Author manuscript, published in "International Workshop on Description Logics (DL), Canada (2010)"

hal-00537790, version 1 - 19 Nov 2010

Updating TBoxes in *DL-Lite*

Dmitriy Zheleznyakov, Diego Calvanese, Evgeny Kharlamov*, and Werner Nutt

KRDB Research Centre
Free University of Bozen-Bolzano, Italy
zheleznyakov,calvanese,kharlamov,nutt@inf.unibz.it

Abstract. We study the problem of updates for TBoxes represented in Description Logics of the *DL-Lite* family. *DL-Lite* is at the basis of OWL 2 QL, one of the tractable fragments of OWL 2, the recently proposed revision of the Web Ontology Language. In this paper, we address for the first time the problem of updating TBoxes. We propose some principles that TBox updates should respect. We review known model- and formula-based approaches for updates of logical theories, and exhibit limitations of model-based approaches to handle TBox updates. We propose a novel formula-based approach, and present a polynomial time algorithm to compute TBox updates for *DL-Lite*$_{\mathcal{FR}}$. We also study the relationship between propositional logic satisfiability for Horn clauses and computation of TBox updates for *DL-Lite*.

1 Introduction

Ontology languages, and hence Description Logics (DLs), provide excellent mechanisms for representing structured knowledge, and as such they have traditionally been used for modeling at the conceptual level the static and structural aspects of application domains [1]. A family of DLs that has received great attention recently, due to its tight connection with conceptual data models, such as the Entity-Relationship model and UML class diagrams, is the *DL-Lite* family [2]. Such a family of DLs exhibits nice computational properties, in particular when complexity is measured wrt the size of the data stored in the ABox of a DL ontology [2, 3]. It is also at the basis of the tractable profiles of OWL 2, the forthcoming edition of the W3C standard Web Ontology Language.

The reasoning services that have been investigated for the currently used DLs and implemented in state-of-the-art DL reasoners [4], traditionally focus on so-called standard reasoning, both at the TBox level (e.g., TBox satisfiability, concept satisfiability and subsumption wrt a TBox), and at the ABox level (e.g., knowledge base satisfiability, instance checking and retrieval, and more recently query answering) [5, 6]. Recently, however, the scope of ontologies has broadened, and they are now considered to be not only at the basis of the design and development of information systems, but also for providing support in the maintenance and evolution phase of such systems. Moreover, ontologies are considered to be the premium mechanism through which services operating in a Web context can be accessed, both by human users and by other services.

* The author is co-affiliated with INRIA Saclay, Île-de-France.

hal-00537790, version 1 - 19 Nov 2010

Supporting all these activities, makes it necessary to equip DL systems with additional kinds of inference tasks that go beyond the traditional ones, most notably that of *ontology evolution* [7], where new knowledge is incorporated into an existing KB. Two main types of ontology evolution have been considered, namely revision and update [8].

In *revision*, we assume that the new knowledge is certainly true in the real world. Therefore, every model of a revised KB should satisfy this knowledge and should have minimal distance to the old KB, where the notion of distance depends on the application. An important feature of revision is that the distance is defined "globally" and depends on all the models of the old KB. In [9, 10] revision of DL knowledge bases was considered. In *update*, we assume that the new knowledge reflects a change in the real world, and we update every model of the old KB with this new knowledge. Note that update operators, in contrast to revsion operators, work "locally". In our work we focus on ontology update.

A request for an ontology update (or simply *update request*) represents the need of changing an ontology so as to take into account changes that occur in the domain of interest described by the ontology. In general, such a request is represented by a set of formulas denoting those properties that should be true after the change. In the case where the update request causes an undesirable interaction with the knowledge encoded in the ontology, e.g., by causing the ontology or relevant parts of it to become unsatisfiable, the update request cannot simply be added to the ontology. Instead, suitable changes need to be made in the ontology so as to avoid the undesirable interaction, e.g., by deleting parts of the ontology that conflict with the update request. Different choices are possible in general, corresponding to different update semantics, which in turn give rise to different update results [11]. Moreover, it is necessary to understand whether the desired update result can be represented at all as a KB in the DL at hand.

Previous work on updates in the context of DL ontologies has addressed ABox (or instance level) update [12, 13], where the update request consists of a set of ABox assertions. In [12] the problem is studied for DLs of the *DL-Lite* family, while [13] considers the case of expressive DLs. Both works show that it might be necessary to extend the ontology language with additional features/constructs in order to guarantee that the updated ontology can be represented.

Instead, the problem of TBox level update has not been considered before. In this paper we take first steps at filling this gap. Specifically, for the case of DLs of the *DL-Lite* family, we study the problem of updating a TBox with a set of TBox assertions. We address first the issue of which semantics to adopt for TBox updates, and specify some general principles that updates should respect. This leads us to argue that none of the previously proposed semantics [14–17], neither model-based nor formula-based is totally appropriate: either too many formulas need to be thrown out in the result of the update, or such a result is not representable as a *DL-Lite* TBox. Hence, we propose an alternative formula-based semantics, called *Bold Semantics* , and provide polynomial time algorithms to compute it for various members of the *DL-Lite* family (we restrict the attention to those DLs of the *DL-Lite* family that exhibit polynomial time TBox reasoning, specifically we consider only the case where the interaction between functionality assertions and role inclusions is restricted). The task at the core of our algorithm is the problem of checking *full satisfiability* of a *DL-Lite* TBox, i.e., whether

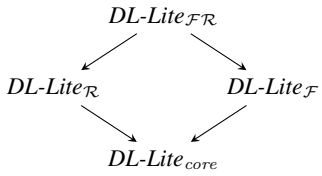


Fig. 1. *DL-Lite* hierarchy.

all atomic concepts and roles are (simultaneously) satisfiable. We provide a novel algorithm for this problem that is based on a reduction to reasoning in propositional binary Horn theories. This gives us also an alternative proof technique that is not based on the Chase for tractability of TBox reasoning in *DL-Lite*.

2 Preliminaries

Description Logics (DLs) [18] are knowledge representation formalisms, tailored for representing the domain of interest in terms of *concepts* and *roles*. In DLs, complex concept and role expressions (or simply, concepts and roles) are obtained starting from atomic concepts and roles (which are simply names) by applying suitable constructs. Concepts and roles are then used in a DL knowledge base (KB) to model the domain of interest. Specifically, a DL KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is formed by two distinct parts, a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. The TBox $\mathcal{T}$ represents the *intensional-level* of the KB, that is, the general knowledge. The ABox provides information on the *instance-level* of the KB. In this paper we focus on a family of DLs called *DL-Lite* [2], that corresponds to one of the tractable fragments of OWL 2, the recently proposed revision of the Web Ontology Language.

The basic logic of the *DL-Lite* family is *DL-Lite*$_{core}$, which includes constructs that are used in all others logics of the family. These constructs are the following:

$$B ::= A \mid \exists R, \qquad C ::= B \mid \neg B, \qquad R ::= P \mid P^-,$$

where A denotes an *atomic concept*, B a *basic concept*, and C a *general concept*. The symbol P denotes an *atomic role*, and R a *basic role*.

A *DL-Lite*$_{core}$ TBox is a set of *concept inclusion assertions* of the form: $B \sqsubseteq C$, and an ABox is a set of *membership assertions* of the form: $A(a),\ P(a, b)$.

The two logics *DL-Lite*$_{\mathcal{F}}$ and *DL-Lite*$_{\mathcal{R}}$ both extend *DL-Lite*$_{core}$. They have ABoxes of the same form as *DL-Lite*$_{core}$, but their TBoxes are different. A *DL-Lite*$_{\mathcal{F}}$ TBox may include *functionality assertions* for roles of the form $(\mathsf{funct}\ R)$. *DL-Lite*$_{\mathcal{R}}$ has *role inclusion assertions* of the form $R_1 \sqsubseteq R_2$ (instead of functionality assertions). There are proposals that consider *DL-Lite*$_{\mathcal{R}}$ also with role disjointness assertions of the form $R_1 \sqsubseteq \neg R_2$, but we do not take them into account in our paper. Both *DL-Lite*$_{\mathcal{F}}$ and *DL-Lite*$_{\mathcal{R}}$ have nice computational properties, for example, knowledge base satisfiability has polynomial-time complexity in the size of the TBox and logarithmic-space complexity in the size of the ABox, so-called *data complexity*.

hal-00537790, version 1 - 19 Nov 2010

hal-00537790, version 1 - 19 Nov 2010

DL-Lite$_{\mathcal{FR}}$ is a hybrid of *DL-Lite*$_{\mathcal{F}}$ and *DL-Lite*$_{\mathcal{R}}$. It allows for both functional assertions and role inclusion assertions in its TBox. The use of functionality and role inclusion assertions together may lead to an increase in the complexity of reasoning. A way to avoid this is to introduce the following syntactic restriction: if $R_1 \sqsubseteq R_2$ appears in a TBox, then $(\mathsf{funct}\ R_2)$ is *not* in the TBox. Hence, when talking about *DL-Lite*$_{\mathcal{FR}}$ knowledge bases in this paper, we assume they satisfy the syntactic restriction above.

In Figure 1 we list the four logics of the *DL-Lite* family and show the relationships between them in terms of expressiveness. If there is an arrow from a logic X to a logic Y in the figure, it means that the logic Y is more expressive than X.

The semantics of a DL is given in terms of first order interpretations. Let Δ be a fixed countably infinite set. All interpretations that we consider are over the same domain Δ.

An *interpretation* $\mathcal{I}$ is a function $\cdot^{\mathcal{I}}$ that assigns to each concept C a subset $C^{\mathcal{I}}$ of Δ, and to each role R a binary relation $R^{\mathcal{I}}$ over Δ in such a way that $A^{\mathcal{I}} \subseteq \Delta$, $P^{\mathcal{I}} \subseteq \Delta \times \Delta$, $(\neg B)^{\mathcal{I}} = \Delta \setminus B^{\mathcal{I}}$, and

$$(\exists R)^{\mathcal{I}} = \left\{a \mid \exists a'.\, (a, a') \in R^{\mathcal{I}}\right\}, \qquad (R^{-})^{\mathcal{I}} = \left\{(a_2, a_1) \mid (a_1, a_2) \in R^{\mathcal{I}}\right\}.$$

An interpretation $\mathcal{I}$ is a *model* of an inclusion assertion $D_1 \sqsubseteq D_2$ if $D_1^{\mathcal{I}} \subseteq D_2^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a *model* of a functionality assertion $(\mathsf{funct}\ R)$ if R is a partial function over Δ, that is, $\mathcal{I} \models \forall x, y_1, y_2.\, \left(R^{\mathcal{I}}(x, y_1) \wedge R^{\mathcal{I}}(x, y_2)\right) \rightarrow y_1 = y_2$.

Given an assertion F and an interpretation $\mathcal{I}$, we denote by $\mathcal{I} \models F$ the fact that $\mathcal{I}$ is a model of F. A model $\mathcal{I}$ is a model of a knowledge base (KB) $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ (denoted as $\mathcal{I} \models \mathcal{K}$) if $\mathcal{I}$ is a model of each of the assertions of $\mathcal{T} \cup \mathcal{A}$. A KB is *satisfiable* if it has at least one model. A KB $\mathcal{K}$ logically implies an assertion F, written $\mathcal{K} \models F$, if all models of $\mathcal{K}$ are also models of F. Similarly, a TBox $\mathcal{T}$ *logically implies* an assertion F, written $\mathcal{T} \models F$, if all models of $\mathcal{T}$ are also models of F.

Let $\mathcal{T}$ be a set of TBox assertions. The deductive *closure of* $\mathcal{T}$, denoted $cl(\mathcal{T})$, is the set of all assertions that are entailed by $\mathcal{T}$. Clearly, the closure $cl(\mathcal{T})$ is quadratic in the number of atoms of $\mathcal{T}$ and can be computed in time polynomial wrt the size of $\mathcal{T}$.

3 Understanding TBox Updates

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB and $\mathcal{U}$ be a set of (TBox or/and ABox) assertions, called an update request. What we want to study is how to "incorporate" the assertions $\mathcal{U}$ into $\mathcal{K}$, that is, to perform an *update* of $\mathcal{K}$. In this paper we consider only updates on the TBox level (*TBox updates*), that is, when $\mathcal{U}$ consists of TBox assertions only.

When dealing with updates, both in the knowledge management and the AI community, it is generally accepted that the updated KB $\mathcal{K}'$, or the *update* for short, should comply with the principle of *Minimality of Change* [11, 17], which states that the knowledge base should change as little as possible if new information is incorporated. There are different approaches to updates, suitable for particular applications, and the current belief is there is no general notion of minimality that will "do the right thing" under all circumstances [17]. A number of candidate semantics for updates have appeared in the literature [14–17]. All these approaches can be classified into two groups: *model-based* and *formula-based*.

hal-00537790, version 1 - 19 Nov 2010

Let us first understand what are the requirements for updates of knowledge bases and then review known model- and formula-based approaches.

3.1 Principles of TBox Updates

Let $\mathcal{T}$ be a TBox, B a basic concept, and R a basic role occurring in $\mathcal{T}$. We say that B (resp. R) is *satisfiable in* $\mathcal{T}$ if there is a model $\mathcal{I} \models \mathcal{T}$ of $\mathcal{T}$ such that $B^{\mathcal{I}} \neq \emptyset$ (resp. $R^{\mathcal{I}} \neq \emptyset$). If all the atomic concepts and roles occurring in $\mathcal{T}$ are satisfiable, then we say that $\mathcal{T}$ is *fully satisfiable*. Intuitively, a concept "makes sense" if one can instantiate it and we assume that we update TBoxes that make sense, that is, that are fully-satisfiable.

Satisfiability Preservation. A TBox update is a modification of a KB on the schema level. Such updates make sense, for example, when a company decides to restructure, say, the sales department, and the update $\mathcal{U}$ consists of new requirements for the department. Our first concern is that updates should not make parts of the schema, or TBox constructs, useless, that is, unsatisfiable. For example, for a basic concept of an enterprise ontology, say the concept *Manager*, we want to reject updates that eliminate managers from the enterprise, that is, that force *Manager* to be unsatisfiable.

Protection. Our next expectation is that the schema update of the sales department should not affect the schema of, say, the production department. At the same time we do not mind if it affects the schemas of other departments, like for instance accounting. That is, we would like some fragment of $\mathcal{T}$, denoted $\mathcal{T}_p$, to be *protected* from any changes, that is, we would like $\mathcal{T}_p$ to be kept in the KB after the update. Therefore, we *accept* an update request $\mathcal{U}$ only if $\mathcal{T}_p \cup \mathcal{U}$ is fully satisfiable, otherwise we *reject* $\mathcal{U}$.

To sum up these desiderata, we list our *update principles*.

Satisfiability Preservation. Updates should preserve satisfiability of basic concepts and roles.

Protection. Updates should preserve the protected fragment of the KB.

3.2 Model-Based Approach to Semantics

Poggi et al. [19, 12] proposed to use Winslett's semantics to update ABoxes. Let us try to understand whether this approach is suitable for TBox updates.

Under the model-based paradigm, the objects of change are individual models $\mathcal{I}$ of $\mathcal{T}$. For a model $\mathcal{I}$ of $\mathcal{T}$, an update with $\mathcal{U}$ results in a set of models of $\mathcal{U}$. In order to update the entire TBox $\mathcal{T}$ with $\mathcal{U}$, one has to

(i) update every model $\mathcal{I} \models \mathcal{T}$ with $\mathcal{U}$, and then
(ii) take the union of the resulting models.

To define the update formally we recall the following definitions. We say that an interpretation $\mathcal{I}$ *is contained in* $\mathcal{I}'$, written $\mathcal{I} \subseteq \mathcal{I}'$, if for every atomic concept or role symbol S it holds that $S^{\mathcal{I}} \subseteq S^{\mathcal{I}'}$. We write $\mathcal{I} \subsetneq \mathcal{I}'$ if $\mathcal{I} \subseteq \mathcal{I}'$ and not $\mathcal{I}' \subseteq \mathcal{I}$. We denote with $\ominus$ the symmetric difference between sets according to the standard definition.

hal-00537790, version 1 - 19 Nov 2010

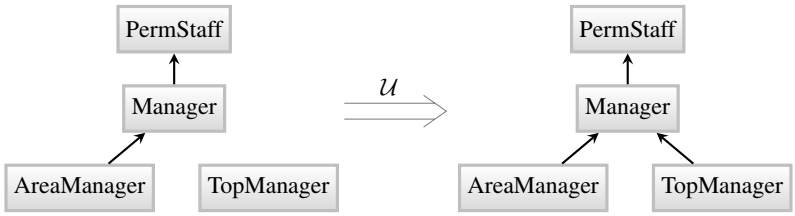


Fig. 2. Updates of ontologies. $\mathcal{U} = \{TopManager \sqsubseteq Manager\}$.

Let $\mathcal{T}_p \subseteq \mathcal{T}$ be the protected fragment of $\mathcal{T}$ and $\mathcal{U}$ an update request accepted for $\mathcal{T}_p$. The *update of an interpretation $\mathcal{I}$ with $\mathcal{U}$ wrt $\mathcal{T}_p$*, denoted *w-upd*$_{\mathcal{T}_p}(\mathcal{I},\mathcal{U})$, where '*w*' indicates Winslett's semantics, is the set of interpretations defined as follows:

$$\{\mathcal{I}' \mid \mathcal{I}' \in Mod(\mathcal{T}_p \cup \mathcal{U}), \text{there is no } \mathcal{I}'' \in Mod(\mathcal{T}_p \cup \mathcal{U}) \text{ s.t. } \mathcal{I} \ominus \mathcal{I}'' \subsetneq \mathcal{I} \ominus \mathcal{I}'\}.$$

Then the *update of a TBox $\mathcal{T}$ with $\mathcal{U}$ wrt $\mathcal{T}_p$* is the following set of interpretations:

$$\textit{w-upd}_{\mathcal{T}_p}(\mathcal{T},\mathcal{U}) = \bigcup_{\mathcal{I} \in Mod(\mathcal{T})} \textit{w-upd}_{\mathcal{T}_p}(\mathcal{I},\mathcal{U}).$$

Returning to a user the result of an update as a set of models is not desirable. What we want is to return a KB that describes exactly this set of models. We say that a TBox $\mathcal{T}'$ *represents* the update *w-upd*$_{\mathcal{T}_p}(\mathcal{T},\mathcal{U})$ if $Mod(\mathcal{T}') = \textit{w-upd}_{\mathcal{T}_p}(\mathcal{T},\mathcal{U})$.

Example 1. Consider the TBox $\mathcal{T}$ of an enterprise on the left diagram of Figure 2. In *DL-Lite*$_{core}$ the diagram can be written as follows:

$$Manager \sqsubseteq PermStaff, \qquad AreaManager \sqsubseteq Manager,$$

where *PermStaff* stand for *Permanent Staff*. The TBox says that every $Manager$ belongs to $PermStaff$ and every $AreaManager$ is a $Manager$. Suppose the TBox is under construction and it was decided to extend it by introducing the inclusion assertion that every $TopManager$ is a $Manager$, that is,

$$\mathcal{U} = \{TopManager \sqsubseteq Manager\}.$$

Since there are no disjointness assertions in both $\mathcal{T}$ and $\mathcal{U}$, the update request will be accepted for $\mathcal{T}$, regardless of which fragment is protected, and the desired result of the update is the one in the right diagram of Figure 2. Unfortunately, Winslett's semantics gives an undesirable result.

First, consider the following model $\mathcal{I}$ of $\mathcal{T}$:

$$TopManager^{\mathcal{I}} = \{john\}, \quad Manager^{\mathcal{I}} = \{frank\}, \quad PermStaff^{\mathcal{I}} = \{frank\}.$$

Assume that the protected fragment of $\mathcal{T}$ is empty. Then, according to Winslett's semantics, the update of the model $\mathcal{I}$ contains the following interpretation $\mathcal{I}'$ is in the update of $\mathcal{I}$, which is a model of $\mathcal{U}$ that differs minimally from $\mathcal{I}$:

$$TopManager^{\mathcal{I}'} = \{john\}, \quad Manager^{\mathcal{I}'} = \{john, frank\}, \quad PermStaff^{\mathcal{I}'} = \{frank\}.$$

hal-00537790, version 1 - 19 Nov 2010

As one can see, in $\mathcal{I}'$ there is a $Manager$, $john$, who does *not* belong to $PermStaff$. Therefore, the update *w-upd*$(\mathcal{T},\mathcal{U})$ does *not* satisfy the assertion $Manager \sqsubseteq PermStaff$.

Second, every *DL-Lite* representation $\mathcal{T}'$ of *w-upd*$(\mathcal{T},\mathcal{U})$ should satisfy the following assertions, which we denote as $\mathcal{T}_0$, that is, $\mathcal{T}_0 \subseteq \mathcal{T}'$:

$$TopManager \sqsubseteq Manager, AreaManager \sqsubseteq Manager, AreaManager \sqsubseteq PermStaff.$$

Is it the case that $\mathcal{T}' = \mathcal{T}_0$? It turns out that not. Consider the following model. Let $\mathcal{I}''$ be an interpretation, where all concepts are empty, except for $Manager$, which contains one individual, say $fred$. It is easy to see that $\mathcal{I}'' \models \mathcal{T}_0$ and $\mathcal{I}'' \models \mathcal{U}$, but it cannot be obtained by minimally changing a model of $\mathcal{T}$. Intuitively, there is no reason for $fred$ to have become a $Manager$.

Therefore, there should be some other inclusion assertions in $\mathcal{T}'$, besides the ones of $\mathcal{T}_0$, that forbid the model $\mathcal{I}''$. One can see that these assertions should be entailed by $\mathcal{T} \cup \mathcal{U}$. Otherwise there are models of $\mathcal{T}$ whose update is not expressed by $\mathcal{T}'$. Hence, the only candidate to be included in $\mathcal{T}'$ is $Manager \sqsubseteq PermStaff$, but it cannot be in $\mathcal{T}'$, due to the first observation above. Therefore, the update is not expressible in *DL-Lite*. □

We conclude that:

(i) Winslett's semantics cannot be expressed by *DL-Lite* TBoxes.

(ii) The principle of minimal change at the level of interpretations forces one to give up important assertions at the TBox level (in Example 1, we gave up the assertion $Manager \sqsubseteq PermStaff$).

We consider this situation as unsatisfactory. Hence, we next examine the formula-based approach to updates and their notion of minimality.

3.3 Formula-Based Approach to Semantics

The key notion in this approach is the one of a *maximal non-contradicting set* of formulas, which we introduce now.

Let $\mathcal{T}$ be a TBox and $\mathcal{U}$ be an update request that is accepted for $\mathcal{T}_p$. We define a *maximal non-contradicting set of formulas for* $\mathcal{T}$ *and* $\mathcal{U}$, denoted by $\mathcal{T}_m$, as a set of TBox assertions that satisfies the conditions:

(i) $\mathcal{T} \models \mathcal{T}_m$,

(ii) $\mathcal{T}_m \cup \mathcal{U}$ is fully satisfiable,

(iii) the set $\mathcal{T}_m$ is maximal (wrt set inclusion) among the sets that satisfy *(i)* and *(ii)*, that is, there is no $\hat{\mathcal{T}}$ satisfying *(i)* and *(ii)* such that $\mathcal{T}_m \subset \hat{\mathcal{T}}$.

Intuitively, $\mathcal{T}_m$ keeps as many TBox assertions as possible that are entailed by $\mathcal{T}$ and do not conflict with $\mathcal{U}$.

Obviously, the set $\mathcal{T}_m$ is not unique. We denote the set of all such $\mathcal{T}_m$ for $\mathcal{T}$ and $\mathcal{U}$ as $\mathcal{M}(\mathcal{T},\mathcal{U})$. There are two main approaches to construct updates $\mathcal{T}'$ of $\mathcal{T}$ with $\mathcal{U}$ based on $\mathcal{T}_m$ [11, 17].

hal-00537790, version 1 - 19 Nov 2010

WIDTIO. The first approach is called *When In Doubt Throw It Out*, or *WIDTIO* for short. It suggests to add to $\mathcal{U}$ the intersection of all $\mathcal{T}_m$-s, as on the left of Equation 1:

$$\mathcal{T}' = \mathcal{U} \cup \bigcap_{\mathcal{T}_m \in \mathcal{M}(\mathcal{T},\mathcal{U})} \mathcal{T}_m, \qquad \mathcal{T}' = \mathcal{U} \cup \{ \bigvee_{\mathcal{T}_m \in \mathcal{M}(\mathcal{T},\mathcal{U})} (\bigwedge_{\phi \in \mathcal{T}_m} \phi)\}. \quad (1)$$

Cross-Product. According to this approach, one adds to $\mathcal{U}$ the disjunction of all $\mathcal{T}_m$-s, viewing each $\mathcal{T}_m$ as the conjunction of its assertions, as on the right of Equation 1.

Example 2. Consider the *DL-Lite* ontology from Example 1 (Figure 2) and the update request $\mathcal{U} = \{AreaManager \sqsubseteq \neg PermStaff\}$. It is easy to see that $\mathcal{U} \cup \mathcal{T}$ is not fully satisfiable and in order to resolve the conflict one can drop either $Manager \sqsubseteq PermStaff$ or $AreaManager \sqsubseteq Manager$. Thus, $\mathcal{M}(\mathcal{T},\mathcal{U}) = \{\mathcal{T}_m^{(1)}, \mathcal{T}_m^{(2)}\}$, where $\mathcal{T}_m^{(1)} = \{Manager \sqsubseteq PermStaff\}$, and $\mathcal{T}_m^{(2)} = \{AreaManager \sqsubseteq Manager\}$. Let us now consider WIDTIO and Cross-Product semantics. According to the left formula of Equation 1, the TBox under WIDTIO semantics is equal to

$$\mathcal{U} \cup \left(\mathcal{T}_m^{(1)} \cap \mathcal{T}_m^{(2)}\right) = \mathcal{U} \cup \emptyset = \mathcal{U} = \{AreaManager \sqsubseteq \neg PermStaff\}.$$

The TBox under Cross-Product semantics is

$$\mathcal{U} \cup \{(Manager \sqsubseteq PermStaff) \vee (AreaManager \sqsubseteq Manager)\},$$

where we have combined DL notation with First Order Logic notation. □

As one can see from the example above, a disadvantage of the WITDIO approach is that it may lose a lot of assertions entailed by $\mathcal{T}$ that *do not* conflict with $\mathcal{U}$. On the other extreme is the Cross-Product approach that suggests to keep all possible entailed and not conflicting assertions. A drawback of the approach is that the result of the update cannot be represented in *DL-Lite* anymore since it requires disjunction. Another drawback is that the resulting set of formulas may be exponentially large wrt the original TBox.

Therefore, any practical solution should be one where one chooses *some* $\mathcal{T}_m^{(0)}$ among the $\mathcal{T}_m$, where the result of the update is:

$$\mathcal{T}' = \mathcal{U} \cup \mathcal{T}_m^{(0)}.$$

We call this semantics *Bold Semantics.* The question is which $\mathcal{T}_m$ to choose. There are basically three options. Choose *(i)* an arbitrary one, *(ii)* one that has maximal cardinality, *(iii)* one that fulfills some preferences. For all options, the solution is expressible in *DL-Lite*. Note that we rely for this on the fact that in *DL-Lite* the set of assertions entailed by a TBox is finite.

The first option has the advantage that $\mathcal{T}'$ is expressible in *DL-Lite* and can be computed in polynomial time. Figure 3 presents a nondeterministc algorithm that, given a TBox $\mathcal{T}$ and an update request $\mathcal{U}$, returns a set $\mathcal{T}_m \subseteq cl(\mathcal{T})$ that is a maximal non-contradicting set of assertions for $\mathcal{T}$ and $\mathcal{U}$. The algorithm loops at most as many times as there are assertions in $cl(\mathcal{T})$. The number of such assertions is at most quadratic in

hal-00537790, version 1 - 19 Nov 2010

INPUT: sets $\mathcal{T}, \mathcal{U}$ of TBox assertions, $\mathcal{T}_p \subseteq \mathcal{T}$ fully satisfiable with $\mathcal{U}$
OUTPUT: a set $\mathcal{T}_m \subseteq cl(\mathcal{T})$ of TBox assertions
[1] $\mathcal{T}_m := \mathcal{U} \cup \mathcal{T}_p$; $\mathcal{S} := cl(\mathcal{T})$
[2] **repeat**
[3] **choose some** $\phi \in \mathcal{S}$; $\mathcal{S} := \mathcal{S} \setminus \{\phi\}$
[4] **if** $\{\phi\} \cup \mathcal{T}_m$ is fully satisfiable **then** $\mathcal{T}_m := \mathcal{T}_m \cup \{\phi\}$
[5] **until** $\mathcal{S} = \emptyset$

Fig. 3. Algorithm *NDMax*$(\mathcal{T}, \mathcal{T}_p, \mathcal{U})$ for nondeterministic computation of $\mathcal{T}_m$

the number of atomic concepts and roles. The crucial step is a check for full satisfiability, which is performed once per loop. If the latter test is polynomial in the size of the input, like in *DL-Lite*$_{\mathcal{FR}}$ (see Section 4), then the entire runtime of the algorithm is polynomial. For the second option we showed $\mathcal{T}_m$ computation is NP-hard, but we cannot present the proof due to lack of space. The third option is good as far as one has reasonable preferences either on the concepts or assertions of the TBox, that gives us polynomial time computation.

Example 3. Consider the KB and the update request from Example 2. As it has been mentioned, $\mathcal{M}(\mathcal{T}, \mathcal{U}) = \{\mathcal{T}_m^{(1)}, \mathcal{T}_m^{(2)}\}$. According to the Bold Semantics computed by the algorithm *NDMax*, the result of the update is a TBox $\mathcal{T} = \mathcal{U} \cup \mathcal{T}_m^{(0)}$ for some $\mathcal{T}_m^{(0)} \in \mathcal{M}(\mathcal{T}, \mathcal{U})$. Thus, the result of the update is either $\mathcal{U} \cup \{AreaManager \sqsubseteq Manager\}$ or $\mathcal{U} \cup \{Manager \sqsubseteq PermStaff\}$. In the former case, the ontology makes sense if manageres could be temporary staff, in the latter one, if area managers are not necessarily managers. Selecting one or the other of these two options could be done by the use of preferences. But we do not consider this here. □

Theorem 4 (Correctness of Semantics). *Bold Semantics satisfies the principles of Satisfiability Preservation and Protection.*

4 Checking Full Satisfiability

Testing full satisfiability is the key operation in computing updates under Bold Semantics. We show that for *DL-Lite*$_{\mathcal{FR}}$ the problem of checking full satisfiability of a TBox can be translated into a problem of propositional Horn logic. The translation can be used as the starting point for the design of efficient algorithms and it provides additional insight as to why full satisfiabilty can be solved in polynomial time for *DL-Lite*$_{\mathcal{FR}}$.

As a first step, we define a translation function ν that translates TBoxes $\mathcal{T}$ into propositional theories $\nu(\mathcal{T})$. For every basic concept B resulting from the signature of $\mathcal{T}$ we introduce a fresh propositional variable v_B and for every basic role R we introduce the two variables $v_{\exists R}$ and $v_{\exists R^-}$ and denote the set of all such variables as $\mathcal{V}_\mathcal{T}$. Then $\nu(\mathcal{T})$ consists of all propositional formulas that can be obtained from $\mathcal{T}$ using the translation in Table 1.

Let $\mathcal{V}$ be a set of propositional variables and $\mathcal{F}$ a set of formulas over $\mathcal{V}$. Then we say that $\mathcal{F}$ is *fully satisfiable* (over $\mathcal{V}$) if $\mathcal{F} \cup \{v\}$ is satisfiable for every $v \in \mathcal{V}$.

hal-00537790, version 1 - 19 Nov 2010

TBox assertion ϕ	PL formulas $\nu(\phi)$
$B_1 \sqsubseteq B_2$	$v_{B_1} \rightarrow v_{B_2}$
$B_1 \sqsubseteq \neg B_2$	$v_{B_1} \rightarrow \neg v_{B_2}$
$R_1 \sqsubseteq R_2$	$v_{\exists R_1} \rightarrow v_{\exists R_2}$, $v_{\exists R_1^-} \rightarrow v_{\exists R_2^-}$

Table 1. Translation of *DL-Lite*$_{\mathcal{FR}}$ TBoxes to propositional theories

Theorem 5. *Let $\mathcal{T}$ be a DL-Lite$_{\mathcal{R}}$ TBox. Then $\mathcal{T}$ is fully satisfiable if and only if $\nu(\mathcal{T})$ is fully satisfiable over $\mathcal{V}_{\mathcal{T}}$.*

Proof. The "only if" direction being clear, we only show the "if" direction.

Suppose that $\nu(\mathcal{T})$ is fully satisfiable over $\mathcal{V}_{\mathcal{T}}$. Then for every basic concept B there is a truth assignment α_B for the variables in $\mathcal{V}_{\mathcal{T}}$ such that $\nu(\mathcal{T}) \cup \{v_B\}$ is satisfiable. Intuitively, this can be seen as putting a test individual into B and letting α_B propagate this individual into additional concepts B' so that the inclusions in $\mathcal{T}$ are satisfied.

Now we choose, for every B, a distinct element $d_B \in \Delta$. Moreover, we define a mapping J that maps every basic concept B' to a subset of Δ by defining $J(B') = \{d_B \mid \alpha_B(v_{B'}) = \mathsf{true}\}$. Intuitively, $J(B')$ consists of all the test individuals d_B that ended up in B' by way of their α_B. Note that due to the construction we have that $J(B') \subseteq J(B'')$ whenever $B' \sqsubseteq B'' \in \mathcal{T}$.

We now define an interpretation $\mathcal{I}'$ by setting $A^{\mathcal{I}'} = J(A)$ for every atomic concept A and $P^{\mathcal{I}'} = J(\exists P) \times J(\exists P^-)$ for every atomic role P. That is, $P^{\mathcal{I}'}$ is the Cartesian product of the sets to which J maps the expressions for the domain and range of P. Clearly, in this way we have that $(\exists P)^{\mathcal{I}'} = J(\exists P)$ and $(\exists P^-)^{\mathcal{I}'} = J(\exists P^-)$. This shows that $\mathcal{I}'$ is a model of $\mathcal{T}$ such that $A^{\mathcal{I}'} \neq \emptyset$ and $P^{\mathcal{I}'} \neq \emptyset$ for all atomic A and P. □

Note that the proof above shows as a byproduct that a fully satisfiable TBox can be fully satisfied by a finite model.

If $\mathcal{T}$ is a *DL-Lite*$_{\mathcal{FR}}$-TBox, we say that an atomic role P is *functional* if $\mathcal{T}$ contains $(\mathsf{funct}\ P)$ or $(\mathsf{funct}\ P^-)$. We say that P *has a subrole* if P or P^- occurs on the right-hand side of some role inclusion.

Lemma 6. *Let $\mathcal{T}$ be a DL-Lite$_{\mathcal{R}}$-TBox and $\mathcal{F}$ a set of functionality assertions. Suppose that no functional role in $\mathcal{T} \cup \mathcal{F}$ has a subrole. Then $\mathcal{T} \cup \mathcal{F}$ is fully satisfiable if $\mathcal{T}$ is fully satisfiable.*

Proof. Let $\mathcal{I}'$ be an intepretation that fully satisfies $\mathcal{T}$. Let D be the set of elements of Δ that are in the interpretation of some atomic concept or role. Without loss of generality we can assume that D is at most countable and that $\Delta \setminus D$ has at least countably many elements. Then there exist countably many sets $D_1, D_2, \ldots \subseteq \Delta$ such that *(i)* every set D_i, $i \in \mathbb{N}$, has the same cardinality as D and *(ii)* the D_i are mutually disjoint.

For every $i \in \mathbb{N}$, let $m_i \colon D \rightarrow D_i$ be a bijection. We use the m_i to extend the interpretations of atomic concepts from D to the union of the D_i. Technically, we define a new interpretation $\mathcal{I}$ by letting $A^{\mathcal{I}} = \bigcup_{i \in \mathbb{N}} m_i(A^{\mathcal{I}'})$ for every atomic concept A. The definition of the role interpretations needs some preparation. For every atomic role P, let $\delta'_P = (\exists P)^{\mathcal{I}'}$ be the domain of P with respect to $\mathcal{I}'$ and $\rho'_P = (\exists P^-)^{\mathcal{I}'}$ be the

hal-00537790, version 1 - 19 Nov 2010

range. We define $\delta_P = \bigcup_{i\in\mathbb{N}} m_i(\delta'_P)$ and, similarly, $\rho_P = \bigcup_{i\in\mathbb{N}} m_i(\rho'_P)$. Note that due to our construction both δ_P and ρ_P are countably infinite.

Now, if P is functional, then let $P^{\mathcal{I}}$ be the graph of an arbitrary bijective function from δ_P to ρ_P. Otherwise, let $P^{\mathcal{I}} = \delta_P \times \rho_P$. Clearly, by construction we have that $(\exists P)^{\mathcal{I}} = \delta_P$ and $(\exists P^-)^{\mathcal{I}} = \rho_P$. Hence, $\mathcal{I}$ satisfies all concept inclusions of $\mathcal{T}$ and all functionality assertions.

Moreover, if $R \sqsubseteq R'$ is a role inclusion in $\mathcal{T}$, we have that $\delta'_R \subseteq \delta'_{R'}$ and $\rho'_R \subseteq \rho'_{R'}$, which implies that $\delta_R \subseteq \delta_{R'}$ and $\rho_R \subseteq \rho_{R'}$. Hence, $R^{\mathcal{I}} \subseteq R'^{\mathcal{I}}$, since R' is not functional and therefore $R'^{\mathcal{I}}$ is the Cartesian product of $\delta_{R'}$ and $\rho_{R'}$. This shows that $\mathcal{I}$ is a model of $\mathcal{T} \cup \mathcal{F}$. □

Recall that in a *DL-Lite*$_{\mathcal{FR}}$ TBox, there can be no role inclusions with a functional role on the right hand side. In addition, we assume that TBoxes do not contain disjointness axioms for roles. Thus, the preceding lemma is applicable.

Theorem 7. *Let $\mathcal{T}$ be a DL-Lite$_{\mathcal{FR}}$ TBox. Then $\mathcal{T}$ is fully satisfiable if and only if $\nu(\mathcal{T})$ is fully satisfiable over $\mathcal{V}_{\mathcal{T}}$.*

Since satisfiability of a set of propositional Horn clauses can be checked in linear time, checking full satisfiablity can be done in time quadratic in the size of the clause set. In [2], polynomiality of concept satisfiablity in *DL-Lite*$_{\mathcal{FR}}$ has been proved using the Chase technique. The techniques used for showing Theorem 3 above provide an alternative proof.

5 Conclusion

To the best of our knowledge, our paper presents the first work on updates for DL TBoxes. We tried to understand what are the natural requirements for such updates and proposed two principles: Satisfiability Preservation and Protection. On the basis of these principles, we examined the well-known semantics for updates poposed by Winslett, which has already been applied by Poggi et al. [19] to ABox updates. The approach turned out to be unintuitive and moreover, the TBox languages of the *DL-Lite* family are not closed under such updates. As an alternative, we examined two formula-based approaches to update semantics: WIDTIO and the Product Approach. The former one leads to an inappropriate loss of knowledge, while for the latter update results are not expressible in *DL-Lite*. As a consequence, we proposed a new semantics for TBox updates, Bold Semantics, that satisfies both our principles. We showed that TBoxes resulting from updates under our semantics can be computed in polynomial time for *DL-Lite*$_{\mathcal{FR}}$. Moreover, we exhibited a tight connection between update computation and reasoning with propositional Horn formulas. This connection can be used as the starting point for the design of efficient update algorithms and it provides additional insight as to why TBox reasoning can be solved in polynomial time for *DL-Lite*$_{\mathcal{FR}}$.

Acknowledgements

The authors are supported by the EU project Ontorule (ICT-231875). The third author is also supported by the European Research Council grant Webdam (under FP7), agreement n. 226513.

hal-00537790, version 1 - 19 Nov 2010

References

1. Borgida, A., Brachman, R.J.: Conceptual modeling with description logics. [18] chapter 10 349–372
2. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. of Automated Reasoning **39**(3) (2007) 385–429
3. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The *DL-Lite* family and relations. J. of Artificial Intelligence Research **36** (2009) 1–69
4. Möller, R., Haarslev, V.: Description logic systems. [18] chapter 8 282–305
5. Haarslev, V., Möller, R.: On the scalability of description logic instance retrieval. J. of Automated Reasoning **41**(2) (2008) 99–142
6. Calvanese, D., De Giacomo, G., Lenzerini, M.: Conjunctive query containment and answering under description logics constraints. ACM Trans. on Computational Logic **9**(3) (2008) 22.1–22.31
7. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: Classification and survey. Knowledge Engineering Review **23**(2) (2008) 117–152
8. Katsuno, H., Mendelzon, A.: On the difference between updating a knowledge base and revising it. In: Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'91). (1991) 387–394
9. Qi, G., Du, J.: Model-based revision operators for terminologies in description logics. In: Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009). (2009) 891–897
10. Flouris, G.: On belief change in ontology evolution. AI Communications—The Eur. J. on Artificial Intelligence **19**(4) (2006)
11. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates and counterfactuals. Artificial Intelligence **57** (1992) 227–270
12. De Giacomo, G., Lenzerini, M., Poggi, A., Rosati, R.: On instance-level update and erasure in description logic ontologies. J. of Logic and Computation, Special Issue on Ontology Dynamics **19**(5) (2009) 745–770
13. Liu, H., Lutz, C., Milicic, M., Wolter, F.: Updating description logic ABoxes. In: Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006). (2006) 46–56
14. Abiteboul, S., Grahne, G.: Update semantics for incomplete databases. In: Proc. of the 11th Int. Conf. on Very Large Data Bases (VLDB'85). (1985)
15. Ginsberg, M.L., Smith, D.E.: Reasoning about action I: A possible worlds approach. Technical Report KSL-86-65, Knowledge Systems, AI Laboratory (1987)
16. Winslett, M.: A model-based approach to updating databases with incomplete information. ACM Trans. on Database Systems **13**(2) (1988) 167–196
17. Winslett, M.: Updating Logical Databases. Cambridge University Press (1990)
18. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2003)
19. De Giacomo, G., Lenzerini, M., Poggi, A., Rosati, R.: On the update of description logic ontologies at the instance level. In: Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006). (2006) 1271–1276