ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-516

Updating inconsistent Description Logic knowledge bases

Maurizio Lenzerini, Domenico Fabio Savo[1]

Abstract. Finding an appropriate semantics for task of updating an inconsistent knowledge base is a challenging problem. In this paper, we consider knowledge bases expressed in Description Logics, and focus on ABox inconsistencies, i.e., the case where the TBox is consistent, but the whole knowledge base is not. Our first contribution is the definition of a new semantics for updating an inconsistent Description Logic knowledge base with both the insertion and the deletions of a set of ABox assertions. We then concentrate on the *DL-Lite* family of Description Logics, and present algorithms for updating a possibly inconsistent knowledge base expressed in the most expressive logic of such family. We show that, by virtue of both the characteristics of our semantics, and the limited expressive power of *DL-Lite*, both insertions and deletions can be done in polynomial time with respect of the size of the ABox.

1 Introduction

It is well known that inconsistency is ubiquitous in many KBs used in real world applications. It follows that a KB management system should be equipped with suitable mechanisms for inconsistency tolerance, aiming at addressing at least the following two issues: (i) How to answer queries that are posed to an inconsistent KB (inconsistency-tolerant query answering); (ii) How to compute the KB resulting from updating a possibly inconsistent KB with the assertion of new assertions (inconsistent-tolerant update).

In this paper, we study inconsistency tolerance in the context of Description Logic (DL) KBs. A KB consists of two components: a TBox, i.e., a set of intensional assertions, and an ABox, i.e., a set of extensional assertions. In our work, we are interested in DL KBs where the TBox is consistent, and inconsistencies may arise from the interaction between the TBox and the ABox. In other words, we deal with ABox inconsistencies, and leave the general case, i.e., when the TBox itself may be inconsistent, to future investigations. ABox inconsistency is common in the context, for example, of Ontology-based Data Access [14, 4] (OBDA), where the TBox is usually a high quality representation of the domain, designed in such a way to avoid inconsistencies in the modeling of concepts and relationships. On the contrary, the ABox derives from data sources which are independent on the conceptualization represented by the TBox, and therefore may contain data which are not coherent with it.

Inconsistency-tolerant query answering has received considerable attention in the last years. Generally speaking, there are many approaches for devising inconsistency-tolerant inference systems [1], originated in different areas, including Artificial Intelligence, and Databases. As we said before, the present paper deals with ABox inconsistency in DL KBs. This scenario has many similarities with the one addressed by the approaches to consistent query answering in databases [1]. These are based on the idea of living with inconsistencies (i.e., data that do not satisfy the integrity constraints) in the database, but trying to obtain only consistent information during query answering. The main tool used for this purpose is the notion of database repair: a *repair* of a database contradicting a set of integrity constraints is a database obtained by applying a minimal set of changes which restore consistency. In general, there are many possible repairs for a database, and inconsistency-tolerant query answering amounts to compute the tuples that are answers to the query in all possible repairs. Recent papers apply the notion of repair to the context of DL KBs, and study different inconsistency-tolerant semantics for such KBs [11, 15, 2]. As we will see, we will refer to such a notion in our work.

Inconsistency-tolerant update in the context of DL KBs is, to our knowledge, a new problem. Indeed, all the approaches to update in DLs that we are aware of, assume that the evolving KB is consistent, and the update itself preserves consistency. Updating a KB means to modify it in order to adhere to a change in the domain of interest (see [10, 16] for other kinds of KB evolution operations, such as revision). The modification may concern either the insertion or the deletion of assertions. In virtually all the approaches to update, some minimality criterion is used in the modifications of the KB that must be undertaken to realize the evolution operations. In other words, the need is commonly perceived of keeping the distance between the original KB and the KB resulting from the application of an evolution operator minimal. There are two main approaches to define such a distance, called model-based (see, e.g., [8, 13]) and formula-based (See [7], also for a comparison of the two approaches in the context of DL KBs). We base our approach on recent papers on updating DL KBs using the formula-based approach, namely [7, 12]. In [7], the authors present two formula-based approaches for updating *DL-Lite* ABoxes with the insertion of new information: the *bold semantics* and the *careful semantics*. The latter aims at solving the problem of unexpected information coming from the update. The *bold semantics* is close to the semantics presented in [12], but only in the case when only one ABox accomplishing the insertion of new information exists. Indeed, we already noted that the formula constituting the result of an evolution operation is not unique in general. While [9] essentially proposes to keep the whole set of such formulas, and [7] relies on a nondeterministic choice, [12] takes a radical approach, and consider their intersection as the result of the evolution, thus following the *When In Doubt Throw It Out* (WIDTIO) [17] principle.

Building specifically on [12], in this paper we present two main contributions. The first contribution is a semantics of updating inconsistent DL KBs with both the insertion and the deletion of a set of ABox assertions. Our update mechanism is based on the idea that realizing an insertion into (resp., deletion from) a KB $\mathcal{K}$ of a set F of ABox assertions means computing the (possibly inconsistent) KB

[1] Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti, SAPIENZA Università di Roma, email: *lenzerini,savo@dis.uniroma1.it*

$\mathcal{K}'$ that minimally differ from $\mathcal{K}$ and such that all the repairs of $\mathcal{K}'$ entails (resp., do not entail) F. Based on the WIDTIO principle, the result of the update is the intersection of all the KBs realizing the update.

The second contribution is the design of two algorithms, one for the update by insertion, and the other for the update by deletion for KBs expressed in the DL *DL-Lite*$_{A,id}$, which is the most expressive logic in the *DL-Lite* family [5]. The *DL-Lite* family [2] has been specifically designed to keep all reasoning tasks polynomially tractable in the size of the ABox. By characterizing the computational complexity of the insertion and the deletion algorithms for *DL-Lite*$_{A,id}$, we show that this property still holds for inconsistency-tolerant update.

The paper is organized as follows. In Section 2 we recall the basic notions of DL KBs and *DL-Lite*$_{A,id}$. Section 3 illustrates our inconsistency-tolerant update semantics. Section 4 presents the algorithms for update by insertion in *DL-Lite*$_{A,id}$, and Section 5 does the same for update by deletion. Section 6 concludes the paper.

2 Preliminaries

In this section, we define some preliminary notions used in the rest of the article.

Description logic knowledge bases. Description Logics (DLs) are logics that allow one to represent the domain of interests in terms of *concepts*, denoting sets of objects, *value-domain*, denoting sets of values, *attributes*, denoting binary relations between objects and values, and *roles* denoting binary relations over objects.

Let $\mathcal{S}$ be a signature of symbols for individual (object and value) constants, and atomic elements, i.e., concepts, value-domains, attributes, and roles. If $\mathcal{L}$ is a DL, then a knowledge base (KB) in $\mathcal{L}$ (or, $\mathcal{L}$-KB) over $\mathcal{S}$ is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$, called *TBox*, is a finite set of intensional assertions over $\mathcal{S}$ expressed in $\mathcal{L}$, and $\mathcal{A}$, called *ABox*, is a finite set of instance assertions, i.e, assertions on individuals, over $\mathcal{S}$. Different DLs allow for different kinds of TBox and/or ABox assertions. In this paper we assume that ABox assertions are always *atomic*, i.e., they correspond to ground atoms, and therefore we omit to refer to $\mathcal{L}$ when we talk about ABox assertions.

In what follows, when we use the term DL KB (or, simply, KB), we mean an $\mathcal{L}$-KB, for some DL $\mathcal{L}$. Also, for simplicity, we always refer to a fixed signature $\mathcal{S}$.

The semantics of a KB is given in terms of first-order interpretations over $\mathcal{S}$. An interpretation $\mathcal{I}$ is a *model* of a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if it satisfies all assertions in $\mathcal{T} \cup \mathcal{A}$, where the notion of satisfaction depends on the constructs and axioms allowed by the specific DL in which $\mathcal{K}$ is expressed. We say that $\mathcal{T}$ is *satisfiable* if there exists at least one model of all the assertions in $\mathcal{T}$. We denote the set of models of $\mathcal{K}$ with $Mod(\mathcal{K})$, and we say that $\mathcal{K}$ is satisfiable or consistent if $Mod(\mathcal{K}) \neq \emptyset$, unsatisfiable or inconsistent otherwise.

Let $\mathcal{T}$ be a TBox in $\mathcal{L}$, and let $\mathcal{A}$ be an ABox. We say that $\mathcal{A}$ is $\mathcal{T}$*-consistent* if $\langle \mathcal{T}, \mathcal{A} \rangle$ is satisfiable, $\mathcal{T}$-inconsistent otherwise. More over, let V be a set of ABox assertions. We say that V is a $\mathcal{T}$*-inconsistent set* if V is $\mathcal{T}$-inconsistent. As usual, a KB $\mathcal{K}$ entails a closed first-order logic (FOL) sentence ϕ, denoted $\mathcal{K} \models \phi$, if $\phi^{\mathcal{I}}$ is true in every $\mathcal{I} \in Mod(\mathcal{K})$. In what follows, if F is a set of closed formulas, then we write $\mathcal{K} \models F$ to mean that every formula in F is entailed by $\mathcal{K}$.

The $\mathcal{T}$*-closure* of $\mathcal{A}$ with respect to $\mathcal{T}$, denoted $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$, is the set of all atomic ABox assertions formed with individuals appearing in $\mathcal{K}$, and are logically implied by $\langle \mathcal{T}, \mathcal{A} \rangle$. Note that, if $\langle \mathcal{T}, \mathcal{A} \rangle$ is an $\mathcal{L}$-KB, then $\langle \mathcal{T}, \mathsf{cl}_{\mathcal{T}}(\mathcal{A}) \rangle$ is an $\mathcal{L}$-KB as well, and, obviously, $\langle \mathcal{T}, \mathcal{A} \rangle$ is logically equivalent to $\langle \mathcal{T}, \mathsf{cl}_{\mathcal{T}}(\mathcal{A}) \rangle$, i.e., $Mod(\langle \mathcal{T}, \mathcal{A} \rangle) = Mod(\langle \mathcal{T}, \mathsf{cl}_{\mathcal{T}}(\mathcal{A}) \rangle)$. $\mathcal{A}$ is said to be $\mathcal{T}$*-closed* if $\mathsf{cl}_{\mathcal{T}}(\mathcal{A}) = \mathcal{A}$.

Given a query $q(\vec{x})$ (either a conjunctive query or an union of conjunctive queries) and a knowledge base $\mathcal{K}$, the *certain answers* to $q(\vec{x})$ over $\mathcal{K}$ is the set $ans(q, \mathcal{K})$ of all tuples $\vec{t}$ of constants appearing in $\mathcal{K}$, such that, when substituted to the variables $\vec{x}$ in $q(\vec{x})$, we have that $\mathcal{K} \models q(\vec{t})$, i.e., such that $\vec{t}^{\mathcal{I}} \in q^{\mathcal{I}}$ for every $\mathcal{I} \in Mod(\mathcal{K})$. Notice that, since $\mathcal{K}$ is finite, $ans(q, \mathcal{K})$ is finite by definition.

The description logic *DL-Lite*$_{A,id}$. The *DL-Lite* family [5] is a family of low complexity DLs particularly suited for dealing with KBs with very large ABoxes. In this paper, we refer to *DL-Lite*$_{A,id}$, which is the most expressive logic in the family. Due to the lack of space, we do not describe this logic here. The reader is referred to [6] for an account of the syntax of such logic.

Let $\mathcal{T}$ be a *DL-Lite*$_{A,id}$-KB. The set of positive (resp., negative) inclusions in $\mathcal{T}$ will be denoted by $\mathcal{T}^+$ (resp., $\mathcal{T}^-$), whereas the set of functionality assertions (resp., identification assertions) in $\mathcal{T}$ will be denoted by $\mathcal{T}_f$ (resp., $\mathcal{T}_{id}$).

Example 1 The following TBox $\mathcal{T}$ is a portion of a knowledge base describing the domain of rowing competitions.

$OA \sqsubseteq ATH$	$CX \sqsubseteq ATH$	$CX \sqsubseteq \neg OA$
$ATH \sqsubseteq \exists mf$	$CH \sqsubseteq \exists mf$	$CH \sqsubseteq \neg ATH$
$\exists mf^- \sqsubseteq RTM$	$CR \sqsubseteq \exists fb$	$\exists fb \sqsubseteq CR$
$\exists fb^- \sqsubseteq ATH$	$(funct\ mf)$	$(funct\ fb^-)$
$(id\ CX\ fb^-)$		

The axioms state that oars (OA) and coxs (CX) are both athletes (ATH), and oars are not coxs. Every athlete is member of (mf) exactly one rowing team (RTM). A crew (CR) is formed by (fb) athletes, among which there is exactly one cox. Finally, a coach (CH) is a member of exactly one rowing team, and is not an athlete.

Let us consider the ABox $\mathcal{A}$ containing the following ABox assertions:

$$\mathcal{A} = \{\ OA(o), mf(o, t_1), fb(w, o), CX(c), CH(h), mf(c, t_1), mf(h, t_2)\ \}.$$

In words, $\mathcal{A}$ specifies that o is an oar, c is a coax, and both are members of the rowing team t_1. Moreover, coach h is a member of the rowing team t_2, and crew w is formed by o. □

We conclude this section with a brief discussion on the complexity of reasoning about a *DL-Lite*$_{A,id}$-KB $\langle \mathcal{T}, \mathcal{A} \rangle$. If $\langle \mathcal{T}, \mathcal{A} \rangle$ is a satisfiable *DL-Lite*$_{A,id}$-KB, then the evaluation of a UCQ posed to $\langle \mathcal{T}, \mathcal{A} \rangle$ can be reduced to standard evaluation of a FOL query, and hence in AC^0 in the size of $\mathcal{A}$ (data complexity), and in P in the size of the whole KB. As for query complexity, answering UCQs in *DL-Lite*$_{A,id}$ is in NP with respect to the size of the query [5, 6]. Moreover, satisfiability can be checked in polynomial time with respect to $|\mathcal{T}|$ and in AC^0 with respect to $|\mathcal{A}|$. Finally, $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ can be computed in quadratic time with respect to $|\mathcal{T}|$ and $|\mathcal{A}|$.

3 Inconsistency-tolerant update

In this section we present our semantics for updating possibly inconsistent DL KBs with both the insertion and the deletion of a finite set of ABox assertions. In what follows, $\mathcal{L}$ is a DL, and $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a

[2] Not to be confused with set of DLs studied in A. Artale, et al, 'The *DL-Lite* family and relations', *J. of Artificial Intelligence Research*, **36**, 1–69, (2009), which form the *DL-Lite*$_{bool}$ family.

possibly inconsistent $\mathcal{L}$-KB, with $\mathcal{T}$ satisfiable. Moreover, F denotes a finite set of $\mathcal{T}$-consistent ABox assertions. In the rest of this paper, we use the term "update" as a generalization of "update by insertion" and "update by deletion".

We follow the idea in [12] for updating consistent KBs, that we now briefly recall.

Firstly, we need to introduce the notion of "few changes" introduced in [9]. Let $\mathcal{A}$, $\mathcal{A}'$, and $\mathcal{A}''$ be three finite set of ABox assertions. We say that $\mathcal{A}'$ has fewer deletions than $\mathcal{A}''$ with respect to $\mathcal{A}$ if $\mathcal{A}\setminus\mathcal{A}' \subset \mathcal{A}\setminus\mathcal{A}''$. Also, we say that $\mathcal{A}'$ and $\mathcal{A}''$ have the same deletions with respect to $\mathcal{A}$ if $\mathcal{A}\setminus\mathcal{A}' = \mathcal{A}\setminus\mathcal{A}''$. Finally, we say that $\mathcal{A}'$ has fewer insertions than $\mathcal{A}''$ with respect to $\mathcal{A}$ if $\mathcal{A}'\setminus\mathcal{A} \subset \mathcal{A}''\setminus\mathcal{A}$.

Definition 1 Let $\mathcal{A}$, $\mathcal{A}_1$, and $\mathcal{A}_2$ be three finite sets of ABox assertions. Then, $\mathcal{A}_1$ has *fewer changes* than $\mathcal{A}_2$ with respect to $\mathcal{A}$ if

(i) $\mathcal{A}_1$ has fewer deletions than $\mathcal{A}_2$ with respect to $\mathcal{A}$, or
(ii) $\mathcal{A}_2$ have the same deletions with respect to $\mathcal{A}$, and $\mathcal{A}_1$ has fewer insertions than $\mathcal{A}_2$ with respect to $\mathcal{A}$.

Now, suppose that $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ is consistent, and we want to update $\mathcal{K}$ with either the insertion or the deletion of F. Essentially, in the case of insertion, the result of the update is the KB formed by $\mathcal{T}$ and the intersection of all the ABoxes accomplishing the insertion of F into $\mathcal{K}$ minimally. Similarly, the result of updating $\mathcal{K}$ with the deletion of F is the KB formed by $\mathcal{T}$ and the intersection of all the ABoxes accomplishing the deletion of F from $\mathcal{K}$ minimally. An ABox $\mathcal{A}'$ accomplishes the insertion (resp., deletion) of F minimally if $\mathcal{A}'$ is $\mathcal{T}$-consistent, $\langle\mathcal{T},\mathcal{A}'\rangle$ logically entails F (resp., does not logically entail F), and no $\mathcal{T}$-consistent ABox $\mathcal{A}''$ exists that logically entails F (resp., does not logically entail F) with $\mathcal{T}$, and such that $\mathsf{cl}_\mathcal{T}(\mathcal{A}'')$ has fewer changes than $\mathsf{cl}_\mathcal{T}(\mathcal{A}')$ with respect to $\mathsf{cl}_\mathcal{T}(\mathcal{A})$.

Note that the above update semantics makes use of the notion of logical entailment. In case where $\mathcal{K}$ is not consistent, the reasoning becomes meaningless, since any conclusion can be inferred from an inconsistent $\mathcal{K}$. So, how can we apply this idea in the case where $\mathcal{K}$ is inconsistent?

Before introducing our solution, we need some preliminary definitions. Given a KB $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$, we denote with $HB(\mathcal{K})$ the *Herbrand Base of* $\mathcal{K}$, i.e. the set of ground atoms that can be built over the symbols appearing in $\mathcal{K}$.

Also, we define the *consistent logical consequences of* $\mathcal{A}$ *with respect to* $\mathcal{T}$ as the set $\mathsf{clc}_\mathcal{T}(\mathcal{A}) = \{\alpha \mid \alpha \in HB(\mathcal{K})$ and there exists $S \subseteq \mathcal{A}$ such that $Mod(\langle\mathcal{T},S\rangle) \neq \emptyset$ and $\langle\mathcal{T},S\rangle \models \alpha\}$. Finally, we say that two KBs $\langle\mathcal{T},\mathcal{A}\rangle$ and $\langle\mathcal{T},\mathcal{A}'\rangle$ are *consistently equivalent* (*C-equivalent*) if $\mathsf{clc}_\mathcal{T}(\mathcal{A}) = \mathsf{clc}_\mathcal{T}(\mathcal{A}')$.

Notice that, if $\langle\mathcal{T},\mathcal{A}\rangle$ is an $\mathcal{L}$-KB, then $\langle\mathcal{T},\mathsf{clc}_\mathcal{T}(\mathcal{A})\rangle$ is an $\mathcal{L}$-KB as well. Notice also that, if $\mathcal{A}$ is $\mathcal{T}$-consistent, then $\mathsf{clc}_\mathcal{T}(\mathcal{A}) = \mathsf{cl}_\mathcal{T}(\mathcal{A})$, i.e., the consistent logical consequences of $\mathcal{K}$ coincides with $\mathsf{cl}_\mathcal{T}(\mathcal{A})$.

Generally speaking, our update semantics makes use of the notion of *repair* of a $\mathcal{T}$-inconsistent ABox. Intuitively, a repair of $\mathcal{A}$ with respect to $\mathcal{T}$ is a $\mathcal{T}$-consistent ABox differing from $\mathcal{A}$ minimally. In order to specify what it means to differ minimally from $\mathcal{A}$, we resort both to $\mathsf{clc}_\mathcal{T}(\mathcal{A})$ and to the notion of "few changes".

Definition 2 Let $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ be a possibly inconsistent KB. A *closed ABox repair (CA-repair)* for $\mathcal{K}$ is a set $\mathcal{A}'$ of membership assertions such that:

(i) $\mathcal{A}'$ is $\mathcal{T}$-consistent, i.e., $Mod(\langle\mathcal{T},\mathcal{A}'\rangle) \neq \emptyset$, and
(ii) there is no set $\mathcal{A}''$ of ABox assertions that is $\mathcal{T}$-consistent, and such that $\mathsf{cl}_\mathcal{T}(\mathcal{A}'')$ has fewer changes than $\mathsf{cl}_\mathcal{T}(\mathcal{A})'$ with respect to $\mathsf{clc}_\mathcal{T}(\mathcal{A})$.

The set of closed ABox repairs for $\mathcal{K}$ is denoted by *CAR-Set*$(\mathcal{K})$.

We now present an example illustrating the notion of *CA*-repair.

Example 2 Consider the *DL-Lite*$_{A,id}$-KB $\mathcal{K}' = \langle\mathcal{T},\mathcal{A}'\rangle$, where $\mathcal{T}$ is the TBox presented in Example 1, and $\mathcal{A}'$ is the ABox formed by the following assertions:

$$\mathcal{A}' = \{\, CX(c_1), mf(c_1,t_1), fb(w,c_1), CX(c_2), mf(c_2,t_1), fb(w,c_2), CH(h), mf(h,t_2), OA(h) \,\}.$$

It is easy to see that $\mathcal{K}'$ is inconsistent, since the crew w has two coxs, namely c_1 and c_2, and h is both a coach and an athlete. Up to logical equivalence, the set *CAR-Set*$(\mathcal{K}')$ is constituted by the following $\mathcal{T}$-consistent ABoxes:

$$\textit{CA-rep}_1 = \{\, CX(c_1), mf(c_1,t_1), CX(c_2), mf(c_2,t_1), fb(w,c_1), mf(h,t_2), CH(h) \,\};$$
$$\textit{CA-rep}_2 = \{\, CX(c_1), mf(c_1,t_1), CX(c_2), mf(c_2,t_1), fb(w,c_1), mf(h,t_2), OA(h) \,\};$$
$$\textit{CA-rep}_3 = \{\, CX(c_1), mf(c_1,t_1), CX(c_2), mf(c_2,t_1), fb(w,c_2), mf(h,t_2), CH(h) \,\};$$
$$\textit{CA-rep}_4 = \{\, CX(c_1), mf(c_1,t_1), CX(c_2), mf(c_2,t_1), fb(w,c_2), mf(h,t_2), OA(h) \,\};$$
$$\textit{CA-rep}_5 = \{\, ATH(c_1), mf(c_1,t_1), CX(c_2), mf(c_2,t_1), fb(w,c_1), fb(w,c_2), mf(h,t_2), CH(h) \,\};$$
$$\textit{CA-rep}_6 = \{\, ATH(c_1), mf(c_1,t_1), CX(c_2), mf(c_2,t_1), fb(w,c_1), fb(w,c_2), mf(h,t_2), OA(h) \,\};$$
$$\textit{CA-rep}_7 = \{\, CX(c_1), mf(c_1,t_1), ATH(c_2), mf(c_2,t_1), fb(w,c_1), fb(w,c_2), mf(h,t_2), CH(h) \,\};$$
$$\textit{CA-rep}_8 = \{\, CX(c_1), mf(c_1,t_1), ATH(c_2), mf(c_2,t_1), fb(w,c_1), fb(w,c_2), mf(h,t_2), OA(h) \,\}.$$

□

Our solution for updating inconsistent KBs is based on a simple modification of the notions of "accomplishing the insertion" and "accomplishing the deletion" given in [12] in case of consistent KBs. We sanction that an ABox $\mathcal{A}'$ accomplishes the insertion of F into a possibly inconsistent KB $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ if for all *CA*-repairs $\mathcal{A}''$ of $\langle\mathcal{T},\mathcal{A}'\rangle$, we have that $\langle\mathcal{T},\mathcal{A}''\rangle$ logically entails F. Similarly, we say that an ABox $\mathcal{A}'$ accomplishes the deletion of F from a possibly inconsistent KB $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ if for all *CA*-repairs $\mathcal{A}''$ of $\langle\mathcal{T},\mathcal{A}'\rangle$, we have that $\langle\mathcal{T},\mathcal{A}''\rangle$ does not logically entail F. More formally.

Definition 3 The ABox $\mathcal{A}'$ *accomplishes the insertion* of F into $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ if for all $\mathcal{A}'' \in$ *CAR-Set*$(\langle\mathcal{T},\mathcal{A}'\rangle)$, we have that $\langle\mathcal{T},\mathcal{A}''\rangle \models F$. Similarly, the ABox $\mathcal{A}'$ *accomplishes the deletion* of F from $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ if for all $\mathcal{A}'' \in$ *CAR-Set*$(\langle\mathcal{T},\mathcal{A}'\rangle)$, we have that $\langle\mathcal{T},\mathcal{A}''\rangle \not\models F$.

Then, we specify when a set of ABox assertions accomplishes the update of $\langle\mathcal{T},\mathcal{A}\rangle$ with F minimally.

Definition 4 Let $\mathcal{A}'$ be an ABox. $\mathcal{A}'$ accomplishes the insertion (resp., deletion) of F into (resp., from) $\langle\mathcal{T},\mathcal{A}\rangle$ *minimally* if $\mathcal{A}'$ accomplishes the insertion (resp., deletion) of F into (resp., from) $\langle\mathcal{T},\mathcal{A}\rangle$, and there is no ABox $\mathcal{A}''$ that accomplishes the insertion (resp., deletion) of F into (resp., from) $\langle\mathcal{T},\mathcal{A}\rangle$, and such that $\mathsf{clc}_\mathcal{T}(\mathcal{A}'')$ has fewer changes than $\mathsf{clc}_\mathcal{T}(\mathcal{A}')$ with respect to $\mathsf{clc}_\mathcal{T}(\mathcal{A})$.

Example 3 Consider the *DL-Lite*$_{A,id}$-KB $\mathcal{K}' = \langle\mathcal{T},\mathcal{A}'\rangle$ presented in Example 2, and suppose that we want to update $\mathcal{K}'$ with the insertion of the set of ABox assertions $F^+ = \{CX(c1), fb(w,c1), mf(h,t_1)\}$. In words, F^+ states that c_1 is the cox of the crew w, and that h is member of the rowing team t_1. Then, it is easy to verify that both the following $\mathcal{T}$-inconsistent ABoxes accomplish the insertion of F^+ into $\mathcal{K}'$ minimally.

$\mathcal{A}_1^+ = \{ CX(c_1), mf(c_1,t_1), fb(w,c_1), mf(c_2,t_1), mf(h,t_1), OA(h), CH(h), RTM(t_2), fb(w,c_2) \};$
$\mathcal{A}_2^+ = \{ CX(c_1), mf(c_1,t_1), fb(w,c_1), mf(c_2,t_1), mf(h,t_1), OA(h), CH(h), RTM(t_2), CX(c_2) \}.$

Moreover, it can been shown that every other ABox accomplishing the insertion of F^+ into $\mathcal{K}'$ minimally has the same consistent logical consequences with respect to $\mathcal{T}$ of $\mathcal{A}_1^+$ or $\mathcal{A}_2^+$. Now, suppose that we want to update $\mathcal{K}'$ with the deletion of $F^- = \{mf(c_1,t_1), OA(h)\}$. Then, it is easy to verify that both the following $\mathcal{T}$-inconsistent ABoxes accomplish the deletion of F^- from $\mathcal{K}'$ minimally.

$\mathcal{A}_1^- = \{ CX(c_1), fb(w,c_1), CX(c_2), mf(c_2,t_1), fb(w,c_2), CH(h), mf(h,t_2), OA(h) \};$
$\mathcal{A}_2^- = \{ CX(c_1), mf(c_1,t_1), fb(w,c_1), CX(c_2), mf(c_2,t_1), fb(w,c_2), CH(h), mf(h,t_2), ATH(h) \}.$

Also in this case, we have that every other ABox accomplishing the deletion of F^- from $\mathcal{K}'$ minimally has the same consistent logical consequences with respect to $\mathcal{T}$ of $\mathcal{A}_1^-$ or $\mathcal{A}_2^-$. □

Finally, according with the WIDTIO principle, we base our semantics for update on the intersection of all the ABoxes accomplishing the update minimally.

Definition 5 Let $\mathcal{U} = \{\mathcal{A}_1, \dots, \mathcal{A}_n\}$ be the set of all ABoxes accomplishing the insertion (resp., deletion) of F into (resp., from) $\langle\mathcal{T},\mathcal{A}\rangle$ minimally, and let $\mathcal{A}'$ be an ABox. Then, $\langle\mathcal{T},\mathcal{A}'\rangle$ is the result of updating $\langle\mathcal{T},\mathcal{A}\rangle$ with the insertion (resp., deletion) of F if $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}') = \bigcap_{1\leq i\leq n} \mathsf{clc}_{\mathcal{T}}(\mathcal{A}_i)$.

Example 4 Consider again $\mathcal{K}'$ and F of Example 3, and the following $\mathcal{T}$-inconsistent ABoxes:

$\mathcal{A}_u^+ = \{ CX(c_1), mf(c_1,t_1), fb(w,c_1), mf(c_2,t_1), mf(h,t_1), OA(h), CH(h), RTM(t_2), ATH(c_2) \}$
$\mathcal{A}_u^- = \{ CX(c_1), fb(w,c_1), CX(c_2), mf(c_2,t_1), fb(w,c_2), CH(h), mf(h,t_2), ATH(h) \}.$

It is immediate that $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}_u^+)$ coincides with the intersection of $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}_1^+)$ and $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}_2^+)$, and that $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}_u^-)$ coincides with the intersection of $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}_1^-)$ and $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}_2^-)$ shown in Example 3. According to Definition 5, we have that $\mathcal{K}_u^+ = \langle\mathcal{T},\mathcal{A}_u^+\rangle$ is therefore the result of updating $\mathcal{K}' = \langle\mathcal{T},\mathcal{A}'\rangle$ with the insertion of F^+, and that $\mathcal{K}_u^- = \langle\mathcal{T},\mathcal{A}_u^-\rangle$ is the result of updating $\mathcal{K}' = \langle\mathcal{T},\mathcal{A}'\rangle$ with the deletion of F^-. □

Observe that if $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ is an inconsistent $\mathcal{L}$-KB and F is $\mathcal{T}$-consistent, then in case of both insertion and deletion, our semantics guarantees that an $\mathcal{L}$-KB $\mathcal{K}_u$ resulting from updating $\mathcal{K}$ with F always exists. Moreover, note that if $\mathcal{K}_u$ results from the insertion of F into $\mathcal{K}$, then $\mathcal{K}_u$ may be inconsistent, but every repair of $\mathcal{K}_u$ entails F. Similarly, if $\mathcal{K}_u$ results from deleting F from $\mathcal{K}$, then $\mathcal{K}_u$ may be inconsistent, but every repair of $\mathcal{K}_u$ does not entail F. Conversely, in the case where the original KB is consistent, our update semantics coincides with the semantics for updating consistent KBs presented in [12], and therefore consistency is preserved by the update operation.

Notably, in the case of inconsistent KBs, our update operators is closed with respect to *C-equivalence*. Indeed, it can be shown that if $\mathcal{K}_1$ and $\mathcal{K}_2$ are two *C-equivalent* KBs, and $\mathcal{K}_1'$ results from the update of $\mathcal{K}_1$ with F, and $\mathcal{K}_2'$ results from the update of $\mathcal{K}_2$ with F, then $\mathcal{K}_1'$ and $\mathcal{K}_2'$ are *C-equivalent*. This implies that, up to *C-equivalent*, exactly one KB results from the update, and this will be called *the result of the update*.

4 Inconsistency-tolerant insertion in *DL-Lite*$_{A,id}$

In this section we address the problem of updating a possibly inconsistent *DL-Lite*$_{A,id}$-KB $\mathcal{K}$ with the insertion of F, according to the semantics given in the previous section.

The algorithm we present is based on the characterization in the context of *DL-Lite*$_{A,id}$ of when an atom in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A})$ does not belong to some ABox accomplishing the insertion of a set of atoms minimally. Indeed, according with the WIDTIO principle, an atom α will not be in the result of the update exactly when it does not appear in at least one ABox accomplishing the insertion of F into $\mathcal{K}$ minimally. The following proposition provides such characterization.

Proposition 1 *Let α be an atom in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \setminus \mathsf{cl}_{\mathcal{T}}(F)$. There is $\mathcal{A}'$ accomplishing the insertion of F into $\langle\mathcal{T},\mathcal{A}\rangle$ minimally with $\alpha \notin \mathsf{clc}_{\mathcal{T}}(\mathcal{A}')$ if and only if there is a $\mathcal{T}$-inconsistent set V in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \cup \mathsf{cl}_{\mathcal{T}}(F)$ such that: (i) $\alpha \in V$; (ii) $F \cup (V \setminus \{\alpha\})$ is $\mathcal{T}$-consistent; (iii) $V \cap \mathsf{cl}_{\mathcal{T}}(F) \neq \emptyset$.*

Example 5 Referring to Example 3, the ABox constituted by $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}') \cup \mathsf{cl}_{\mathcal{T}}(F^+)$ contains the following $\mathcal{T}$-inconsistent sets:

$inset_1 = \{ CX(c_1), CX(c_2), fb(w,c_1), fb(w,c_2) \};$
$inset_2 = \{ mf(h,t_1), mf(h,t_2) \};$
$inset_3 = \{ CH(h), ATH(h) \};$
$inset_4 = \{ CH(h), OA(h) \}$

where $inset_1$ contradicts the ID $(id\ CX\ \ fb^-)$, $inset_2$ contradicts the functionality assertion $(funct\ mf)$, and both $inset_3$ and $inset_4$ contradict the negative inclusion $CH \sqsubseteq \neg ATH$. Note that only $inset_1$ and $inset_2$ overlap with $\mathsf{cl}_{\mathcal{T}}(F)$. According to Definition 5, and by exploiting Proposition 1, is easy to verify that the *DL-Lite*$_{A,id}$-KB $\mathcal{K}_u^+ = \langle\mathcal{T},\mathcal{A}_u^+\rangle$, where

$\mathcal{A}_u^+ = \{ CX(c_1), mf(c_1,t_1), fb(w,c_1), mf(c_2,t_1), mf(h,t_1), OA(h), CH(h), RTM(t_2), ATH(c_2) \}$

is the result of updating $\langle\mathcal{T},\mathcal{A}\rangle$ with the insertion of F. □

From Proposition 1 one can derive the following algorithm for computing the result of an update by insertion.

Algorithm $\mathsf{ComputeInsertion}_{CAR}(\mathcal{K}, F)$
Input: a *DL-Lite*$_{A,id}$-KB $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$, a set of ABox assertions F;
Output a *DL-Lite*$_{A,id}$-KB.
begin
  $W \leftarrow \mathsf{InconsistentSets}(\langle\mathcal{T},\mathcal{A}\cup F\rangle)$;
  $D \leftarrow \emptyset$;
  foreach $\alpha \in \mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \setminus \mathsf{cl}_{\mathcal{T}}(F)$ **do**
    if $\exists w \in W$ s.t.
      (i) $\alpha \in w$ and
      (ii) $F \cup (V \setminus \{\alpha\})$ is $\mathcal{T}$-consistent, and
      (iii) $\mathsf{cl}_{\mathcal{T}}(F) \cap w \neq \emptyset$
    then $D \leftarrow D \cup \{\alpha\}$;
  return $\langle\mathcal{T}, F \cup \mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \setminus D\rangle$;
end

The algorithm essentially computes the set D of ABox assertions in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \setminus \mathsf{cl}_{\mathcal{T}}(F)$ which do not belong to at least one ABox accomplishing the insertion of F into $\mathcal{K}$ minimally. It proceeds as follows. In order to exploit Proposition 1 the algorithm needs to compute all $\mathcal{T}$-inconsistent sets in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \cup \mathsf{cl}_{\mathcal{T}}(F)$. To this end, $\mathsf{ComputeInsertion}_{CAR}$ uses the algorithm $\mathsf{InconsistentSets}$ that computes the set W of all $\mathcal{T}$-inconsistent sets in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \cup \mathsf{cl}_{\mathcal{T}}(F)$. Afterwards it adds to the set D each assertion $\alpha \in \mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \setminus \mathsf{cl}_{\mathcal{T}}(F)$ that is contained in at least one $w \in W$ that overlaps with $\mathsf{cl}_{\mathcal{T}}(F)$ and such that $F \cup w \setminus \{\alpha\}$ is $\mathcal{T}$-consistent. Finally, the algorithm returns the KB $\langle \mathcal{T}, F \cup \mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \setminus D\rangle$.

It remains to present the algorithm InconsistentSets. As shown in [6], by virtue of the characteristics of *DL-Lite*$_{A,id}$, we can compute the set W of all $\mathcal{T}$-inconsistent sets in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \cup \mathsf{cl}_{\mathcal{T}}(F)$ in polynomial time with respect to the size of $\mathcal{A} \cup F$. Indeed, the key property of *DL-Lite*$_{A,id}$ that we exploit is that for every $\mathcal{T}$-inconsistent sets W in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \cup \mathsf{cl}_{\mathcal{T}}(F)$, there is one assertions in $\mathcal{T}^- \cup \mathcal{T}_f \cup \mathcal{T}_{id}$ that is violated by W. Therefore, every $\mathcal{T}$-inconsistent set in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \cup \mathsf{cl}_{\mathcal{T}}(F)$ corresponds to the certain answer of a suitable conjunctive query built out of one assertion α in $\mathcal{T}^- \cup \mathcal{T}_f \cup \mathcal{T}_{id}$. Essentially, such conjunctive query looks for the tuples that form the facts satisfying the negation of α.

To illustrate the technique, we define a translation function φ from assertion in $\mathcal{T}^- \cup \mathcal{T}_f \cup \mathcal{T}_{id}$ to boolean conjunctive query with inequalities. In what follows, B is a basic concept, Q is a basic role, U is an attribute and x, x_1, and x_2 are variables.

- $\varphi((\mathsf{funct}\ P))$: $q() = P(x, x_1) \wedge P(x, x_2) \wedge x_1 \neq x_2$
- $\varphi((\mathsf{funct}\ P^-))$: $q() = P(x_1, x) \wedge P(x_2, x) \wedge x_1 \neq x_2$
- $\varphi((\mathsf{funct}\ U))$: $q() = U(x, x_1) \wedge U(x, x_2) \wedge x_1 \neq x_2$
- $\varphi(B_1 \sqsubseteq \neg B_2)$: $q() = \gamma_1(B_1, x) \wedge \gamma_2(B_2, x)$
- $\varphi(Q_1 \sqsubseteq \neg Q_2)$: $q() = \sigma(Q_1, x_1, x_2) \wedge \sigma(Q_2, x_1, x_2)$
- $\varphi(U_1 \sqsubseteq \neg U_2)$: $q() = U_1(x_1, x_2) \wedge U_2(x_1, x_2)$
- $\varphi((id\ B\ \pi_1, \dots, \pi_n))$: $q() = \gamma(B, x) \wedge \gamma(B, x') \wedge x \neq x' \wedge \bigwedge_{1 \leq i \leq n}(\rho(\pi_i(x, x_i)) \wedge \rho(\pi_i(x', x_i)))$

where: $\gamma(B, x) = A(x)$ if $B = A$, $\gamma(B, x) = P(x, y_{new})$ (resp., $P(y_{new}, x)$) if $B = \exists P$ (resp., $B = \exists P^-$), or $\gamma(B, x) = U(x, y_{new})$ if $B = \delta(U)$, where y_{new} is a fresh variable.

$\sigma(Q, x, y) = P(x, y)$ if $Q = P$, or $\sigma(Q, x, y) = P(y, x)$ if $Q = P^-$, and $\rho(\pi(x, y))$ is inductively defined on the structure of path π as follows:

1. If $\pi = B_1? \circ \cdots \circ B_h? \circ Q \circ B'_1? \circ \cdots \circ B'_k?$ (with $h \geq 0$, $k \geq 0$), then $\rho(\pi(x, y)) = \gamma(B_1, x) \wedge \cdots \wedge \gamma(B_h, x) \wedge Q(x, y) \wedge \gamma(B'_1, y) \wedge \cdots \wedge \gamma(B'_k, y)$.
2. If $\pi = \pi_1 \circ \pi_2$, where $length(\pi_1) = 1$ and $length(\pi_2) \geq 1$, then $\rho(\pi(x, y)) = \rho(\pi_1(x, z)) \wedge \rho(\pi_2(z, y))$, with z a fresh variable symbol.

If $q() \leftarrow \psi(\vec{x})$ is a boolean CQ, then we denote with $q(\vec{x})$ the conjunctive query obtained by transforming all existential variables in $q() \leftarrow \psi(\vec{x})$ into distinguished variables. Moreover, if $q(\vec{x}) \leftarrow \psi(\vec{x})$ is a CQ with inequalities and without non-distinguished variables, and $\vec{t}$ is a tuple of constants, we denote by $\mathsf{facts}(q, \vec{t})$ the set of ABox assertions in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A})$ built by replacing the variables $\vec{x}$ in $\psi(\vec{x})$ with the tuple $\vec{t}$.

We are ready to present the algorithm InconsistentSets whose goal is to compute all $\mathcal{T}$-inconsistent sets in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}) \cup \mathsf{cl}_{\mathcal{T}}(F)$.

Algorithm InconsistentSets($\langle \mathcal{T}, \mathcal{A}\rangle$)
Input: *DL-Lite*$_{A,id}$-KB $\langle \mathcal{T}^+ \cup \mathcal{T}^- \cup \mathcal{T}_f \cup \mathcal{T}_{id}, \mathcal{A}\rangle$;
Output: a set of ABox assertions.
begin
  $W \leftarrow \emptyset$;
  foreach $\alpha \in \mathcal{T}^- \cup \mathcal{T}_f \cup \mathcal{T}_{id}$ **do**
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathsf{PerfectRef}_{\neq}(\mathcal{T}^+, \varphi(\alpha))$;
  foreach $q() \in \mathcal{Q}$ **and** $\vec{t} \in ans(q(\vec{x}), \langle \emptyset, \mathsf{clc}_{\mathcal{T}}(\mathcal{A})\rangle)$ **do**
    $W \leftarrow W \cup \{\mathsf{facts}(q(\vec{x}), \vec{t})\}$;
  return V;
end

Firstly, InconsistentSets computes, for each assertion $\alpha \in \mathcal{T}^- \cup \mathcal{T}_f \cup \mathcal{T}_{id}$, the boolean conjunctive query with inequalities corresponding to the negation of α, by means of the translation function φ, and then it computes the perfect reformulation [5] of $\varphi(\alpha)$ by means of the algorithm $\mathsf{PerfectRef}_{\neq}$. We remind the reader that UCQs in *DL-Lite*$_{A,id}$ are FOL-rewritable, i.e., for every union of conjunctive queries q and every *DL-Lite*$_{A,id}$ TBox $\mathcal{T}$, there exists a FOL query q_r, over the alphabet of $\mathcal{T}$, such that for every non-empty ABox $\mathcal{A}$ it holds that $\langle \mathcal{T}, \mathcal{A}\rangle \models q$ if and only if q_r evaluates to true over $\mathcal{A}$, i.e., $\langle \emptyset, \mathcal{A}\rangle \models q_r$. The query q_r is called the *perfect FOL reformulation* of q w.r.t. $\mathcal{T}$. An algorithm for computing such reformulation, called PerfectRef, is provided in [5]. In a nutshell, PerfectRef takes as input a UCQ q and a *DL-Lite*$_{A,id}$ TBox $\mathcal{T}$ and compiles in q the knowledge of $\mathcal{T}$ useful for answering q, returning another UCQs over $\mathcal{T}$ which is the perfect FOL reformulation of q w.r.t. $\mathcal{T}$. The reformulation computed by $\mathsf{PerfectRef}_{\neq}$ is a slight variation of the one computed by the algorithm PerfectRef: in this modified version, inequality is considered as a primitive role, therefore never "expanded", and the variables occurring in inequality atoms are never "reduced" (i.e., transformed by unification steps into non-join variables). Note that the result of $\mathsf{PerfectRef}_{\neq}$ is a boolean UCQ with inequalities, represented as a set of boolean CQ with inequalities, as usual. Every boolean query $q()$ in such a set is then transformed by InconsistentSets into the corresponding query $q(\vec{x})$ without non-distinguished variables and evaluated over the KB $\langle \emptyset, \mathsf{clc}_{\mathcal{T}}(\mathcal{A})\rangle$, so that every certain answer $\vec{t}$ thus computed produces the set $\mathsf{facts}(q(\vec{x}), \vec{t})$ that is inserted into V, where $\mathsf{facts}(q(\vec{x}), \vec{t})$ is a $\mathcal{T}$-inconsistent set violating the assertion in $\mathcal{T}^- \cup \mathcal{T}_f \cup \mathcal{T}_{id}$ corresponding to $q()$. At the end, V contains all $\mathcal{T}$-inconsistent sets in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A})$.

Proposition 2 *Let $\langle \mathcal{T}, \mathcal{A}\rangle$ be a DL-Lite$_{A,id}$-KB. Then InconsistentSets($\langle \mathcal{T}, \mathcal{A}\rangle$) terminates, and computes all $\mathcal{T}$-inconsistent sets in $\mathsf{clc}_{\mathcal{T}}(\mathcal{A})$ in polynomial time with respect to $|\mathcal{T} \setminus \mathcal{T}_{id}|$ and $|\mathcal{A}|$, and in exponential time with respect to $|\mathcal{T}_{id}|$.*

Finally, from Proposition 1 and Proposition 2, one can immediately derive the following theorem.

Theorem 1 *ComputeInsertion$_{CAR}$($\langle \mathcal{T}, \mathcal{A}\rangle, F$) terminates, and computes a KB which is the result of updating $\langle \mathcal{T}, \mathcal{A}\rangle$ with the insertion of F in polynomial time with respect to $|\mathcal{T} \setminus \mathcal{T}_{id}|$, $|\mathcal{A}|$ $|F|$, and in exponential time with respect to $|\mathcal{T}_{id}|$.*

5 Inconsistency-tolerant deletion in *DL-Lite*$_{A,id}$

In this section we study deletion under the assumption that the DL language $\mathcal{L}$ is *DL-Lite*$_{A,id}$. Thus, in what follows, we implicitly refer to a *DL-Lite*$_{A,id}$-KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$, and we address the problem of updating $\mathcal{K}$ with the deletion of a finite set F of ABox assertions according to the semantics given in Section 3.

By Definition 3 we have that an ABox $\mathcal{A}'$ accomplishes the deletion of F from $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ if every CAR-repair of $\langle \mathcal{T}, \mathcal{A}'\rangle$ does not imply F. It follows that, in the case where every CAR-repair of the original KB $\mathcal{K}$ does not imply F, we have that the ABox $\mathcal{A}$ accomplishes the deletion of F from $\mathcal{K}$. Moreover, since no changes have been made, $\mathcal{A}$ accomplishes the deletion of F from $\mathcal{K}$ minimally. From the observations above, we have the following proposition.

Proposition 3 *Let $\langle \mathcal{T}, \mathcal{A}\rangle$ be a possibly inconsistent DL-Lite$_{A,id}$-KB, and let F be a set of ABox assertions. If F is $\mathcal{T}$-inconsistent, or $F \not\subseteq \mathsf{clc}_{\mathcal{T}}(\mathcal{A})$, then the result of updating $\langle \mathcal{T}, \mathcal{A}\rangle$ with the deletion of F is $\langle \mathcal{T}, \mathcal{A}\rangle$ itself.*

In other words, Proposition 3 states that if F is $\mathcal{T}$-inconsistent, or if $F \not\subseteq \mathsf{clc}_{\mathcal{T}}(\mathcal{A})$, then the deletion operator does not modify the original KB. In what follows we focus on the case where F is $\mathcal{T}$-consistent and $F \subseteq \mathsf{clc}_{\mathcal{T}}(\mathcal{A})$.

We start analyzing the problem in the case where the set F is constituted by a single ABox assertion f. By virtue of the characteristics of *DL-Lite*$_{A,id}$ we have that the ABox computed by removing from $\mathsf{clc}_{\mathcal{T}}(\mathcal{A})$ every assertion α such that $\langle \mathcal{T}, \alpha \rangle \models f$ accomplishes the deletion of $\{f\}$ from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally. Note that from the definition of *consistent logical consequences* given in Section 3, we have that $\{\alpha\}$ is $\mathcal{T}$-consistent for every ABox assertion $\alpha \in \mathsf{clc}_{\mathcal{T}}(\mathcal{A})$.

We now consider the case of arbitrary F, i.e., the case where $F = \{f_1, \ldots, f_m\}$, for $m \geq 0$. Let $\mathcal{U} = \{\mathcal{A}_1 \ldots \mathcal{A}_m\}$ be a set of ABoxes $\mathcal{A}_i$, such that, for every $1 \leq i \leq m$, $\mathcal{A}_i$ accomplishes the deletion of $\{f_i\} \subseteq F$ from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally. The following lemmas are the key to our solution.

Lemma 1 *Let $\mathcal{A}_i$ and $\mathcal{A}_j$ be two ABoxes in $\mathcal{U}$ such that $\mathcal{A}_i$ and $\mathcal{A}_j$ accomplishes respectively the deletion of $\{f_i\} \subseteq F$ from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally, and the deletion of $\{f_i\} \subseteq F$ from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally. $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}_i)$ has fewer changes than $\mathsf{clc}_{\mathcal{T}}(\mathcal{A}_j)$ with respect to $\mathsf{clc}_{\mathcal{T}}(\mathcal{A})$ if and only if $\langle \mathcal{T}, \{f_i\} \rangle \models \langle \mathcal{T}, \{f_j\} \rangle$ and $\langle \mathcal{T}, \{f_j\} \rangle \not\models \langle \mathcal{T}, \{f_i\} \rangle$.*

As a consequence of Lemma 1 we have that if an ABox $\mathcal{A}_i$ accomplishes the deletion of $\{f_i\} \subseteq F$ from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally, and there does not exists any other assertion $f_j \neq f_i$ in F such that $\langle \mathcal{T}, \{f_j\} \rangle \models f_i$ and $\langle \mathcal{T}, \{f_i\} \rangle \not\models f_j$, then $\mathcal{A}_i$ accomplishes the deletion of F from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally.

The following lemma guarantees that no ABox, other than those contained in $\mathcal{U}$, accomplishes the deletion of F from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally.

Lemma 2 *Let $\langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_{A,id}$-KB, and let F be a set of ABox assertions such that $\langle \mathcal{T}, \mathcal{A} \rangle \models F$. If $\mathcal{A}'$ accomplishes the deletion of F from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally, then there exists an assertion $f' \in F$ such that $\mathcal{A}'$ accomplishes the deletion of $\{f'\}$ from $\langle \mathcal{T}, \mathcal{A} \rangle$ minimally.*

Proposition 3, Lemma 1, and Lemma 2 suggest a direct strategy for computing the result of updating a possibly inconsistent *DL-Lite*$_{A,id}$-KB with the deletion of a set of ABox assertions. Such a strategy is illustrated in the algorithm $\mathsf{ComputeDeletion}_{CAR}$ below.

Algorithm $\mathsf{ComputeDeletion}_{CAR}(\mathcal{K}, F)$
Input: a *DL-Lite*$_{A,id}$-KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, a set of ABox assertions F;
Output a *DL-Lite*$_{A,id}$-KB.
begin
  if $Mod(\langle \mathcal{T}, F \rangle) = \emptyset$ or $F \not\subseteq \mathsf{clc}_{\mathcal{T}}(\mathcal{A})$
  then return $\langle \mathcal{T}, \mathcal{A} \rangle$;
  $F' \leftarrow F$;
  foreach $f_i \in F'$ and $f_j \in F$ such that $f_i \neq f_j$ **do**
    if $\langle \mathcal{T}, \{f_j\} \rangle \models f_i$ and $\langle \mathcal{T}, \{f_i\} \rangle \not\models f_j$
    then $F' \leftarrow F' \setminus \{f_i\}$;
  $F^- \leftarrow \emptyset$;
  foreach $f \in F'$ **do**
    foreach $\alpha \in \mathsf{clc}_{\mathcal{T}}(\mathcal{A})$ **do**
      if $\langle \mathcal{T}, \{\alpha\} \rangle \models f$ **then** $F^- \leftarrow F^- \cup \{\alpha\}$;
  return $\langle \mathcal{T}, \mathsf{cl}_{\mathcal{T}}(\mathcal{A}) \setminus F^- \rangle$;
end

Finally, we have the following theorem.

Theorem 2 *$\mathsf{ComputeDeletion}_{CAR}(\langle \mathcal{T}, \mathcal{A} \rangle, F)$ terminates, and computes a KB which is the result of updating $\langle \mathcal{T}, \mathcal{A} \rangle$ with the deletion of F in polynomial time with respect to $|\mathcal{T}|$, $|\mathcal{A}|$ and $|F|$.*

6 Conclusions

We have presented a new approach to updating inconsistent DL KBs based on the WIDTIO principle, and we have illustrated specific techniques for the case of *DL-Lite*$_{A,id}$-KBs. We are currently extending our work in several directions. In particular, we are studying how to extend the results presented in this paper to the case of full-fledged OBDA systems, i.e., systems where data reside in external sources, and suitable mappings specify how to interpret such data in terms of instances of the concepts and the roles of the *DL-Lite*$_{A,id}$ TBox. Similarly to the case of data integration [3], the main challenge in this context is to devise suitable techniques for pushing the updates to the sources, so as to realize the desired update by means of appropriate operations on the data at the sources.

Acknowledgments. This research has been partially supported by the EU under FP7 project ACSI – Artifact-Centric Service Interoperation (grant n. FP7-257593), and by Regione Lazio under the project "Integrazione semantica di dati e servizi per le aziende in rete".

REFERENCES

[1] *Inconsistency Tolerance*, eds., Leopoldo E. Bertossi, Anthony Hunter, and Torsten Schaub, volume 3300 of *LNCS*, Springer, 2005.
[2] Meghyn Bienvenu, 'First-order expressibility results for queries over inconsistent *DL-Lite* knowledge bases', in *Proc. of DL 2011*, volume 745 of *CEUR*, `ceur-ws.org`, (2011).
[3] Andrea Calì and Diego Calvanese and De Giacomo, Giuseppe and Maurizio Lenzerini, 'Accessing Data Integration Systems through Conceptual Schemas', in *Proc. of ER 2011*, volume 2224 of *LNCS*, 270–284. Springer, 2001.
[4] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo, 'The Mastro system for ontology-based data access', *Semantic Web J.*, **2**(1), 43–53, (2011).
[5] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, 'Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family', *J. of Automated Reasoning*, **39**(3), 385–429, (2007).
[6] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, 'Path-based identification constraints in description logics', in *Proc. of KR 2008*, 231–241, (2008).
[7] Diego Calvanese, Evgeny Kharlamov, Werner Nutt, and Dmitriy Zheleznyakov, 'Evolution of *DL-Lite* knowledge bases', in *Proc. of ISWC 2010*, volume 6496 of *LNCS*, 112–128. Springer, (2010).
[8] De Giacomo, Giuseppe and Maurizio Lenzerini and Antonella Poggi and Riccardo Rosati, 'On the Update of Description Logic Ontologies at the Instance Level', in *Proc. of AAAI 2006*, 1271–1276, 2006.
[9] Ronald Fagin, Jeffrey D. Ullman, and Moshe Y. Vardi, 'On the semantics of updates in databases', in *Proc. of PODS'83*, 352–365, (1983).
[10] Hirofumi Katsuno and Alberto Mendelzon, 'On the difference between updating a knowledge base and revising it', in *Proc. of KR'91*, 387–394, (1991).
[11] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo, 'Inconsistency-tolerant semantics for description logics', in *Proc. of RR 2010*, (2010).
[12] Maurizio Lenzerini and Domenico Fabio Savo, 'On the evolution of the instance level of *DL-Lite* knowledge bases', in *Proc. of DL 2011*, volume 745 of *CEUR*, `ceur-ws.org`, (2011).
[13] H. Liu and C. Lutz and M. Milicic and F. Wolter, 'Updating Description Logic ABoxes', in *Proc. of KR 2006*, 46–56, (2006).
[14] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, 'Linking data to ontologies', *J. on Data Semantics*, **X**, 133–173, (2008).
[15] Riccardo Rosati, 'On the complexity of dealing with inconsistency in description logic ontologies', in *Proc. of IJCAI 2011*, 1057–1062, (2011).
[16] Zhe Wang, Kewen Wang, and Rodney W. Topor, 'A new approach to knowledge base revision in *DL-Lite*', in *Proc. of AAAI 2010*, (2010).
[17] Marianne Winslett, *Updating Logical Databases*, Cambridge University Press, 1990.