Inconsistency-tolerant Semantics for Description Logic Ontologies (extended abstract)*

Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati,
Marco Ruzzi, and Domenico Fabio Savo

Dipartimento di Informatica e Sistemistica
Sapienza Università di Roma
lastname@dis.uniroma1.it

1 Introduction

It is well-known that inconsistency causes severe problems in classical logic. In particular, since an inconsistent logical theory has no model, it logically implies every formula, and, therefore, query answering on an inconsistent knowledge base becomes meaningless. In this paper, we address the problem of dealing with inconsistencies in Description Logic (DL) knowledge bases. Our goal is both to study DL semantical frameworks which are inconsistency-tolerant, and to devise techniques for answering queries posed to DL knowledge bases under such inconsistency-tolerant semantics.

A DL knowledge base is constituted by two components, called the TBox and the ABox, respectively. Intuitively, the TBox includes axioms sanctioning general properties of concepts and relations (such as Dog isa Animal), whereas the ABox contains axioms asserting properties of instances of concepts and relations (such as Bob is an instance of Dog). The various DLs differ in the language (set of constructs) used to express such axioms. We are particularly interested in using DLs for the so-called "ontology-based data access" [8] (ODBA), where a DL TBox acts as an ontology used to access a set of data sources. Since it is often the case that, in this setting, the size of the data at the sources largely exceeds the size of the ontology, DLs where query answering is tractable with respect to the size of the ABox have been studied recently. In this paper, we will consider DLs specifically tailored towards ODBA, in particular DLs of the *DL-Lite* family [8], where query answering can be done efficiently with respect to the size of the ABox.

Depending on the expressive power of the underlying language, the TBox alone might be inconsistent, or the TBox might be consistent, but the axioms in the ABox might contradict the axioms in the TBox. Since in ODBA the ontology is usually represented as a consistent TBox, whereas the data at the sources do not necessarily conform to the ontology, the latter situation is the one commonly occurring in practice. Therefore, our study is carried out under the assumption that the TBox is consistent, and inconsistency may arise between the ABox and the TBox (inconsistencies in the TBox are considered, e.g., in [5, 9]).

There are many approaches for devising inconsistency-tolerant inference systems [1], originated in different areas, including Logic, Artificial Intelligence, and Databases. Our work is especially inspired by the approaches to consistent query answering in databases [3], which are based on the idea of living with inconsistencies (i.e.,

* This paper is an extended abstract of [6].

data that do not satisfy the integrity constraints) in the database, but trying to obtain only consistent information during query answering. But how can one obtain consistent information from an inconsistent database? 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 D, and, therefore, the approach sanctions that what is consistently true in D is simply what is true in all possible repairs of D. Thus, inconsistency-tolerant query answering amounts to compute the tuples that are answers to the query in all possible repairs.

In [7], a semantics for inconsistent knowledge bases expressed in *DL-Lite* has been proposed, based on the notion of repair. More specifically, an ABox $\mathcal{A}'$ is a repair of the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is the TBox and $\mathcal{A}$ is the ABox, if $\mathcal{A}'$ is a maximal subset of $\mathcal{A}$ consistent with $\mathcal{T}$. In this paper, we call such semantics the *ABox Repair (AR) semantics*, and we show that for the DLs of the *DL-Lite* family, inconsistency-tolerant query answering under such a semantics is coNP-complete even for ground atomic queries, thus showing that inconsistency-tolerant instance checking is already intractable. For this reason, we propose a variant of the AR-semantics, based on the idea that inconsistency-tolerant query answering should be done by evaluating the query over the intersection of all AR-repairs. The new semantics, called the *Intersection ABox Repair (IAR) semantics*, is a sound approximation of the AR-semantics, and it enjoys a desirable property, namely that inconsistency-tolerant query answering is polynomially tractable.

Then, we highlight some drawbacks of the AR semantics, and propose a variant called the *Closed ABox Repair (CAR) semantics*, that essentially considers only repairs that are "closed" with respect to the knowledge represented by the TBox. We show that, while inconsistency-tolerant instance checking is tractable under this new semantics in *DL-Lite*, query answering is coNP-complete for unions of conjunctive queries. For this reason, we also study the "intersection-based" version of the CAR-semantics, called the *Intersection Closed ABox Repair ($ICAR$) semantics*, showing that it is a sound approximation of the CAR-semantics, and that inconsistency-tolerant query answering under this new semantics is again polynomially tractable.

2 Preliminaries

Description Logics (DLs) are logics that represent the domain of interest in terms of *concepts*, denoting sets of objects, *value-domains*, denoting sets of values, *attributes*, denoting binary relations between objects and values, and *roles*, denoting binary relations over objects. DL expressions are built starting from an alphabet Γ of symbols for atomic concepts, atomic value-domains, atomic attributes, atomic roles, and object and value constants. We denote by Γ_O the set of object constants, and by Γ_V the set of value constants. Complex expressions are constructed starting from atomic elements, and applying suitable constructs. Different DLs allow for different constructs.

A DL knowledge base (KB) is constituted by two main components: a *TBox* (i.e.,"Terminological Box"), which contains a set of universally quantified assertions stating general properties of concepts and roles, thus representing intensional knowledge of the domain, and an *ABox* (i.e.,"Assertional Box"), which is constituted by assertions on individual objects, thus specifying extensional knowledge. Again, different DLs allow for different kinds of TBox and/or ABox assertions.

Formally, if $\mathcal{L}$ is a DL, then an $\mathcal{L}$-knowledge base $\mathcal{K}$ is a pair $\langle\mathcal{T},\mathcal{A}\rangle$, where $\mathcal{T}$ is a TBox expressed in $\mathcal{L}$ and $\mathcal{A}$ is a ABox. In this paper we assume that the ABox assertions are *atomic*, i.e., they involve only atomic concepts, attributes and roles. The alphabet of $\mathcal{K}$, denoted by $\Gamma_{\mathcal{K}}$, is the set of symbols from Γ occurring in $\mathcal{T}$ and $\mathcal{A}$. The semantics of a DL knowledge base is given in terms of first-order (FOL) interpretations. We denote with *Mod*($\mathcal{K}$) the set of models of $\mathcal{K}$, i.e., the set of FOL interpretations that satisfy all the assertions in $\mathcal{T}$ and $\mathcal{A}$, where the definition of satisfaction depends on the kind of expressions and assertions in the specific DL language in which $\mathcal{K}$ is specified. As usual, a KB $\mathcal{K}$ is said to be *satisfiable* if it admits at least one model, i.e., if $Mod(\mathcal{K}) \neq \emptyset$, and $\mathcal{K}$ is said to *entail* a First-Order Logic (FOL) sentence ϕ, denoted $\mathcal{K} \models \phi$, if $\phi^{\mathcal{I}} = \texttt{true}$ for all $\mathcal{I} \in Mod(\mathcal{K})$. In the following, we are interested in particular in UCQ entailment, i.e., the problem of establishing whether a DL KB entails a *boolean union of conjunctive queries (UCQ)*, i.e., a first order sentence of the form $\exists \boldsymbol{y_1}.conj_1(\boldsymbol{y_1}) \vee \cdots \vee \exists \boldsymbol{y_n}.conj_n(\boldsymbol{y_n})$, where $\boldsymbol{y_1}, \ldots, \boldsymbol{y_n}$ are terms (i.e., constants or variables), and each $conj_i(\boldsymbol{y_i})$ is a conjunction of atoms of the form $A(z)$, $P(z, z')$ and $U(z, z')$ where A is an atomic concept, P is an atomic role and U is an atomic attribute, and z, z' are terms.

3 Inconsistency-tolerant semantics

In this section we present our inconsistency-tolerant semantics for DL knowledge bases. As we said in the introduction, we assume that for a knowledge base $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$, $\mathcal{T}$ is satisfiable, whereas $\mathcal{A}$ may be inconsistent with $\mathcal{T}$, i.e., the set of models of $\mathcal{K}$ may be empty. The challenge is to provide semantic characterizations for $\mathcal{K}$, which are *inconsistency-tolerant*, i.e., they allow $\mathcal{K}$ to be interpreted with a non-empty set of models even in the case where it is unsatisfiable under the classical first-order semantics.

The inconsistency-tolerant semantics we give below are based on the notion of *repair*. Intuitively, given a DL KB $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$, a repair $\mathcal{A}_R$ for $\mathcal{K}$ is an ABox such that the KB $\langle\mathcal{T},\mathcal{A}_R\rangle$ is satisfiable under the first-order semantics, and $\mathcal{A}_R$ "minimally" differs from $\mathcal{A}$. Notice that in general not a single, but several repairs may exist, depending on the particular minimality criteria adopted. We consider here different notions of "minimality", which give rise to different inconsistency-tolerant semantics. In all cases, such semantics coincide with the classical first-order semantics when inconsistency does not come into play, i.e., when the KB is satisfiable under standard first-order semantics.

The first notion of repair that we consider can be phrased as follows: a repair $\mathcal{A}_R$ of a KB $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ is a maximal subset of $\mathcal{A}$ such that $\langle\mathcal{T},\mathcal{A}_R\rangle$ is satisfiable under the first-order semantics, i.e., there does not exist another subset of $\mathcal{A}$ that strictly contains $\mathcal{A}_R$ and that is consistent with $\mathcal{T}$. Intuitively, each such repair is obtained by throwing away from $\mathcal{A}$ a minimal set of assertions to make it consistent with $\mathcal{T}$. In other words, adding to $\mathcal{A}_R$ another assertion of $\mathcal{A}$ would make the repair inconsistent with $\mathcal{T}$. The formal definition is given below.

Definition 1. *Let $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ be a DL KB. An* ABox Repair *(AR) of $\mathcal{K}$ is a set $\mathcal{A}'$ of membership assertions such that: (i) $\mathcal{A}' \subseteq \mathcal{A}$; (ii) $Mod(\langle\mathcal{T},\mathcal{A}'\rangle) \neq \emptyset$; (iii) there exists no $\mathcal{A}''$ such that $\mathcal{A}' \subset \mathcal{A}'' \subseteq \mathcal{A}$ and $Mod(\langle\mathcal{T},\mathcal{A}''\rangle) \neq \emptyset$. The set of AR-repairs for $\mathcal{K}$ is denoted by AR-$Rep(\mathcal{K})$.*

Based on the above notion of repair, we now define ABox repair models.

Definition 2. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB. An interpretation $\mathcal{I}$ is an* ABox repair model*, or simply an AR-*model*, of $\mathcal{K}$ if there exists $\mathcal{A}' \in$ AR-Rep$(\mathcal{K})$ such that $\mathcal{I} \models \langle \mathcal{T}, \mathcal{A}' \rangle$. The set of ABox repair models of $\mathcal{K}$ is denoted by AR-Mod$(\mathcal{K})$.*

The following notion of consistent entailment is the natural generalization of classical entailment to the ABox repair semantics.

Definition 3. *Let $\mathcal{K}$ be a DL KB, and let ϕ be a first-order sentence. We say that ϕ is AR-*consistently entailed*, or simply AR-*entailed*, by $\mathcal{K}$, written $\mathcal{K} \models_{AR} \phi$, if $\mathcal{I} \models \phi$ for every $\mathcal{I} \in$ AR-Mod$(\mathcal{K})$.*

Example 1. Consider the *DL-Lite*$_{\mathcal{A}}$ knowledge base $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$, where $\mathcal{T}$ contains the following assertions:

$\mathsf{Mechanic} \sqsubseteq \mathsf{TeamMember}$	$\mathsf{Driver} \sqsubseteq \mathsf{TeamMember}$	$\mathsf{Driver} \sqsubseteq \neg\mathsf{Mechanic}$
$\exists\mathsf{drives} \sqsubseteq \mathsf{Driver}$	$\exists\mathsf{drives}^- \sqsubseteq \mathsf{Car}$	$(\mathsf{funct}\ \mathsf{drives})$

Assertions from the first row, from left to right, respectively specify that drivers are team members, mechanics are team members, and drivers are not mechanics (disjointness). In the second row, first two assertions say that the role drives is specified between Driver (domain) and Car (range), and that it is functional, i.e., every driver can drive at most one car. $\mathcal{A}'$ is an ABox constituted by the set of assertions $\{\mathsf{Driver}(felipe), \mathsf{Mechanic}(felipe), \mathsf{TeamMember}(felipe), \mathsf{drives}(felipe, ferrari)\}$. This ABox states that $felipe$ is a team member and that he is both a driver and a mechanic. Notice that this implies that $felipe$ drives $ferrari$ and that $ferrari$ is a car. It is easy to see that $\mathcal{K}$ is unsatisfiable, since $felipe$ violates the disjointness between driver and mechanic. The set *AR-Rep*$(\mathcal{K}')$ is constituted by the set of $\mathcal{T}$-*consistent* ABoxes:

AR-rep_1= $\{\mathsf{Driver}(felipe), \mathsf{drives}(felipe, ferrari), \mathsf{TeamMember}(felipe)\}$;
AR-rep_2= $\{\mathsf{Mechanic}(felipe), \mathsf{TeamMember}(felipe)\}$.

The AR-semantics given above in fact coincides with the inconsistency-tolerant semantics for DL KBs presented in [7], and with the loosely-sound semantics studied in [2] in the context of inconsistent databases. Although this semantics can be considered to some extent the natural choice for the setting we are considering, since each ABox repair stays as close as possible to the original ABox, it has the characteristic to be dependent from the form of the knowledge base. Suppose that $\mathcal{K}'' = \langle \mathcal{T}, \mathcal{A}'' \rangle$ differs from the inconsistent knowledge base $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$, simply because $\mathcal{A}''$ includes assertions that logically follow, using $\mathcal{T}$, from a consistent subset of $\mathcal{A}$ (implying that $\mathcal{K}''$ is also inconsistent). One could argue that the repairs of $\mathcal{K}''$ and the repairs of $\mathcal{K}'$ should coincide. Conversely, the next example shows that, in the AR-semantics the two sets of repairs are generally different.

Example 2. Consider the KB $\mathcal{K}'' = \langle \mathcal{T}, \mathcal{A}'' \rangle$, where $\mathcal{T}$ is the same as in $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ of Example 1, and the ABox $\mathcal{A}''$ is as follows:

$$\mathcal{A}'' = \{\mathsf{Driver}(felipe), \mathsf{Mechanic}(felipe), \mathsf{TeamMember}(felipe), \mathsf{Car}(ferrari), \mathsf{drives}(felipe, ferrari)\}.$$

Notice that $\mathcal{A}''$ can be obtained by adding $\mathsf{Car}(ferrari)$ to $\mathcal{A}'$. Since $\mathsf{Car}(ferrari)$ is entailed by the KB $\langle \mathcal{T}, \{\mathsf{drives}(felipe, ferrari)\}\rangle$, i.e., a KB constituted by the TBox $\mathcal{T}$ of $\mathcal{K}'$ and a subset of $\mathcal{A}'$ that is consistent with $\mathcal{T}$, one intuitively would expect that $\mathcal{K}'$ and $\mathcal{K}''$ have the same repairs under the AR-semantics. This is however not the case, since we have that *AR-Rep*$(\mathcal{K}'')$ is formed by:

AR-rep_3= $\{\mathsf{Driver}(felipe), \mathsf{drives}(felipe, ferrari), \mathsf{TeamMember}(felipe),$
$\mathsf{Car}(ferrari)\}$;
AR-rep_4= $\{\mathsf{Mechanic}(felipe), \mathsf{TeamMember}(felipe), \mathsf{Car}(ferrari)\}$.

Let us finally consider the ground sentence $\mathsf{Car}(ferrari)$. It is easy to see that $\mathsf{Car}(ferrari)$ is AR-entailed by the KB $\mathcal{K}''$ but it is not AR-entailed by the KB $\mathcal{K}'$.

Depending on the particular scenario, and the specific application at hand, the above behavior might be considered incorrect. This motivates the definition of a new semantics that does not present such a characteristic. According to this new semantics, that we call *Closed ABox Repair*, the repairs take into account not only the assertions explicitly included in the ABox, but also those that are implied, through the TBox, by at least one subset of the ABox that is consistent with the TBox.

To formalize the above idea, we need some preliminary definitions. Given a DL 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 ABox assertions that can be built over the alphabet of $\Gamma_{\mathcal{K}}$. Then we define the *consistent logical consequences of* $\mathcal{K}$ as the set *clc*$(\mathcal{K}) = \{\alpha \mid \alpha \in$ *HB*$(\mathcal{K})$ and there exists $S \subseteq \mathcal{A}$ such that *Mod*$(\langle \mathcal{T}, S\rangle) \neq \emptyset$ and $\langle \mathcal{T}, S\rangle \models \alpha\}$. With the above notions in place we can now give the definition of Closed ABox Repair.

Definition 4. *Let* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ *be a DL KB. A* Closed ABox Repair (CAR) *for* $\mathcal{K}$ *is a set* $\mathcal{A}'$ *of membership assertions such that:* $(i)\mathcal{A}' \subseteq$ *clc*$(\mathcal{K})$*,* (ii)*Mod*$(\langle \mathcal{T}, \mathcal{A}'\rangle) \neq \emptyset$*,* (iii) *there exists no* $\mathcal{A}'' \subseteq$ *clc*$(\mathcal{K})$ *such that Mod*$(\langle \mathcal{T}, \mathcal{A}''\rangle) \neq \emptyset$ *The set of* CAR*-repairs for* $\mathcal{K}$ *is denoted by CAR-Rep*$(\mathcal{T}, \mathcal{A})$*.*

In words, a CAR-repair is a maximal subset of *clc*$(\mathcal{K})$ consistent with $\mathcal{T}$. The set of CAR-models of a KB $\mathcal{K}$, denoted *CAR-Mod*$(\mathcal{K})$, is defined analogously to AR-models (cf. Definition 2). Also, CAR-entailment, denoted $\models_{CAR}$, is analogous to AR-entailment (cf. Definition 3).

Example 3. Consider the two KBs $\mathcal{K}'$ and $\mathcal{K}''$ presented in the Example 1 and Example 2. It is easy to see that both *CAR-Rep*$(\mathcal{K}')$ and *CAR-Rep*$(\mathcal{K}'')$ are constituted by the two sets below:

CAR-rep_1=$\{\mathsf{Driver}(felipe), \mathsf{drives}(felipe, ferrari), \mathsf{TeamMember}(felipe),$
$\mathsf{Car}(ferrari)\}$;
CAR-rep_2=$\{\mathsf{Mechanic}(felipe), \mathsf{TeamMember}(felipe), \mathsf{Car}(ferrari)\}$.

It follows that both $\mathcal{K}'$ and $\mathcal{K}''$ CAR*-entail* the ground sentence $\mathsf{Car}(ferrari)$, differently from what happen under the AR-semantics, as showed in Example 2.

As we will see in the next section, entailment of a union of conjunctive queries from a KB $\mathcal{K}$ is intractable both under the AR-semantics and the CAR-semantics. Since this can be an obstacle in the practical use of such semantics, we introduce here approximations of the two semantics, under which we will show in the next section that entailment of unions of conjunctive queries is polynomial. In both cases, the approximation consists in taking as unique repair the intersection of the AR-repairs and of the CAR-repairs, respectively. This actually corresponds to follow the WIDTIO (When you are in doubt throw it out) approach, proposed in belief revision and update [10, 4].

Definition 5. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB. An* Intersection ABox Repair (IAR) *for $\mathcal{K}$ is the set $\mathcal{A}'$ of membership assertions such that $\mathcal{A}' = \bigcap_{\mathcal{A}_i \in \textit{AR-Rep}(\mathcal{K})} \mathcal{A}_i\}$. The (singleton) set of IAR-repairs for $\mathcal{K}$ is denoted by* ***IAR-Rep***$(\mathcal{K})$.

Analogously, we give below the definition of Intersection Closed ABox Repair.

Definition 6. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB. An* Intersection Closed ABox Repair ($ICAR$) *for $\mathcal{K}$ is the set $\mathcal{A}'$ of membership assertions such that $\mathcal{A}' = \bigcap_{\mathcal{A}_i \in \textit{CAR-Rep}(\mathcal{K})} \mathcal{A}_i\}$. The (singleton) set of $ICAR$-repairs for $\mathcal{K}$ is denoted by* ***ICAR-Rep***$(\mathcal{K})$.

The sets ***IAR-Mod***$(\mathcal{K})$ and ***ICAR-Mod***$(\mathcal{K})$ of IAR-models and $ICAR$-models, respectively, and the notions of IAR-entailment and $ICAR$-entailment are defined as usual (cf. Definition 2 and Definition 3). Consider for example the KB $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ presented in Example 1. Then ***IAR-Rep***$(\mathcal{K}')$ is the singleton formed by the ABox $IAR\text{-}rep = AR\text{-}rep_1 \cap AR\text{-}rep_2 = \{\mathsf{TeamMember}(felipe)\}$. In turn, referring to Example 3, ***ICAR-Rep***$(\mathcal{K}')$ is the singleton formed by the ABox $ICAR\text{-}rep_1 = CAR\text{-}rep_1 \cap CAR\text{-}rep_2$ = $\{\mathsf{TeamMember}(\text{felipe}), \mathsf{Car}(\text{ferrari})\}$. It is not difficult to show that the IAR-semantics is a sound approximation of the AR-semantics, and that the $ICAR$-semantics is a sound approximation of the CAR-semantics. It is also easy to see that the converse is not true in general. For instance, the sentence $\mathsf{Driver}(felipe)$ is entailed by $\mathcal{K} = \langle \mathcal{T}, \{\mathsf{drives}(felipe, ferrari), \mathsf{drives}(felipe, mcLaren)\} \rangle$, where $\mathcal{T}$ is the TBox of Example 1, under the AR-semantics, but it is not entailed under the IAR-semantics. It can also be proved that the IAR-semantics is a sound approximation of the $ICAR$-semantics (and not vice versa).

4 Reasoning

In this section we study reasoning in the inconsistency-tolerant semantics introduced in the previous section. In particular, we analyze the problem of UCQ entailment under such semantics in the specific DL *DL-Lite*$_{\mathcal{A}}$ [8] for which reasoning under standard FOL semantics is tractable. We will also consider instance checking, which is a restricted form of UCQ entailment. In this section we will focus on the *data complexity* of query answering, i.e., we will measure the computational complexity only with respect to the size of the ABox (which is usually much larger than the TBox and the queries). It follows from the results in [8] that query answering in *DL-Lite*$_{\mathcal{A}}$ is in AC_o, which is a complexity class contained in PTIME, and therefore is tractable in data complexity.

We start by considering the AR-semantics. It is known that UCQ entailment is intractable under this semantics [7]. Here, we strengthen this result, and show that instance checking under the AR-semantics is already coNP-complete in data complexity even if the KB is expressed in *DL-Lite*$_{core}$. We recall that *DL-Lite*$_{core}$ is the least expressive logic in the *DL-Lite* family, as it only allows for concept expressions of the form $C ::= A|\exists R|\exists R^-$, and for TBox assertions of the form $C_1 \sqsubseteq C_2$, $C_1 \sqsubseteq \neg C_2$.

Theorem 1. *Let $\mathcal{K}$ be a DL-Lite$_{core}$ KB and let α be an ABox assertion. Deciding whether $\mathcal{K} \models_{AR} \alpha$ is coNP-complete with respect to data complexity.*

Next, we focus on the CAR-semantics, and obtain that UCQ entailment under this semantics is coNP-complete even if the TBox language is restricted to *DL-Lite*$_{core}$.

Theorem 2. *Let $\mathcal{K}$ be a DL-Lite$_{core}$ KB and let Q be a UCQ. Deciding whether $\mathcal{K} \models_{CAR} Q$ is coNP-complete with respect to data complexity.*

Notice that, differently from the AR-semantics, the above intractability result for the CAR-semantics does not hold already for the instance checking problem: we will show later in this section that instance checking is indeed tractable under the CAR-semantics.

We now turn our attention to the IAR-semantics, and define the algorithm *Compute-IAR-Repair* (see Figure 1) for computing the IAR-repair of a *DL-Lite*$_{\mathcal{A}}$ KB $\mathcal{K}$. The algorithm simply computes the set $\mathcal{D} \subseteq \mathcal{A}$ of ABox assertions which must be eliminated from the IAR-repair of $\mathcal{K}$.

Algorithm *Compute-IAR-Repair*$(\mathcal{K})$
input: *DL-Lite*$_{\mathcal{A}}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$
output: *DL-Lite*$_{\mathcal{A}}$ ABox $\mathcal{A}'$
begin
  let $\mathcal{D} = \emptyset$;
  for each fact $\alpha \in \mathcal{A}$ **do**
    if $\langle \mathcal{T}, \{\alpha\} \rangle$ unsatisfiable
    then let $\mathcal{D} = \mathcal{D} \cup \{\alpha\}$;
  for each pair of facts $\alpha_1, \alpha_2 \in \mathcal{A} - \mathcal{D}$ **do**
    if $\langle \mathcal{T}, \{\alpha_1, \alpha_2\} \rangle$ unsatisfiable
    then let $\mathcal{D} = \mathcal{D} \cup \{\alpha_1, \alpha_2\}$;
  return $\mathcal{A} - \mathcal{D}$
end

Algorithm *Compute-ICAR-Repair*$(\mathcal{K})$
input: *DL-Lite*$_{\mathcal{A}}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$
output: *DL-Lite*$_{\mathcal{A}}$ ABox $\mathcal{A}'$
begin
  Compute *clc*$(\mathcal{K})$;
  let $\mathcal{D} = \emptyset$;
  for each pair of facts $\alpha_1, \alpha_2 \in$ *clc*$(\mathcal{K})$ **do**
    if $\langle \mathcal{T}, \{\alpha_1, \alpha_2\} \rangle$ unsatisfiable
    then let $\mathcal{D} = \mathcal{D} \cup \{\alpha_1, \alpha_2\}$;
  return *clc*$(\mathcal{K}) - \mathcal{D}$
end

Fig. 1. The *Compute-IAR-Repair* and *Compute-ICAR-Repair* algorithms

The following property, based on the correctness of the previous algorithm, establishes tractability of UCQ entailment under IAR-semantics.

Theorem 3. *Let $\mathcal{K}$ be a DL-Lite$_{\mathcal{A}}$ KB, and let Q be a UCQ. Deciding whether $\mathcal{K} \models_{IAR} Q$ is in PTIME with respect to data complexity.*

We now turn our attention to the $ICAR$-semantics and present the algorithm *Compute-ICAR-Repair* (see Figure 1) for computing the $ICAR$-repair of a *DL-Lite*$_{\mathcal{A}}$ KB $\mathcal{K}$. This algorithm is analogous to the previous algorithm *Compute-IAR-Repair*. The main differences are: (i) the algorithm *Compute-ICAR-Repair* returns (and operates on) a subset of *clc*$(\mathcal{K})$, while the algorithm *Compute-IAR-Repair* returns a subset of the original ABox $\mathcal{A}$; (ii) differently from the algorithm *Compute-IAR-Repair*, the algorithm *Compute-ICAR-Repair* does not need to eliminate ABox assertions α such that $\langle \mathcal{T}, \{\alpha\} \rangle$ is unsatisfiable, since such facts cannot occur in *clc*$(\mathcal{K})$.

Again, through the algorithm *Compute-ICAR-Repair* it is possible to establish the tractability of UCQ entailment under $ICAR$-semantics.

Theorem 4. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{A}}$ KB and let Q be a UCQ. Deciding whether $\mathcal{K} \models_{ICAR} Q$ is in PTIME with respect to data complexity.*

Finally, we consider the instance checking problem under CAR-semantics, and obtain that instance checking under CAR-semantics coincides with instance checking under the $ICAR$-semantics.

Lemma 1. *Let $\mathcal{K}$ be a DL-Lite$_{\mathcal{A}}$ KB, and let α be an ABox assertion. Then, $\mathcal{K} \models_{CAR} \alpha$ iff $\mathcal{K} \models_{ICAR} \alpha$.*

The above property and Theorem 4 allow us to establish tractability of instance checking under the CAR-semantics.

Theorem 5. *Let $\mathcal{K}$ be a DL-Lite$_{\mathcal{A}}$ KB, and let α be an ABox assertion. Deciding whether $\mathcal{K} \models_{CAR} \alpha$ is in PTIME with respect to data complexity.*

We remark that the analogous of Lemma 1 does *not* hold for AR, because AR-repairs are not deductively closed. This is the reason why instance checking under AR-semantics is harder, as stated by Theorem 1.

5 Conclusions

Our work can proceed along different directions. One notable problem we aim at addressing is the design of new algorithms for inconsistency-tolerant query answering both under the IAR-semantics and the $ICAR$-semantics, based on the idea of rewriting the query into a FOL query to be evaluated directly over the inconsistent ABox. We would also like to study reasoning under inconsistency-tolerant semantics in Description Logics outside the DL-lite family.

References

1. L. E. Bertossi, A. Hunter, and T. Schaub, editors. *Inconsistency Tolerance*, volume 3300 of *LNCS*. Springer, 2005.
2. A. Calì, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proc. of PODS 2003*, pages 260–271, 2003.
3. J. Chomicki. Consistent query answering: Five easy pieces. In *Proc. of ICDT 2007*, volume 4353 of *LNCS*, pages 1–17. Springer, 2007.
4. T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artificial Intelligence*, 57:227–270, 1992.
5. Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In *Proc. of IJCAI 2005*, pages 454–459, 2003.
6. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant semantics for description logics. In *Proc. of RR 2010*, 2010.
7. D. Lembo and M. Ruzzi. Consistent query answering over description logic ontologies. In *Proc. of RR 2007*, 2007.
8. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.
9. G. Qi and J. Du. Model-based revision operators for terminologies in description logics. In *Proc. of IJCAI 2009*, pages 891–897, 2009.
10. M. Winslett. *Updating Logical Databases*. Cambridge University Press, 1990.