On the Update of Description Logic Ontologies at the Instance Level

Giuseppe De Giacomo, Maurizio Lenzerini, Antonella Poggi, Riccardo Rosati

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
lastname@dis.uniroma1.it

Abstract

We study the notion of update of an ontology expressed as a Description Logic knowledge base. Such a knowledge base is constituted by two components, called TBox and ABox. The former expresses general knowledge about the concepts and their relationships, whereas the latter describes the state of affairs regarding the instances of concepts. We investigate the case where the update affects only the instance level of the ontology, i.e., the ABox. Building on classical approaches on knowledge base update, our first contribution is to provide a general semantics for instance level update in Description Logics. We then focus on *DL-Lite*, a specific Description Logic where the basic reasoning tasks are computationally tractable. We show that *DL-Lite* is closed with respect to instance level update, in the sense that the result of an update is always expressible as a new *DL-Lite* ABox. Finally we provide an algorithm that computes the result of an update in *DL-Lite*, and we show that it runs in polynomial time with respect to the size of both the original knowledge base and the update formula.

Introduction

Several areas of Computer Science and various application domains have witnessed a growing interest in ontologies in the last years. In particular, ontologies are considered as one of the key concepts in the Semantic Web (Berners Lee, Hendler, & Lassila 2001), where they can be used to describe the semantics of information at various sites, overcoming the problem of implicit and hidden knowledge, and thus enabling content exchange. It is widely accepted that Description Logics (Baader *et al.* 2003) (DL) provide a solid foundation for ontology representation and reasoning. Considering an ontology as a knowledge base expressed in DL enables re-phrasing system services of ontology tools in terms of logical reasoning problems. In turn, this view allows us to take advantage of the whole body of research on algorithms for and complexity of reasoning in DLs, in the endeavor to build well-founded tools supporting inferences over ontologies (Acciarri *et al.* 2005; Baader, Horrocks, & Sattler 2005).

While results on DLs may directly provide effective reasoning techniques to ontology management tools (Horrocks 1998; Haarslev & Möller 2001; Sirin & Parsia 2004), they cannot be used to support other tasks. One notable example is ontology update. Update is related to the need of changing an ontology in order to reflect a change in the domain of interest the ontology is supposed to represent. Generally speaking, an update is represented by a formula that is intended to sanction a set of properties that are true in the state resulting from the change. One of the major challenges when dealing with an update is how to react to the case where the update is inconsistent with the current knowledge. Since a principled approach to this issue in the context of ontologies is missing, existing ontology management tools[1] adopt ad-hoc solutions to this problem, for example, just rejecting the update. Indeed, despite the importance of update, this issue is largely unexplored in both ontologies and DLs. Notable exceptions are (Haase & Stojanovic 2005; Liu *et al.* 2006). In particular, in (Liu *et al.* 2006) the authors propose a formal semantics for updates in DLs, and present interesting results on various aspects related to computing updates. However, since the problem is addressed under the assumption that the knowledge base is specified only at the extensional (i.e., instance) level, the paper does not take into account the impact of the intensional level on ontology update.

The aim of this paper is to present the first results of a systematic investigation on the notion of update of ontologies expressed as DL knowledge bases. Each such knowledge base is constituted by two components, called TBox and ABox. The former expresses the intensional level, i.e., the general knowledge about the concepts and their relationships, whereas the latter describes the state of affairs regarding the instances of concepts, called the extensional level. Although the problem of update in its generality should consider the case of updating the whole knowledge base, i.e., either the TBox (Flouris, Plexousakis, & Antoniou 2005; Meyer, Lee, & Booth 2005), or the ABox, or both, here we restrict our attention to the case where the ontology is specified by both a TBox and an ABox, but the update affects only the instance level of the ontology, i.e., the ABox. We believe that this setting, although simplified, not only allows us to study the fundamental properties of update, but is very relevant in practice, since, in many applications, the knowledge about the instances will change much more frequently than the conceptualization of the domain of interest repre-

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

[1] http://www.xml.com/2002/11/06/Ontology_Editor_Survey.html.

sented by the TBox.

The main contributions of this paper are as follows.

- First, we define the notion of update of the extensional level of an ontology. Building on classical approaches on knowledge base update, we provide a general semantics for instance level update in DLs. In particular, we follow the approach of (Liu *et al.* 2006), and we adapt Winslett's semantics (Winslett 1988; 1990) to the case where the ontology is described by both a TBox and an ABox.
- Second, we study update in the context of the *DL-Lite* (Calvanese *et al.* 2005) DL. The main feature of *DL-Lite* is that all reasoning tasks in this logic can be done in polynomial time with respect to the size of the ontology. We prove that *DL-Lite* is closed with respect to instance level update, in the sense that the result of an update is always expressible by a new *DL-Lite* ABox (which may involve variables, see below). This has to be contrasted with the results in (Liu *et al.* 2006), which imply that, if we use more expressive logics, instance level updates are generally not expressible in the logic of the original knowledge base.
- Finally, we provide an algorithm that computes the update over a *DL-Lite* knowledge base. We prove that this algorithm is correct, and we show that it runs in polynomial time with respect to the size of the original knowledge base. To the best of our knowledge, this is the first algorithm for a well-founded approach to ontology update in DLs taking into account both the TBox and the ABox.

The paper is organized as follows. In the first two sections, we provide a general overview of DLs and *DL-Lite*, respectively. Then, we present the general framework for instance level update of DL ontologies, by specifying, in particular, the formal semantics of update. Then, we address the issue of update in the context of *DL-Lite*: we describe the algorithm for computing updates in *DL-Lite*, and we discuss its formal and computational properties. We conclude the paper by discussing interesting open problems related to ontology update.

Description Logic ontologies

Description Logics (DLs) (Baader *et al.* 2003) are knowledge representation formalisms that are tailored for representing the domain of interest in terms of *concepts* (or classes), which denote sets of objects, and *roles* (or relations), which denote denote binary relations between objects. DLs *knowledge bases* are formed by two distinct parts: the so-called *TBox*, which contains intensional description of the domain of interest; and the so-called *ABox*, which contains extensional information.

When DLs are used to express ontologies (Baader, Horrocks, & Sattler 2005), the TBox is used to express the *intensional level of the ontology*, i.e., the shared conceptualization of the domain of interest, while the ABox is used to represent the *instance level of the ontology*, i.e., the information on actual objects that are instances of the concepts and roles defined at the intensional level. We are going to assume that the intensional level of the ontology is invariant, i.e., it does not change while the ontology is used[2], while the instance-level of the ontology can indeed change as information in it are updated. From a formal point of view, a DL knowledge base is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where:

- $\mathcal{T}$, the *TBox*, is formed by a finite set of *universal assertions*. The precise form of such assertions depends on the specific DL. However, we insist that the TBox actually is able to place constraints on the extensions of the primitive concepts and roles used to describe the domain of interest. This contrasts with the so called acyclic TBoxes, which are used to just introduce abbreviations for complex combinations of primitive concepts and roles.
- $\mathcal{A}$, the *ABox*, is formed by a finite set of *membership assertions* stating that a given object (or pair of objects) is an instance of a concept (or a role). As usual in DLs, we do enforce the open world assumption (and not the closed world assumption, typical of databases), i.e., it may happen that, for an object a and concept C, $\mathcal{K}$ does not imply neither that a is an instance of C, nor that a is not an instance of C; similarly for roles.

We give the semantics of a DL knowledge base in terms of interpretations over a fixed infinite domain Δ of objects. We assume to have a constant for each object in Δ denoting exactly that object. In this way we blur the distinction between constants and objects, so that we can use them interchangeably (with a little abuse of notation) without causing confusion.[3] An interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ consists of a first order structure over Δ, where $\cdot^{\mathcal{I}}$ is the interpretation function, i.e., a function mapping each concept to a subset of Δ and each role to a subset of $\Delta \times \Delta$. We say that $\mathcal{I}$ is a *model of a (TBox or ABox) assertion* α, or also that $\mathcal{I}$ *satisfies a (TBox or ABox) assertion* α, if α is true in $\mathcal{I}$[4]. We say that $\mathcal{I}$ is a *model of the knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, or also that $\mathcal{I}$ satisfies $\mathcal{K}$, if $\mathcal{I}$ is a model of all the assertions in $\mathcal{T}$ and $\mathcal{A}$. Given a set of (TBox or ABox) assertions $\mathcal{S}$, we denote as $Mod(\mathcal{S})$ the set of interpretations that are models of all assertions in $\mathcal{S}$. In particular, the set of *models of a knowledge base* $\mathcal{K}$, denoted as $Mod(\mathcal{K})$, is the set of models of all assertions in $\mathcal{T}$ and $\mathcal{A}$, i.e. $Mod(\mathcal{K}) = Mod(\langle \mathcal{T}, \mathcal{A} \rangle) = Mod(\mathcal{T}) \cap Mod(\mathcal{A})$. A knowledge base $\mathcal{K}$ is *consistent* if $Mod(\mathcal{K}) \neq \emptyset$, i.e. it has at least one model. A knowledge base $\mathcal{K}$ *logically implies* an assertion α, written $\mathcal{K} \models \alpha$, if for every interpretation $\mathcal{I} \in Mod(\mathcal{K})$, $\mathcal{I} \in Mod(\alpha)$, i.e. all the models of $\mathcal{K}$ are also models of α. On the contrary, we say that a knowledge base $\mathcal{K}$ *does not logically imply* an assertion α, written $\mathcal{K} \not\models \alpha$, if there exists at least one model of $\mathcal{K}$ that is not a model of α.

[2] In other words, in this paper we are not considering the so-called "ontology evolution problem".

[3] We use a shared domain of interpretation, i.e., we use the so-called standard names (Levesque & Lakemeyer 2001), to simplify comparison between models needed for updates. In fact, the use of standard names could be avoided, but this would make some of the definitions below clumsier.

[4] Obviously, the notion an assertion α being true in an interpretation $\mathcal{I}$ depends on the particular DL that we are using for expressing the ontology.

The *DL-Lite* Description Logic

The *DL-Lite* family (Calvanese *et al.* 2006; 2005) is a family of DLs that are tailored towards capturing conceptual modeling constructs, while keeping reasoning, including conjunctive query answering, tractable and first-order reducible (i.e, LOGSPACE in data complexity). The member of the *DL-Lite* family that we consider in this paper, which for simplicity we call simply *DL-Lite*, is a slightly extended version of *DL-Lite* presented in (Calvanese *et al.* 2005). More precisely, *DL-Lite* concepts are defined as follows:

$$\begin{array}{rcl} B & ::= & A \mid \exists R \\ C & ::= & B \mid \neg B \\ R & ::= & P \mid P^- \end{array}$$

where A denotes an atomic concept, P an atomic role, B a basic concept, and C a general concept. A basic concept can be either an atomic concept, a concept of the form $\exists P$, i.e. the standard DL construct of unqualified existential quantification on roles, or a concept of the form $\exists P^-$, which involves *inverse roles*. The universal assertions allowed in a *DL-Lite* TBox are of the form $B \sqsubseteq C$, and more precisely of the form

$$\begin{array}{ll} B_1 \sqsubseteq B_2 & \text{inclusion assertion} \\ B_1 \sqsubseteq \neg B_2 & \text{disjointness assertion} \end{array}$$

and of the form

$$(funct R) \quad \text{functionality assertions}$$

An inclusion assertion specifies that each instance of the basic concept B_1 is also an instance of the basic concept B_2, i.e., B_1 is subsumed by B_2. A disjointness assertion specifies that each instance of a basic concept B_1 is not an instance of the basic concept B_2, i.e., B_1 and B_2 are disjoint. Finally, a functionality assertion expresses the (global) functionality of an atomic role, or of the inverse of an atomic role. Note that negation is used in a restricted way, in particular for asserting disjointness of concepts, and disjunction is disallowed. Notably, if we remove either of these limitations, reasoning becomes intractable, see (Calvanese *et al.* 2006).

The membership assertions allowed in a *DL-Lite* ABox are of the form:

$$C(a), \quad R(a,b), \quad C(z) \quad \text{membership assertions}$$

These assertions state, respectively, that the object a is an instance of the general concept C, that the pair of objects (a, b) is an instance of the role R, and that there exists an object, denoted by the *variable* z, that is an instance of the general concept C. Variables are used to express the existence of objects that are instance of concepts, without actually naming the objects. We will show that knowledge of this form may arise as a result of an update. Note that the presence of variables and the possibility of using negation in membership assertions distinguish the version of *DL-Lite* that we consider here from the one considered in (Calvanese *et al.* 2005). It can be shown, but it is out of the scope of this paper, that all computational properties of the version of *DL-Lite* in (Calvanese *et al.* 2005) hold for this version as well.

Given an interpretation $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ the interpretation function $\cdot^{\mathcal{I}}$ interprets the constructs of *DL-Lite* as follows:

$$\begin{array}{ll} A^{\mathcal{I}} \subseteq \Delta & P^{\mathcal{I}} \subseteq \Delta \times \Delta \\ (P^-)^{\mathcal{I}} = \{(c', c) \mid (c, c') \in P^{\mathcal{I}}\} & \\ (\exists R)^{\mathcal{I}} = \{c \mid \exists c'.(c, c') \in R^{\mathcal{I}}\} & (\neg B)^{\mathcal{I}} = \Delta \setminus B^{\mathcal{I}} \end{array}$$

An interpretation $\mathcal{I}$ is a *model* of an inclusion assertion $B_1 \sqsubseteq B_2$ if $B_1^{\mathcal{I}} \subseteq B_2^{\mathcal{I}}$. $\mathcal{I}$ is a model of a disjointness assertion n $B_1 \sqsubseteq \neg B_2$ if $B_1^{\mathcal{I}} \subseteq \neg B_2^{\mathcal{I}}$. $\mathcal{I}$ is a model of a functionality assertion $(funct R)$ if $(c, c') \in R^{\mathcal{I}}$ and $(c, c'') \in R^{\mathcal{I}}$ implies $c' = c''$. $\mathcal{I}$ is a model of a membership assertion $C(a)$ (resp., $R(a, b)$) iff $a \in C^{\mathcal{I}}$ (resp., $(a, b) \in R^{\mathcal{I}}$). Finally, $\mathcal{I}$ is a model of a membership assertion $C(z)$ if there exists an assignment of the variable z to some $c \in \Delta$ such that $c \in C^{\mathcal{I}}$, with the proviso that such an assignment must be the same for every membership assertion involving z.

Example 1 Let us consider a simple DL-Lite ontology describing a basketball players' domain. In particular, the intensional level of the ontology is expressed by means of a *DL-Lite* TBox constituted by the set of assertions: $\{\exists WillPlay \sqsubseteq AvailablePlayer,\ AvailablePlayer \sqsubseteq Player,\ Injured \sqsubseteq \neg AvailablePlayer\}$. This means that someone who will play in a game is an available player, an available player is a player, and someone who is injured cannot be an available player. Consider the instance level of the ontology expressed by means of the *DL-Lite* ABox: $\{WillPlay(\text{“}John\text{”}, \text{“}allstargame06\text{”})\}$. Then, for example we can infer by logical implication that "John" is an available player $AvailablePlayer(\text{“}John\text{”})$, and hence also a player, $Player(\text{“}John\text{”})$; and that moreover "$John$" is not injured, $\neg Injuried(\text{“}John\text{”})$. □

Instance-level ontology update

Several approaches to update have been considered in literature, see, e.g., (Eiter & Gottlob 1992) for a survey. Here, we essentially follow Winslett's approach (Winslett 1988; 1990). The intuition behind such approach is the following. There is an actual state-of-affairs of the world of which however we have only an incomplete description. Such description identifies a (typically infinite) set of models, each corresponding to a state-of-affairs that we consider possible. Among them, there is one model corresponding to the actual state-of-affairs, but we don't know which. Now, when we perform an update we are changing the actual state-of-affairs. However, since we don't really know which of our models corresponds to the actual state-of-affairs we apply the change on every possible model, thus getting a new set of models representing the updated situation. Among them, we do have the model corresponding to performing the update on the actual state-of-affairs, but again we don't know which. As for how we update each model, only the changes that are absolutely required to accommodate what explicitly asserted in the update will be performed.

Observe that this intuition is essentially the one behind most of the research on reasoning about actions. For example this vision is completely shared by Reiter's variant of Situation Calculus (Reiter 2001). See in particular (Scherl

& Levesque 2003) where possible words are considered explicitly and actions act on such words exactly as said above.[5]

Winslett's approach to update is also completely compatible with the proposal in (Liu *et al.* 2006; Baader *et al.* 2005) where updates of a DL ABox without TBox is studied for several expressive DLs. Observe that in those works, since the TBox is not present[6], the intensional level of the ontology is not specified, and updates are only relative to the instance level represented by the ABox.

Here, instead, we do consider the intensional level of the ontology, represented by a TBox, although, as we said above, we insist that such level is invariant. Updates again impact only the instance-level of the ontology, which however changes according to what specified in the update, in a way to keep the universal assertions in the intensional level satisfied.

In order to define formally such a notion of update, we first need to provide some definitions.

Definition 2 (**Containment between interpretations**) Let $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ and $\mathcal{I}' = \langle \Delta, \cdot^{\mathcal{I}'} \rangle$ be two interpretations (over the same alphabet). We say that $\mathcal{I}$ *is contained in* $\mathcal{I}'$, written $\mathcal{I} \subseteq \mathcal{I}'$, iff $\mathcal{I}, \mathcal{I}'$ are such that:

- if $a \in A^{\mathcal{I}}$ then $a \in A^{\mathcal{I}'}$, for every $a \in \Delta$, and atomic concept A;
- if $(a, b) \in R^{\mathcal{I}}$ then $(a, b) \in R^{\mathcal{I}'}$, for every $(a, b) \in \Delta \times \Delta$ and atomic role R.

We say that $\mathcal{I}$ *is properly contained* $\mathcal{I}'$, written $\mathcal{I} \subset \mathcal{I}'$, iff $\mathcal{I} \subseteq \mathcal{I}'$ but $\mathcal{I}' \not\subseteq \mathcal{I}$. □

Definition 3 (**Difference between interpretation**) Let $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ and $\mathcal{I}' = \langle \Delta, \cdot^{\mathcal{I}'} \rangle$ be two interpretations (over the same alphabet). We define the *difference between* $\mathcal{I}$ *and* $\mathcal{I}'$, written $\mathcal{I} \ominus \mathcal{I}'$, as the interpretation $(\Delta, \cdot^{\mathcal{I} \ominus \mathcal{I}'})$ such that:

- $A^{\mathcal{I} \ominus \mathcal{I}'} = A^{\mathcal{I}} \ominus A^{\mathcal{I}'}$, for every atomic concept A;
- $R^{\mathcal{I} \ominus \mathcal{I}'} = R^{\mathcal{I}} \ominus R^{\mathcal{I}'}$, for every atomic role R.

where $S \ominus S'$ denotes the symmetric difference between sets S and S', i.e. $S \ominus S' = (S \cup S') \setminus (S \cap S')$. □

With this notion in place we can now define updates.

Definition 4 (**Model update**) Let $\mathcal{T}$ be a TBox in a DL $\mathcal{L}$, $\mathcal{I}$ a model of $\mathcal{T}$, and $\mathcal{F}$ a finite set of membership assertions expressed in $\mathcal{L}$ such that $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \neq \emptyset$. We define the *(result of) the update of* $\mathcal{I}$ *with* $\mathcal{F}$, denoted by $U^{\mathcal{T}}(\mathcal{I}, \mathcal{F})$, as follows:

$$U^{\mathcal{T}}(\mathcal{I}, \mathcal{F}) = \{\mathcal{I}' \mid \mathcal{I}' \in Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \text{ and there exists no } \mathcal{I}'' \in Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \text{ s.t. } \mathcal{I} \ominus \mathcal{I}'' \subset \mathcal{I} \ominus \mathcal{I}'\}$$ □

[5] Actually (Scherl & Levesque 2003) studies also "knowledge producing actions" (i.e., sensing actions), which are more related with belief revision than update.

[6] Or, if present, it is assumed to be acyclic. Acyclic TBoxes cannot be used to model the intensional level of an ontology, since the abbreviations that they introduce can be eliminated without semantic loss. Naturally, since acyclic TBoxes may provide compact representation of complex concepts, they may have an impact on the computational complexity of reasoning.

Observe that $U^{\mathcal{T}}(\mathcal{I}, \mathcal{F})$ is the set of models of $\mathcal{T}$ and $\mathcal{F}$ whose difference w.r.t. $\mathcal{I}$ is $\subseteq$-minimal, and that such a set is non-empty.

Definition 5 (**Update**) Let $\mathcal{T}$ be TBox expressed in a DL $\mathcal{L}$, $\mathcal{M} \subseteq Mod(\mathcal{T})$ a set of models of $\mathcal{T}$ and $\mathcal{F}$ a finite set of membership assertions expressed in $\mathcal{L}$ such that $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \neq \emptyset$. We define the *(result of the) update* of $\mathcal{M}$ with $\mathcal{F}$, denoted as $\mathcal{M} \circ_{\mathcal{T}} \mathcal{F}$, as the following set of models: $\mathcal{M} \circ_{\mathcal{T}} \mathcal{F} = \bigcup_{\mathcal{I} \in \mathcal{M}} U^{\mathcal{T}}(\mathcal{I}, \mathcal{F})$. □

Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base in $\mathcal{L}$ and $\mathcal{F}$ a finite set of membership assertions expressed in $\mathcal{L}$ such that $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \neq \emptyset$. With a little abuse of notation and terminology we will write $\mathcal{K} \circ_{\mathcal{T}} \mathcal{F}$ to denote $Mod(\mathcal{K}) \circ_{\mathcal{T}} \mathcal{F}$ and talk about the update of $\mathcal{K}$ instead of talking about the update of the models $Mod(\mathcal{K})$ of $\mathcal{K}$.

A basic question arises from such definitions. Is the result of updating a knowledge base still expressible as a new knowledge base in the same DL? Let us introduce the following definition.

Definition 6 (**Expressible update**) Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be knowledge base expressed in a DL $\mathcal{L}$ and $\mathcal{F}$ a set of membership assertions expressed in $\mathcal{L}$ such that $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \neq \emptyset$. We say that the *update of* $\mathcal{K}$ *with* $\mathcal{F}$ *is expressible in* $\mathcal{L}$ iff there exists an ABox $\mathcal{A}'$ expressed in $\mathcal{L}$ such that $\mathcal{K} \circ_{\mathcal{T}} \mathcal{F} = Mod(\langle \mathcal{T}, \mathcal{A}' \rangle)$ □

The results in (Liu *et al.* 2006) show that, for several quite standard DLs, updates are not expressible in the original language of the knowledge base, even if TBoxes are not considered. Instead, in the case of *DL-Lite* we have the notable property that updates are always expressible in *DL-Lite* itself, as we show in the next section.

Computing updates in *DL-Lite*

We address instance-level updates in *DL-Lite*. In particular:

- we show that the result of an update is always expressible within *DL-Lite*: i.e., there always exist a new *DL-Lite* ABox that reflects the changes of the update to the original knowledge base (obviously the TBox remains unchanged as required);
- we show that the new ABox resulting from an update can be automatically computed;
- finally, we show that the size of such an ABox is polynomially bounded by the original knowledge base, and moreover that it can be computed in polynomial time.

Before starting the technical development we illustrate the update on an example to gain some intuition on the problem.

Example 7 Consider the ontology presented in Example 1. Now suppose that *John* gets injured: we update the ontology with the membership assertion *Injured("John")*. Based on the semantics presented above, the result of the update can be expressed by the following ABox: {*Injured("John"), Player("John")*}. Note that the new instance level reflects that *John* is a player who is actually injured. Interestingly, the fact that *John* is injured implies that he is not an available player anymore and therefore will not play *"allstargame06"*, Nevertheless, he remains a player, and this would not be captured by simply removing the ABox assertions that are inconsistent with the update. □

INPUT: finite set of ground membership assertions $\mathcal{F}$, satisfiable *DL-Lite*-KB $\langle \mathcal{T}, \mathcal{A} \rangle$
OUTPUT: an ABox $\mathcal{A}'$, or ERROR
[1] if $\langle \mathcal{T}, \mathcal{F} \rangle$ is not satisfiable **then** ERROR
[2] else for each $F \in \mathcal{F}$ **do**
[3] **if** $F = R(a, b)$ **then** $\mathcal{F} := \mathcal{F} \cup \{\exists R(a), \exists R^-(b)\}$
[4] $\mathcal{A}' := \mathcal{A} \cup \mathcal{F}$; $\mathcal{F}' := \emptyset$
[5] **for each** $F_1 \in \mathcal{F}$ **do**
[6] **if** $F_1 = C(a)$ **then**
[7] **for each** $F' \in Expand(\neg C(a), \mathcal{T})$ **do**
[8] **if** $F' = C'(a)$ and $\langle \mathcal{T}, \mathcal{A} \rangle \models C'(a)$ **then**
[9] $\mathcal{F}' := \mathcal{F}' \cup \{C'(a)\}$
[10] **else if** $F' = \exists R'(a)$ **then**
[11] $\mathcal{F}' := \mathcal{F}' \cup \{R'(a, b) \mid R'(a, b) \in \mathcal{A}\}$
[12] **else if** $F_1 = R(a, b)$ **then**
[13] **if** $(funct R)$ in $\mathcal{T}$ **then**
[14] **for each** $b' \neq b$ s.t. $\langle \mathcal{T}, \mathcal{A} \rangle \models R(a, b')$ **do**
[15] $\mathcal{F}' := \mathcal{F}' \cup \{R(a, b')\}$
[16] **if** $(funct R^-)$ in $\mathcal{T}$ **then**
[17] **for each** $a' \neq a$ s.t. $\langle \mathcal{T}, \mathcal{A} \rangle \models R(a', b)$ **do**
[18] $\mathcal{F}' := \mathcal{F}' \cup \{R(a', b)\}$
[19] **for each** $F' \in \mathcal{F}'$ **do**
[20] **if** $F' = C'(a)$ **then**
[21] $\mathcal{A}' := \mathcal{A}' \setminus \{C'(a)\}$
[22] **for each** $C' \sqsubseteq C_1$ in $cl(\mathcal{T})$ **do**
[23] **if** $(C_1(a) \notin \mathcal{F}')$ **then** $\mathcal{A}' := \mathcal{A}' \cup \{C_1(a)\}$
[24] **if** $F' = \exists R(a)$ **then**
[25] **for each** $\exists R^- \sqsubseteq C_2$ in $cl(\mathcal{T})$ **do**
[26] $\mathcal{A}' := \mathcal{A}' \cup \{C_2(z)\}$, with z new variable
[27] **else if** $F' = R(a, b)$ **then**
[28] $\mathcal{A}' := \mathcal{A}' \setminus \{R(a, b), \exists R(a), \exists R^-(b)\}$
[29] **for each** $\exists R \sqsubseteq C_3$ in $cl(\mathcal{T})$ **do**
[30] **if** $C_3(a) \notin \mathcal{F}'$ **then** $\mathcal{A}' := \mathcal{A}' \cup \{C_3(a)\}$
[31] **for each** $\exists R^- \sqsubseteq C_4$ in $cl(\mathcal{T})$ **do**
[32] **if** $C_4(b) \notin \mathcal{F}'$ **then** $\mathcal{A}' := \mathcal{A}' \cup \{C_4(b)\}$

Figure 1: Algorithm $ComputeUpdate(\mathcal{T}, \mathcal{A}, \mathcal{F})$

In Fig 1, we provide an algorithm to perform an update over a *DL-Lite* knowledge base. To simplify the presentation we make use of the following notation. First we denote by R^- the inverse of R, i.e., if R is an atomic role, then R^- is its inverse, while if R is the inverse of an atomic role, then R is the atomic role itself. Second, we write $\neg C$ to denote $\neg B$ if C is B, and B if C is $\neg B$. Also, we use the notation $C_1 \sqsubseteq C_2$ to denote either assertions of the form $B_1 \sqsubseteq B_2$, $B_1 \sqsubseteq \neg B_2$, or $\neg B_1 \sqsubseteq \neg B_2$. Finally we denote by $cl(\mathcal{T})$ the deductive closure of $\mathcal{T}$ i.e., $\{C_1 \sqsubseteq C_2 \mid \langle \mathcal{T}, \mathcal{A} \rangle \models C_1 \sqsubseteq C_2\}$. It can be shown that in *DL-Lite*, $cl(T)$ can be computed in polynomial time by applying recursively the following rules: *(i)* $\mathcal{T} \subseteq cl(\mathcal{T})$; *(ii)* if $C_1 \sqsubseteq C_2$ in $cl(\mathcal{T})$ then $\neg C_2 \sqsubseteq \neg C_1$ in $cl(\mathcal{T})$; *(iii)* if $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_3$ in $cl(\mathcal{T})$, then $C_1 \sqsubseteq C_3$ in $cl(\mathcal{T})$.

The algorithm in Fig. 1 takes as input a *DL-Lite* satisfiable knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, and a finite set of ground (i.e., not involving variables) membership assertions $\mathcal{F}$, and returns either ERROR (if $\langle \mathcal{T}, \mathcal{F} \rangle$ is unsatisfiable), or an ABox $\mathcal{A}'$ (otherwise). Roughly speaking, the algorithm proceeds as follows. After a satisfiability check, it inserts into $\mathcal{A}'$ all the membership assertions in $\mathcal{A}$ and $\mathcal{F}$ (lines 3–4), and then uses the algorithm shown in Fig. 2 to compute the set $\mathcal{F}'$ of membership assertions that are logically implied by $\mathcal{K}$ and contradict $\mathcal{F}$ according to $\mathcal{T}$ (lines 5–18). Finally, for each $F' \in \mathcal{F}'$, the algorithm deletes F' from $\mathcal{A}'$, but inserts into

INPUT: membership assertion $C(a)$, TBox $\mathcal{T}$
OUTPUT: set $\mathcal{P}$ of ground membership assertions
$\mathcal{P} := \{C(a)\}$
repeat
$\mathcal{P}' = \mathcal{P}$
for each $C(a) \in \mathcal{P}'$
if $C' \sqsubseteq C \in cl(\mathcal{T})$ **then** $\mathcal{P} := \mathcal{P} \cup \{C'(a)\}$
until $\mathcal{P} = \mathcal{P}'$

Figure 2: Algorithm $Expand(C(a), \mathcal{T})$

$\mathcal{A}'$ those membership assertions that are logically implied by the membership assertions deleted and do not contradict $\mathcal{F}$ (lines 19–32).

Next, we deal with termination, soundness and completeness of the algorithm shown in Fig. 1.

Lemma 8 *(Termination) Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite knowledge base, $\mathcal{F}$ a finite set of ground DL-Lite membership assertions. Then the algorithm $ComputeUpdate(\mathcal{T}, \mathcal{A}, \mathcal{F})$ terminates, returning ERROR if $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) = \emptyset$, and an ABox $\mathcal{A}'$ such that $\langle \mathcal{T}, \mathcal{A}' \rangle$ is a DL-Lite knowledge base, otherwise.*

Lemma 9 *(Soundness) Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite knowledge base, $\mathcal{F}$ a finite set of ground DL-Lite membership assertions such that $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \neq \emptyset$, and $\mathcal{A}'$ the ABox returned by $ComputeUpdate(\mathcal{T}, \mathcal{A}, \mathcal{F})$. Then, for every model $\mathcal{I}' \in Mod(\langle \mathcal{T}, A' \rangle)$, we have that $\exists \mathcal{I} \in Mod(\mathcal{K})$ s.t. $\mathcal{I}' \in U^{\mathcal{T}}(\mathcal{I}, \mathcal{F})$.*

Lemma 10 *(Completeness) Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite knowledge base, $\mathcal{F}$ a finite set of ground DL-Lite membership assertions such that $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \neq \emptyset$, and $\mathcal{A}'$ the ABox returned by $ComputeUpdate(\mathcal{T}, \mathcal{A}, \mathcal{F})$. Then, for every model $\mathcal{I} \in Mod(\mathcal{K})$, we have that $U^{\mathcal{T}}(\mathcal{I}, \mathcal{F}) \subseteq Mod(\langle \mathcal{T}, A' \rangle)$.*

From the two lemmas above, we get the following theorem, that sanctions the correctness of our algorithm.

Theorem 11 *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite knowledge base $\mathcal{F}$ a finite set of ground DL-Lite membership assertions such that $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \neq \emptyset$, and $\mathcal{A}' = ComputeUpdate(\mathcal{T}, \mathcal{A}, \mathcal{F})$. Then $\mathcal{K} \circ_{\mathcal{T}} \mathcal{F} \equiv Mod(\langle \mathcal{T}, A' \rangle)$.*

Interestingly, if we do not allow for membership assertions involving variables in the ABox, then we lose expressibility, as shown by the following example.

Example 12 The TBox $\{\exists P^- \sqsubseteq A_1, A_2 \sqsubseteq \neg \exists P\}$ and the ABox $\{\exists P(a)\}$ imply that there exists an object that is both a P-successor of a, and an instance of A_1. Now let us consider the update $\{A_2(a)\}$. As a result of the update, $A_2(a)$ must be logically implied, hence we must remove $\exists P(a)$ from the ABox, but the fact that there is an instance of A_1 must remain logically implied after the update. It can be easily seen that to express this in the new ABox we must use $A_1(z)$ where z is a new variable. □

Next we turn to the computational complexity of computing the update. By analyzing the algorithm we get:

Theorem 13 *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite knowledge base, $\mathcal{F}$ a finite set of ground DL-Lite membership assertions such that $Mod(\mathcal{T}) \cap Mod(\mathcal{F}) \neq \emptyset$, and $\mathcal{A}' = ComputeUpdate(\mathcal{T}, \mathcal{A}, \mathcal{F})$. Then:*

- *the size of $\mathcal{A}'$ is polynomially bounded by the size of $\mathcal{T} \cup \mathcal{A} \cup \mathcal{F}$;*
- *computing $\mathcal{A}'$ can be done in polynomial time in the size of $\mathcal{T} \cup \mathcal{A} \cup \mathcal{F}$.*

Conclusion

In this paper we have studied the problem of updating ontologies at the extensional level, under the assumption that the intensional level of the ontology is immutable. We have considered the case of ontologies represented as DL knowledge bases, by looking in particular at *DL-Lite*. Our results show that such a DL has several nice characteristics w.r.t. updates, including the fact that the result of an update is always expressible within *DL-Lite* itself.

There are several interesting directions for continuing our research. First, the expressibility property previously mentioned is missing in many well-known DLs such as $\mathcal{ALC}$ (Liu *et al.* 2006) or certain versions of *DL-Lite* (see above). Note, however, that even if the result of the update is not expressible in a DL, we might be interested in reasoning, e.g., answering queries, on the ontology resulting from the update. Indeed, to do so we do not necessarily need to "materialize" the new knowledge base, but actually we could reason on the original knowledge base by taking into account the update in a "virtual" way. In a sense, this is analogous to the distinction between projection via regression vs. progression in reasoning about actions (Reiter 2001).

Secondly, in this paper we adopted a classical model-based approach to update, stemming from the existing literature on updating knowledge bases. Other approaches to update have been studied and their application to ontology might be of interest, as well as approaches based on belief revision (see e.g. (Eiter & Gottlob 1992) as a starting point). We believe that, in principle, several approaches to update and belief revision could coexist on the same ontology management system, in order to model different types of services involving some sort of ontology evolution.

Finally, it is worth noting that updates bring in the general issue of dealing with inconsistency in ontologies. The semantics that we have considered in this paper addresses the issue of solving inconsistency between the current instance level of the ontology and what has been asserted by the update, while it does not deal with inconsistencies between the update and the intensional level. It would be interesting to study possible semantics that are tolerant w.r.t. the latter form of inconsistency.

Acknowledgments This research has been partially supported by the FET project TONES: Thinking ONtologiES, funded by the EU under contract number FP6-7603, the project HYPER, funded by IBM through a Shared University Research (SUR) Award grant, and by MIUR FIRB 2005 project "Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet" (TOCAI.IT).

References

Acciarri, A.; Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Palmieri, M.; and Rosati, R. 2005. QUONTO: QUerying ONTOlogies. In *Proc. of AAAI 2005*, 1670–1671.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Baader, F.; Lutz, C.; Milicic, M.; Sattler, U.; and Wolter, F. 2005. Integrating description logics and action formalisms: First results. In *Proc. of AAAI 2005*, 572–577.

Baader, F.; Horrocks, I.; and Sattler, U. 2005. Description logics as ontology languages for the semantic web. In *Mechanizing Mathematical Reasoning: Essays in Honor of Jrg Siekmann on the Occasion of His 60th Birthday*, number 2605 in LNAI. Springer. 228–248.

Berners Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web. *Scientific American*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*, 602–607.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2006. Data complexity of query answering in description logics. In *Proc. of KR 2006*. To appear.

Eiter, T., and Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artificial Intelligence* 57:227–270.

Flouris, G.; Plexousakis, D.; and Antoniou, G. 2005. On applying the AGM theory to DLs and OWL. In *Proc. of ISWC 2005*.

Haarslev, V., and Möller, R. 2001. RACER system description. In *Proc. of IJCAR 2001*, volume 2083 of *LNAI*, 701–705. Springer.

Haase, P., and Stojanovic, L. 2005. Consistent evolution of owl ontologies. In *Proc. of ESWC*, 182–197.

Horrocks, I. 1998. The FaCT system. In de Swart, H., ed., *Proc. of TABLEAUX'98*, volume 1397 of *LNAI*, 307–312. Springer.

Levesque, H. J., and Lakemeyer, G. 2001. *The Logic of Knowledge Bases*. The MIT Press.

Liu, H.; Lutz, C.; Milicic, M.; and Wolter, F. 2006. Updating description logic aboxes. In *Proc. of KR 2006*. To appear.

Meyer, T.; Lee, K.; and Booth, R. 2005. Knowledge integration for description logics. In *Proc. of AAAI 2005*, 645–650.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press.

Scherl, R. B., and Levesque, H. J. 2003. Knowledge, action, and the frame problem. *Artificial Intelligence* 144(1-2):1–39.

Sirin, E., and Parsia, B. 2004. Pellet: An owl dl reasoner. In *Proc. of DL 2004*.

Winslett, M. 1988. Reasoning about action using a possible models approach. In *Proc. of AAAI'98*.

Winslett, M. 1990. *Updating Logical Databases*. Cambridge University Press.