DL-Lite_{core} TBox if and only if each element of M is a T-model of the TBox. The following lemma shows the converse is also true—given a T-model τ of a TBox \mathcal{T} , a DL model of \mathcal{T} can be constructed from τ .

Lemma 2. *Given a DL-Lite_{core} TBox \mathcal{T} and a T-model τ of \mathcal{T} , there exists a DL model \mathcal{I} of \mathcal{T} and a domain element $d \in \Delta^{\mathcal{I}}$ such that $\tau_d^{\mathcal{I}} = \tau$.*

From Lemmas 1 and 2, we can prove Proposition 1.

Proposition 1 shows that T-models capture the semantics of DL-Lite_{core} TBoxes. In order to define the revision of a TBox by an ABox, our type semantics needs to define also (partially) the interpretations on the ABox, that is, we need to define which types constitute to the rational interpretations of the ABox. While type semantics cannot completely capture the semantics of ABoxes, due to the fact that types do not refer to individual names, we can define the “type-models” of an ABox \mathcal{A} as those types determined by the models of \mathcal{A} .

As in the definition of interpretation-determined types, where each element in an interpretation uniquely determines a type, each named individual b in \mathcal{A} and each interpretation \mathcal{I} of \mathcal{A} uniquely determines a type $\tau_b^{\mathcal{I}} = \{B \in \mathcal{B} \mid b^{\mathcal{I}} \in B^{\mathcal{I}}\}$. As there are often multiple models of \mathcal{A} , b is associated with multiple types in \mathcal{A} . Formally, type τ is an *instance type of b* (or a *b -type*) in \mathcal{A} iff it is the type of $b^{\mathcal{I}}$ in some model \mathcal{I} of \mathcal{A} . Intuitively, a b -type in \mathcal{A} is the set of basic concepts to whom b is interpreted as a member by some model of \mathcal{A} . For example, given $\mathcal{A} = \{A(b), P(b, c)\}$, type $\tau = \{A, \exists P\}$ is a b -type but not a c -type in \mathcal{A} . This is because there exists a model \mathcal{I} of \mathcal{A} (e.g., the model obtained by seeing \mathcal{A} as a complete interpretation) such that $\tau_b^{\mathcal{I}} = \tau$; and for each model \mathcal{I} of \mathcal{A} , $c^{\mathcal{I}} \in (\exists P^-)^{\mathcal{I}}$ but $\exists P^- \notin \tau$. Let $TM(b, \mathcal{A})$ be the set consisting of all the b -types in \mathcal{A} . $TM(\mathcal{A})$ is the set of instance types in \mathcal{A} , i.e., $TM(\mathcal{A}) = \bigcup_{b \text{ occurs in } \mathcal{A}} TM(b, \mathcal{A})$.

The following proposition says that a TBox \mathcal{T} is consistent with an ABox \mathcal{A} iff for each individual b occurring in \mathcal{A} , there exists a T-model of \mathcal{T} that is a b -type.

Proposition 2. *Given a TBox \mathcal{T} and an ABox \mathcal{A} in DL-Lite_{core}, $\mathcal{T} \cup \mathcal{A}$ is consistent iff $TM(\mathcal{T}) \cap TM(b, \mathcal{A}) \neq \emptyset$ for each individual b occurring in \mathcal{A} .*

Without conjunction and disjunction, DL-Lite_{core} cannot capture the full propositional logic. As a result, given a set M of types, M is not always expressible by DL-Lite_{core} TBoxes, i.e., M may not be the set of T-models of any DL-Lite_{core} TBox. Also, the set of T-models of a DL-Lite_{core} TBox \mathcal{T} needs to be role balanced. Formally, a set of types is called *role balanced* if it includes a type containing $\exists R$ whenever it includes a type containing $\exists R^-$ for all $R \in \mathcal{R}$.

Proposition 3. *$TM(\mathcal{T})$ is role balanced for each DL-Lite_{core} TBox \mathcal{T} .*

For a given set M of types, we define a corresponding TBox for M to be one that has the minimal set of T-models that includes M . Formally, a DL-Lite_{core} TBox \mathcal{T} is a *corresponding TBox* to M if $M \subseteq TM(\mathcal{T})$ and there is no DL-Lite_{core} TBox \mathcal{T}' such that $M \subseteq TM(\mathcal{T}') \subset TM(\mathcal{T})$. $\mathcal{T}_{core}(M)$ denotes a corresponding TBox for M . The corresponding TBoxes are not unique in general, as intuitively often multiple role-balanced minimal supersets exist. For example, suppose M consists of only one type $\tau = \{\exists P\}$. There are two corresponding TBoxes for M that are $\{\exists P^- \sqsubseteq \exists P\}$ and $\{\exists P^- \sqsubseteq \neg \exists P\}$, with the T-models (other than τ) $\{\exists P, \exists P^-\}$ and $\{\exists P^-\}$, respectively. Intuitively, Definition 1 enforces that a corresponding TBox \mathcal{T} of M cannot infer $\exists P^- \sqsubseteq \perp$, and a DL model of \mathcal{T} must instantiate $\exists P^-$ with at least one member e . The above two TBoxes correspond to the cases where e is and is not a member of $\exists P$, respectively. The following proposition shows that role-balance guarantees the uniqueness.

Proposition 4. *Each role balanced set of types has a unique corresponding TBox (up to semantic equivalence).*

When a unique corresponding TBox exists, the TBox has the same set of logical consequences in $\text{DL-Lite}_{\text{core}}$ as the set of types.

Lemma 3. *Let M be a role balanced set of types and \mathcal{T} be its unique corresponding TBox. For each inclusion $B \sqsubseteq C$, $M \subseteq \text{PM}(B \sqsubseteq C)$ iff $\mathcal{T} \models B \sqsubseteq C$.*

4 TBox Revision

In this section, we use our new semantic characterisation to define a specific TBox revision operator. In classical model-based revision, to revise a propositional knowledge base ϕ by a propositional formula ψ , it is sufficient to select the models of ψ that have minimal distances to those of ϕ , and to define the propositional knowledge base corresponding to the selected models to be the result of revision. Inspired by classical model-based revision, we first introduce a notion of distance between T-models, and then define a selection function that selects from a set of types those types having minimal distances to the T-models of the initial TBox. As we will see later, however, a simple adoption of the classical approach to DL-Lite is insufficient to ensure the result of revision has desired properties. We present our novel revision approach tailored for instance-driven TBox revision applications.

We adopt a prominent notion of model distance in classical model-based revision introduced by Satoh [21], where the distance between two propositional models (seen as sets of propositional atoms) is defined as the symmetric difference between the two models. The *distance* between two types τ and τ' is their symmetric difference $sd(\tau, \tau') = (\tau \setminus \tau') \cup (\tau' \setminus \tau)$. Intuitively, a pair of types τ_1, τ'_1 are considered to be (strictly) closer than another pair τ_2, τ'_2 if $sd(\tau_1, \tau'_1) \subset sd(\tau_2, \tau'_2)$. While we adopt Satoh's distance, we note that other notions of distance can also be applied in our revision framework. Further, given two sets M and M' of types, a selection function γ selects the types in M that are closest to those in M' , based on the notion of distance. In particular,

$$\gamma(M, M') = \{\tau_0 \in M \mid \tau'_0 \in M' \text{ exists s.t.} \\ \text{for all } \tau \in M \text{ and } \tau' \in M', sd(\tau, \tau') \not\subset sd(\tau_0, \tau'_0)\}$$

For a TBox \mathcal{T} and an ABox \mathcal{A} , we could define the revision of \mathcal{T} by \mathcal{A} to be a TBox corresponding to $\gamma(TM(\mathcal{A}), TM(\mathcal{T}))$, as a straightforward adoption of Satoh's model-based revision to DL-Lite. However, a result of revision defined in this manner may not be coherent, and moreover, the result of revision is not guaranteed to be consistent with the ABox \mathcal{A} . For example, let $\mathcal{T} = \{A \sqsubseteq B, A \sqsubseteq \neg B\}$ and $\mathcal{A} = \{A(b), B(c)\}$. Then, $\gamma(TM(\mathcal{A}), TM(\mathcal{T})) = TM(\mathcal{A}) \cap TM(\mathcal{T})$ consists of a single type $\{B\}$. The corresponding TBox $\{A \sqsubseteq \perp\}$ is incoherent and is inconsistent with \mathcal{A} . Note that from the above example, to achieve coherence and to ensure consistency with the ABox are not necessarily two separate goals. That is, in many cases like \mathcal{T} , the TBox is inconsistent with the ABox because it is incoherent. Hence, if the inconsistency is caused by an unsatisfiable concept or role, then both consistency between the TBox and the ABox and

the coherence of the TBox can be recovered simultaneously by revising only the TBox (without changing the ABox). To this end, we introduce the *coherence closure* \mathcal{A}^* of an ABox \mathcal{A} , which extends \mathcal{A} with $A(c_A)$ for each $A \in \mathcal{S}_C$ such that no assertion $A(a)$ occurs in \mathcal{A} , where c_A is a fresh individual name not occurring in \mathcal{A} , and extends \mathcal{A} with $P(c_P, d_P)$ for each $P \in \mathcal{S}_R$ such that no assertion $P(a, b)$ occurs in \mathcal{A} , where c_P, d_P are fresh individual names not occurring in \mathcal{A} . In the above example, $\mathcal{A}^* = \mathcal{A}$.

Lemma 4. *For a TBox \mathcal{T} and an ABox \mathcal{A} , \mathcal{T} is coherent and $\mathcal{T} \cup \mathcal{A}$ is consistent iff $\mathcal{T} \cup \mathcal{A}^*$ is consistent.*

The lemma show that the task of maintaining both the consistency of KB $\mathcal{T} \cup \mathcal{A}$ and the coherence of TBox \mathcal{T} can be reduced to that of maintaining only the consistency of KB $\mathcal{T} \cup \mathcal{A}^*$.

If $\mathcal{T} \circ \mathcal{A}$ is the result of revising \mathcal{T} by \mathcal{A} , by Proposition 2, the consistency of KB $(\mathcal{T} \circ \mathcal{A}) \cup \mathcal{A}^*$ can be guaranteed by the condition $TM(\mathcal{T} \circ \mathcal{A}) \cap TM(b, \mathcal{A}^*) \neq \emptyset$ for each b occurring in \mathcal{A}^* , i.e., the set of T-models of $\mathcal{T} \circ \mathcal{A}$ must contain at least one b -type for each individual b in \mathcal{A}^* . It is clear that $\gamma(TM(\mathcal{A}^*), TM(\mathcal{T}))$ does not necessarily satisfy this condition, as can be seen from the above example: There are two b -types in \mathcal{A}^* , $\{A, B\}$ and $\{A\}$, and $\gamma(TM(\mathcal{A}^*), TM(\mathcal{T})) = \gamma(TM(\mathcal{A}), TM(\mathcal{T}))$ does not contain either of them. Intuitively, for $\mathcal{T} \circ \mathcal{A}$ to be consistent with \mathcal{A}^* , the T-models of $\mathcal{T} \circ \mathcal{A}$ needs to contain at least one of the b -types. In the example, both b -types have the same distance to the T-models of \mathcal{T} (types $\{B\}$ and \emptyset) and both b -types should be selected as T-models of $\mathcal{T} \circ \mathcal{A}$. In general, for each individual b occurring in \mathcal{A}^* , the b -types in \mathcal{A}^* that are closest to $TM(\mathcal{T})$ should be selected. Also, when \mathcal{T} is already coherent and is consistent with \mathcal{A} , the revision operation should not change \mathcal{T} . To this end, all the initial T-models of \mathcal{T} are also added as T-models of $\mathcal{T} \circ \mathcal{A}$.

Based on the above intuitions, we define our TBox revision operator as follows.

Definition 2. *Let \mathcal{T} be a TBox and \mathcal{A} be an ABox in $DL\text{-Lite}_{core}$. The revision of \mathcal{T} by \mathcal{A} is $\mathcal{T} \circ \mathcal{A} = \mathcal{T}_{core}(TM(\mathcal{T}) \cup M_{\mathcal{A}^*})$ where*

$$M_{\mathcal{A}^*} = \bigcup_{b \text{ occurs in } \mathcal{A}^*} \gamma(TM(b, \mathcal{A}^*), TM(\mathcal{T})).$$

Intuitively, $M_{\mathcal{A}^*}$ is constituted by, for each individual b occurring in \mathcal{A}^* , a set of b -types that are closest to the T-models of \mathcal{T} . $M_{\mathcal{A}^*}$ is constructed as such in order that the result of revision is consistent with \mathcal{A}^* ; and the result of revising \mathcal{T} by \mathcal{A} is defined to be a corresponding TBox of $TM(\mathcal{T}) \cup M_{\mathcal{A}^*}$. It is not hard to see that $TM(\mathcal{T}) \cup M_{\mathcal{A}^*}$ is role balanced: For each $P \in \mathcal{S}_R$, assertion $P(b, c)$ occurs in \mathcal{A}^* for some $b, c \in \mathcal{S}_I$, and as a result, a b -type containing $\exists P$ and a c -type containing $\exists P^-$ are contained in $M_{\mathcal{A}^*}$. From Proposition 4, $\mathcal{T} \circ \mathcal{A}$ is uniquely defined (up to semantic equivalence).

Consider the following example adapted from the NCI ontology [9].

Example 1. Let TBox \mathcal{T} consists of the following inclusions

$$\text{Heart_Disease} \sqsubseteq \exists \text{has_Site}, \quad (1)$$

$$\exists \text{has_Site}^- \sqsubseteq \text{Cardiovascular_System}, \quad (2)$$

$$\exists \text{has_Site}^- \sqsubseteq \text{Respiratory_System}, \quad (3)$$

$$\text{Respiratory_System} \sqsubseteq \text{Organ_System}, \quad (4)$$

$$\text{Cardiovascular_System} \sqsubseteq \text{Organ_System}, \quad (5)$$

$$\text{Respiratory_System} \sqsubseteq \neg \text{Cardiovascular_System}, \quad (6)$$

$$\text{Heart_Disease} \sqsubseteq \neg \text{Organ_System}. \quad (7)$$

The following two inclusions in the original NCI ontology are not DL-Lite_{core} axioms and thus they are replaced with inclusions (1)–(3). While this is not a precise translation, it is sufficient to demonstrate the application of our new revision in handling such medical ontologies.

$$\text{Heart_Disease} \sqsubseteq \exists \text{has_Site}.\text{Cardiovascular_System},$$

$$\text{Heart_Disease} \sqsubseteq \exists \text{has_Site}.\text{Respiratory_System}.$$

The TBox \mathcal{T} is incoherent, as both concept Heart_Disease and role has_Site are unsatisfiable.

Given $\mathcal{A} = \{\text{Heart_Disease}(\text{hd1}), \text{has_Site}(\text{hd1}, \text{s1}), \text{Organ_System}(\text{s1})\}$. Then, by Definition 2, $\mathcal{T} \circ \mathcal{A}$ consists of inclusions (1), (4)–(7), and the following inclusion

$$\exists \text{has_Site}^- \sqsubseteq \text{Organ_System}. \quad (8)$$

The inclusion (8) can be seen as a revision from axioms (2)–(5).

The example shows that our TBox revision operator allows modification of axioms, in contrast to most existing TBox repair approaches that only support axiom removal.

Furthermore, the revision operator satisfies the following desired properties presented as AGM-style postulates [12]. Note that our revision operator is different from the classical AGM ones in nature. The classical AGM approach concerns revision between propositional formulas of the same type, whereas our revision approach addresses revising TBox axioms by ABox assertions, which is between different types of first-order formulas. As a result, the following postulates are specially tailored for instance-driven TBox revision.

Proposition 5. *Let \mathcal{T} be a TBox and \mathcal{A} be an ABox in DL-Lite_{core}. The following properties hold.*

(R1) $\mathcal{T} \circ \mathcal{A}$ is coherent.

(R2) $(\mathcal{T} \circ \mathcal{A}) \cup \mathcal{A}$ is consistent.

(R3) If \mathcal{T} is coherent and $\mathcal{T} \cup \mathcal{A}$ is consistent, then $\mathcal{T} \circ \mathcal{A} \equiv \mathcal{T}$.

(R4) If $\mathcal{T}_1 \equiv \mathcal{T}_2$ and $\mathcal{A}_1 \equiv \mathcal{A}_2$, then $\mathcal{T}_1 \circ \mathcal{A}_1 \equiv \mathcal{T}_2 \circ \mathcal{A}_2$.

The first two properties (R1) and (R2) show that the result of revision is coherent and is consistent with the new ABox as required. Then, (R3) says that no revision is needed if the initial TBox is already coherent and consistent with the new ABox. Finally, (R4) shows that the revision operator is syntax-independent, *i.e.*, revising equivalent TBoxes with equivalent new ABoxes results in equivalent revision.

5 Computational Aspects

In this section, we first look into the computational complexity of the *entailment problem* for revision, *i.e.*, the problem of deciding whether $\mathcal{T} \circ \mathcal{A}$ entails a given TBox axiom $B \sqsubseteq C$. We show that the entailment can be checked without first computing $\mathcal{T} \circ \mathcal{A}$, and is achieved through a reduction to propositional logic. After that, we will provide an algorithm to compute $\mathcal{T} \circ \mathcal{A}$ through entailment checking. We show that when the size of \mathcal{A} is bounded, the computation can be done in polynomial time.

We first show that the entailment problem in DL-Lite can be reduced to one in propositional logic. To this end, we introduce a propositional transformation for DL-Lite TBoxes. This allows us to build essentially a connection between our revision operator and propositional belief revision, and to make use of the existing belief revision results and algorithms in propositional logics, *e.g.*, [7]. The transformation is inspired by [1], which introduces a transformation from a DL-Lite TBox into a theory in the one variable fragment of first order logic. Using type semantics instead of classical DL models allows us to obtain a novel transformation to propositional logic, which is much simpler than theirs.

To achieve this, each basic concept in \mathcal{B} is assigned a distinct propositional atom, each A to p_A and each $\exists R$ to $p_{\exists R}$. Function $\phi(\cdot)$ maps each basic concept in \mathcal{B} to its corresponding propositional atom, and a general concept to a propositional literal as follows.

$$\phi(\perp) = \perp, \quad \phi(A) = p_A, \quad \phi(\exists R) = p_{\exists R}, \quad \phi(\neg B) = \neg\phi(B).$$

Then, $\phi(\cdot)$ can be extended to map a DL-Lite_{core} TBox \mathcal{T} to a propositional formula as follows

$$\phi(\mathcal{T}) = \bigwedge_{B \sqsubseteq C \in \mathcal{T}} (\phi(B) \rightarrow \phi(C)) \wedge \bigwedge_{R \in \mathcal{R}, \mathcal{T} \models \exists R \sqsubseteq \perp} (\neg\phi(\exists R) \wedge \neg\phi(\exists R^-)).$$

Further, to encode the instance types of an ABox, a pair of a DL-Lite_{core} ABox \mathcal{A} and an individual name b occurring in \mathcal{A} is mapped to a propositional formula as follows.

$$\phi(b, \mathcal{A}) = \bigwedge_{A(b) \in \mathcal{A}} \phi(A) \wedge \bigwedge_{P(b,c) \in \mathcal{A}} \phi(\exists P) \wedge \bigwedge_{P(c,b) \in \mathcal{A}} \phi(\exists P^-)$$

Note that the mapping does not translate an ABox into a propositional formula. Instead, formula $\phi(b, \mathcal{A})$ is used to capture the b -types in \mathcal{A} . Intuitively, the models of $\phi(\mathcal{T})$ correspond to the T-models of \mathcal{T} , and the models of $\phi(b, \mathcal{A})$ correspond to the b -types in \mathcal{A} , as shown in the following lemma. Let $\omega_\tau = \{\phi(B) \mid B \in \tau\}$ for a type τ . $\text{mod}(\phi)$ denotes the set of models of propositional formula ϕ .

Lemma 5. *Let \mathcal{T} be a TBox and \mathcal{A} be an ABox in DL-Lite_{core}. Then,*

1. $\text{mod}(\phi(\mathcal{T})) = \{\omega_\tau \mid \tau \in TM(\mathcal{T})\}$;
2. $\text{mod}(\phi(b, \mathcal{A})) = \{\omega_\tau \mid \tau \in TM(b, \mathcal{A})\}$.

Let \circ_s be Satoh's revision operator in propositional logic. The following connection between our revision operator and Satoh's holds.

Proposition 6. *Let \mathcal{T} be a TBox and \mathcal{A} be an ABox in DL-Lite_{core}. Given a TBox axiom $B \sqsubseteq C$, $\mathcal{T} \circ \mathcal{A} \models B \sqsubseteq C$ iff $\phi(\mathcal{T}) \vee \bigvee_{b \text{ occurs in } \mathcal{A}^*} (\phi(\mathcal{T}) \circ_s \phi(b, \mathcal{A}^*)) \vdash \phi(B) \rightarrow \phi(C)$.*

Proposition 6 allows us to reduce the entailment problem in DL-Lite_{core} to a corresponding problem in propositional logic, and thus to make use of existing complexity results for propositional revision. Note that $\phi(\mathcal{T})$ and $\phi(b, \mathcal{A})$ are both Horn propositional formulas, and the sizes of them are polynomial in the size of $\mathcal{T} \cup \mathcal{A}$. From the complexity results in [7], it is easy to conclude the following complexity results. In particular, our TBox revision operator for DL-Lite_{core} does not shift the complexity of the propositional case.

Theorem 1. *For a TBox \mathcal{T} , an ABox \mathcal{A} , and a TBox axiom $B \sqsubseteq C$ in DL-Lite_{core}, deciding $\mathcal{T} \circ \mathcal{A} \models B \sqsubseteq C$ is in coNP.*

If the size of \mathcal{A} is bounded, i.e., a constant k exists s.t. $|\mathcal{A}| \leq k$, then deciding $\mathcal{T} \circ \mathcal{A} \models B \sqsubseteq C$ is in PTIME.

The coNP upper bound is not necessarily tight, but mainly shows the connection of our revision operator with the propositional ones. We leave a tight complexity result for the entailment problem in DL-Lite_{core} (with inputs of unbounded sizes) open. Having a bound on the size of \mathcal{A} and the PTIME complexity are practically relevant, as in ontology learning, the newly added example data is often small. The proof is not straightforward as $|\mathcal{A}|$ being bounded does not necessarily implies that $|\mathcal{A}^*|$ is bounded. Indeed, the PTIME complexity result holds as long as the number of assertions about each individual in \mathcal{A} is bounded.

Algorithm 1: Revision

Input: TBox \mathcal{T} and ABox \mathcal{A} in DL-Lite_{core}
Output: a DL-Lite_{core} TBox \mathcal{T}'
 $\mathcal{T}' := \emptyset$;
foreach $B_1, B_2 \in \mathcal{B}$ s.t. $B_1 \neq B_2$ **do**
 if $\mathcal{T} \circ \mathcal{A} \models B_1 \sqsubseteq B_2$ **then**
 $\mathcal{T}' := \mathcal{T}' \cup \{B_1 \sqsubseteq B_2\}$;
 end
 if $\mathcal{T} \circ \mathcal{A} \models B_1 \sqsubseteq \neg B_2$ **then**
 $\mathcal{T}' := \mathcal{T}' \cup \{B_1 \sqsubseteq \neg B_2\}$;
 end
end
return \mathcal{T}' ;

We provide an algorithm to compute $\mathcal{T} \circ \mathcal{A}$ using entailment checking as a sub-method. The algorithm verifies each possible entailment and add those entailed axioms to the result.

Proposition 7. *Let \mathcal{T} be a TBox, \mathcal{A} be an ABox in $DL\text{-Lite}_{core}$, and \mathcal{T}' be the TBox returned by the algorithm Revision. Then, $\mathcal{T}' \equiv \mathcal{T} \circ \mathcal{A}$.*

Although DL-Lite ontologies are often associated with large ABoxes, the input ABox \mathcal{A} consists of only updated data and thus can be expected to be small. If the size of \mathcal{A} is bounded, the algorithm Revision runs in PTIME in the size of $\mathcal{T} \cup \mathcal{A}$. In particular, the number of possible inclusions in $DL\text{-Lite}_{core}$ is quadratic to the number of basic concepts in $\mathcal{T} \cup \mathcal{A}$, and the entailment check is in PTIME as discussed.

6 Conclusion

We have presented a novel approach to revise a DL-Lite TBox according to newly added ABox instances, and the revision incurs only minimal change to the initial TBox measured in a model-theoretic manner. To the best of our knowledge, our work is the first attempt to address particularly the revision of a TBox by ABox assertions. To tackle this problem, we have introduced the type semantics for $DL\text{-Lite}_{core}$ TBoxes, which is superior over the classical DL semantics in that each TBox has a finite number of type models and the new semantics still preserves most of major reasoning forms for DLs. Based on this new semantics, we have developed an operator for revising a TBox by an ABox. We show that the proposed revision operator possesses several desired properties and have also developed an efficient algorithm for computing the revision.

We are currently working on extending our approach to $DL\text{-Lite}_{\mathcal{R}}$. To address this issue, a key task is to extend the type semantics to $DL\text{-Lite}_{\mathcal{R}}$, which needs to take into account the additional non-propositional inferences introduced by role inclusions. It would be also interesting to look into practical applications of our revision approach in ontology learning.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyashev, M.: The DL-Lite family and relations. *Journal of Artificial Intelligence Research* 36, 1–69 (2009)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: *The Description Logic Handbook*. Cambridge University Press (2003)
3. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing description logic knowledge bases using formal concept analysis. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*. pp. 230–235 (2007)
4. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automatic Reasoning* 39(3), 385–429 (2007)
5. Calvanese, D., Kharlamov, E., Nutt, W., Zheleznyakov, D.: Evolution of DL-Lite knowledge bases. In: *Proceedings of the 9th International Semantic Web Conference (ISWC-10)*. pp. 112–128 (2010)
6. Cuenca Grau, B., Ruiz, E.J., Kharlamov, E., Zhelenyakov, D.: Ontology evolution under semantic constraints. In: *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR-12)*. pp. 137–147 (2012)
7. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence* 57, 227–270 (1992)

8. Haase, P., Stojanovic, L.: Consistent evolution of OWL ontologies. In: Proceedings of the 2nd European Semantic Web Conference (ESWC-05). pp. 182–197 (2005)
9. Hartel, F.W., de Coronado, S., Dionne, R., Fragoso, G., Golbeck, J.: Modeling a description logic vocabulary for cancer research. *Journal of Biomedical Informatics* 38(2), 114–129 (2005)
10. Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in OWL. In: Proceedings of the 7th International Semantic Web Conference (ISWC-08). pp. 323–338 (2008)
11. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca Grau, B.: Repairing unsatisfiable concepts in OWL ontologies. In: Proceedings of the 3rd European Semantic Web Conference (ESWC-06). pp. 170–184 (2006)
12. Katsuno, H., Mendelzon, A.: Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52(3), 263–294 (1991)
13. Konev, B., Lutz, C., Wolter, F.: Exact learning of tboxes in EL and DL-Lite. In: Informal Proceedings of the 26th International Workshop on Description Logics (DL-13). pp. 341–352 (2013)
14. Kontchakov, R., Wolter, F., Zakharyashev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artificial Intelligence* 174(15), 1093–1141 (2010)
15. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. *Machine Learning* 78(1-2), 203–250 (2010)
16. Nikitina, N., Rudolph, S., Glimm, B.: Interactive ontology revision. *Journal of Web Semantics* 12, 118–130 (2012)
17. Qi, G., Du, J.: Model-based revision operators for terminologies in description logics. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09). pp. 891–897 (2009)
18. Qi, G., Haase, P., Huang, Z., Ji, Q., Pan, J.Z., Völker, J.: A kernel revision operator for terminologies - algorithms and evaluation. In: Proceedings of the 7th International Semantic Web Conference (ISWC-08). pp. 419–434 (2008)
19. Qi, G., Yang, F.: A survey of revision approaches in description logics. In: Proceedings of the 2nd International Conference of Web Reasoning and Rule Systems (RR-08). pp. 74–88 (2008)
20. Ribeiro, M.M., Wassermann, R.: Base revision for ontology debugging. *Journal of Logic Computation* 19(5), 721–743 (2009)
21. Satoh, K.: Nonmonotonic reasoning by minimal belief revision. In: Proceedings of the International Conference on Fifth Generation Computer Systems. pp. 455–462 (1988)
22. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. *Journal of Automated Reasoning* 39(3), 317–349 (2007)
23. Spackman, K.A.: M. d. computing. SNOMED RT and SNOMED CT. *Promise of an international clinical terminology* 17(6), 29 (2000)
24. Staab, S., Studer, R. (eds.): *Handbook on Ontologies*. Springer-Verlag, 2. edn. (2009)
25. Wang, Z., Wang, K., Topor, R.: A new approach to knowledge base revision in DL-Lite. In: Proceedings of the 24th National Conference on Artificial Intelligence (AAAI-10). pp. 369–374 (2010)
26. Zheleznyakov, D., Calvanese, D., Kharlamov, E., Nutt, W.: Updating TBoxes in DL-Lite. In: Proceedings of the 23rd International Workshop on Description Logics (DL-10) (2010)