Inconsistency-tolerant Semantics for Description Logics

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

Dipartimento di Informatica e Sistemistica,
Università di Roma "La Sapienza",
Via Ariosto 25, 00185 Roma, Italy
lastname@dis.uniroma1.it

Abstract. We address the problem of dealing with inconsistencies in Description Logic (DL) knowledge bases. Our general goal is both to study DL semantical frameworks which are inconsistency-tolerant, and to devise techniques for answering unions of conjunctive queries posed to DL knowledge bases under such inconsistency-tolerant semantics. Our work is inspired by the approaches to consistent query answering in databases, which are based on the idea of living with inconsistencies in the database, but trying to obtain only consistent information during query answering, by relying on the notion of database repair. We show that, if we use the notion of repair studied in databases, inconsistency-tolerant query answering is intractable, even for the simplest form of queries. Therefore, we study different variants of the repair-based semantics, with the goal of reaching a good compromise between expressive power of the semantics and computational complexity of inconsistency-tolerant query answering.

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 general goal is both to study DL semantical frameworks which are inconsistency-tolerant, and to devise techniques for answering unions of conjunctive 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 [1]. 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" [13] (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 [4], 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 [12, 9, 8, 14, 11]).

There are many approaches for devising inconsistency-tolerant inference systems [2], originated in different areas, including Logic, Artificial Intelligence, and Databases. Our work is especially inspired by the approaches to consistent query answering in databases [5], which 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. 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 [10], 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 it is consistent with $\mathcal{T}$, and there exists no ABox consistent with $\mathcal{T}$ that is "closer" to $\mathcal{A}$, where an ABox $\mathcal{A}''$ is closer to $\mathcal{A}$ than $\mathcal{A}'$ if $\mathcal{A} \cap \mathcal{A}''$ is a proper superset of $\mathcal{A} \cap \mathcal{A}'$. 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 an approximation of the AR-semantics, and it enjoys a desirable property, namely that inconsistency-tolerant query answering is polynomially tractable.

Both the AR-semantics and the IAR-semantics suffer from a drawback. 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 $\mathcal{A}$. This implies that $\mathcal{K}'$ is also inconsistent, and one would expect that the repairs of $\mathcal{K}'$ and the repairs of $\mathcal{K}$ coincide. On the contrary, since the AR-semantics is not independent from the form of the knowledge base, one can show that, in general, inconsistency-tolerant query answering in the two knowledge bases yields different results. To overcome this drawback, we propose a new variant of the AR-semantics, 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, 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 an approximation of the CAR-semantics, and that inconsistency-tolerant query answering under this new semantics is again polynomially tractable.

The paper is organized as follows. In Section 2 we briefly describe the DL we use in our work. In Section 3 we present the various inconsistency-tolerant semantics we have studied in our investigation. In Section 4 we present the complexity results about such semantics, in terms of both lower bounds and upper bounds. Finally, Section 5 concludes the paper.

2 Preliminaries

Description Logics (DLs) [1] 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 (cf. [1]). We denote with $\mathit{Mod}(\mathcal{K})$ 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 $\mathit{Mod}(\mathcal{K}) \neq \emptyset$, and $\mathcal{K}$ is said to *entail* a First-Order Logic (FOL) sentence ϕ, denoted $\mathcal{K} \models \phi$, if $\phi^{\mathcal{I}} = \mathtt{true}$ for all $\mathcal{I} \in \mathit{Mod}(\mathcal{K})$.

We now provide some details about the DL *DL-Lite*$_{\mathcal{A}}$, a member of the *DL-Lite* family [4]. This is a family of tractable DLs particularly suited for dealing with KBs with very large ABoxes, and is at the basis of OWL 2 QL, one of the profiles of OWL 2, the official ontology language of the World Wide Web Consortium (W3C). In *DL-Lite*$_{\mathcal{A}}$, the alphabet Γ is partitioned into 6 subsets, for object constants, value constants, and atomic concept, value-domain, attribute, role symbols, respectively. Concept, role, attribute, and value-domain expressions are formed according to the following syntax:

$$\begin{array}{ll} B \longrightarrow A \mid \exists Q \mid \delta(U) & E \longrightarrow \rho(U) \\ C \longrightarrow B \mid \neg B & F \longrightarrow \top_D \mid T_1 \mid \cdots \mid T_n \\ Q \longrightarrow P \mid P^- & V \longrightarrow U \mid \neg U \\ R \longrightarrow Q \mid \neg Q & \end{array}$$

where A, P, and U are symbols in Γ denoting an atomic concept, an atomic role and an atomic attribute respectively, $\top_D$ is the universal value-domain, and $T_1, \ldots, T_n$ are symbols in Γ for atomic value-domains. The expression P^- denotes the inverse of an atomic role, $\delta(U)$ denotes the *domain* of U, i.e., the set of objects that U relates to values, $\rho(U)$ denotes the *range* of U, i.e., the set of values that U relates to objects.

A *DL-Lite*$_{\mathcal{A}}$ KB is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is the TBox and $\mathcal{A}$ the ABox. The TBox $\mathcal{T}$ is a finite set of assertions of the form

$$B \sqsubseteq C \qquad Q \sqsubseteq R \qquad E \sqsubseteq F \qquad U \sqsubseteq V \qquad (\mathsf{funct}\ Q) \qquad (\mathsf{funct}\ U)$$

From left to right, the first four assertions denote inclusions between concepts, roles, value-domains, and attributes, respectively. The last two assertions denote functionality on roles and on attributes. *DL-Lite*$_{\mathcal{A}}$ TBoxes are subject to the restriction that roles and attributes occurring in functionality assertions cannot be specialized (i.e., they cannot occur in the right-hand side of inclusions).

A *DL-Lite*$_{\mathcal{A}}$ ABox $\mathcal{A}$ is a finite set of assertions of the form $A(a)$, $P(a, b)$, and $U(a, v)$, where A, P, and U are as above, a and b are object constants from Γ_O, and v is a value constant from Γ_V.

Example 1. We consider a simple *DL-Lite*$_{\mathcal{A}}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ describing the "Formula One Teams" domain, where the TBox $\mathcal{T}$ is constituted by the following assertions:

$$\begin{array}{lll} \mathsf{Mechanic} \sqsubseteq \mathsf{TeamMember} & \mathsf{Driver} \sqsubseteq \mathsf{TeamMember} & \mathsf{Driver} \sqsubseteq \neg \mathsf{Mechanic} \\ \exists \mathsf{drives} \sqsubseteq \mathsf{Driver} & \exists \mathsf{drives}^- \sqsubseteq \mathsf{Car} & (\mathsf{funct}\ \mathsf{drives}) \end{array}$$

In words, $\mathcal{T}$ specifies that drivers and mechanics are team members, but drivers are not mechanics (note that $\mathsf{Driver} \sqsubseteq \neg\mathsf{Mechanic}$ is called a disjointness assertion, because it states that the two concepts Driver and $\mathsf{Mechanic}$ are dosjoint). Moreover, the role drives has Driver as domain and Car as range, and it is also functional, i.e., every driver can drive at most one car. The ABox $\mathcal{A} = \{\mathsf{Driver}(felipe), \mathsf{TeamMember}(felipe), \mathsf{drives}(felipe, ferrari)\}$ asserts that $felipe$ is both a driver and a team member, and that he drives the car $ferrari$.

The semantics of a *DL-Lite*$_{\mathcal{A}}$ KB is given in terms of FOL interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is the interpretation domain and $\cdot^{\mathcal{I}}$ is the interpretation function. In particular, $\Delta^{\mathcal{I}}$ is a non-empty set partitioned into Δ_V and $\Delta^{\mathcal{I}}_O$, where $\Delta^{\mathcal{I}}_O$ is the subset of Δ used to interpret object constants in Γ_O, and Δ_V is the subset of Δ used to interpret data values. In other words, for every $c \in \Gamma_O$, $c^{\mathcal{I}} \in \Delta_O$ and for every $f \in \Gamma_V$, $f^{\mathcal{I}} \in \Delta_V$. The interpretation function $\cdot^{\mathcal{I}}$ is defined as follows.

- For every $c \in \Gamma_O$, $c^{\mathcal{I}} \in \Delta_O$, and for every $f \in \Gamma_V$, $f^{\mathcal{I}} \in \Delta_V$.
- For all $d_1, d_2 \in \Gamma_O \cup \Gamma_V$, $d_1^{\mathcal{I}} \neq d_2^{\mathcal{I}}$ (i.e., interpretations for *DL-Lite* follow the unique name assumption).
- For all $1 \leq i \leq n$, T_i is an unbound set of values.
- The following equations are satisfied by $\cdot^{\mathcal{I}}$:

$$
\begin{array}{llll}
A^{\mathcal{I}} & \subseteq \Delta_O^{\mathcal{I}} & P^{\mathcal{I}} & \subseteq \Delta_O^{\mathcal{I}} \times \Delta_O^{\mathcal{I}} \\
(\delta(U))^{\mathcal{I}} & = \{\, o \mid \exists v.\, (o,v) \in U^{\mathcal{I}} \,\} & (P^-)^{\mathcal{I}} & = \{\, (o,o') \mid (o',o) \in P^{\mathcal{I}} \,\} \\
(\exists Q)^{\mathcal{I}} & = \{\, o \mid \exists o'.\, (o,o') \in Q^{\mathcal{I}} \,\} & (\neg Q)^{\mathcal{I}} & = (\Delta_O^{\mathcal{I}} \times \Delta_O^{\mathcal{I}}) \setminus Q^{\mathcal{I}} \\
(\neg B)^{\mathcal{I}} & = \Delta_O^{\mathcal{I}} \setminus B^{\mathcal{I}} & U^{\mathcal{I}} & \subseteq \Delta_O^{\mathcal{I}} \times \Delta_V \\
(\rho(U))^{\mathcal{I}} & = \{\, v \mid \exists o.\, (o,v) \in U^{\mathcal{I}} \,\} & (\neg U)^{\mathcal{I}} & = (\Delta_O^{\mathcal{I}} \times \Delta_V) \setminus U^{\mathcal{I}} \\
T_i^{\mathcal{I}} & \subseteq \top_D^{\mathcal{I}} = \Delta_V \ (1 \leq i \leq n) & T_i \cap T_j & = \emptyset \quad (1 \leq i,j \leq n)
\end{array}
$$

An interpretation $\mathcal{I}$ satisfies a concept (resp., role) inclusion assertion $B \sqsubseteq C$ (resp., $Q \sqsubseteq R$) if $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ (resp., $Q^{\mathcal{I}} \subseteq R^{\mathcal{I}}$), and satisfies a role functionality assertion $(\mathsf{funct}\ Q)$ if, for each $o, o', o'' \in \Delta_O^{\mathcal{I}}$, we have that $(o, o') \in Q^{\mathcal{I}}$ and $(o, o'') \in Q^{\mathcal{I}}$ implies $o' = o''$. The semantics for attribute and value-domain inclusion assertions, and for functionality assertions over attributes can be defined analogously. Finally, $\mathcal{I}$ satisfies ABox assertions $A(a)$, $P(a, b)$ and $U(a, v)$ if $a^{\mathcal{I}} \in A^{\mathcal{I}}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$ and $(a^{\mathcal{I}}, v^{\mathcal{I}}) \in U^{\mathcal{I}}$ respectively.

In the following, we are interested in particular in the problem of answering queries posed to a *DL-Lite*$_{\mathcal{A}}$-KB. More specifically, we deal with the problem of establishing whether a *DL-Lite*$_{\mathcal{A}}$ 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 attribute name, and z, z' are terms. Notice that all the results we achieve about this reasoning task can be easily extended in the standard way to the presence of free variables in queries (see e.g. [7]).

In the rest of this paper 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 [4, 13] 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.

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:*

1. $\mathcal{A}' \subseteq \mathcal{A}$,
2. *Mod*$(\langle \mathcal{T}, \mathcal{A}' \rangle) \neq \emptyset$,
3. *there exist 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 can now give the definition of ABox repair model.

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 2. Consider the *DL-Lite*$_{\mathcal{A}}$ knowledge base $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$, where $\mathcal{T}$ is the TBox of the KB presented in the Example 1, and $\mathcal{A}'$ is the ABox constituted by the set of assertions:

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

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:

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

Note that to obtain $AR\text{-}rep_1$ it is sufficient to remove $\mathsf{Mechanic}(felipe)$ from $\mathcal{A}$, whereas to obtain $AR\text{-}rep_2$, we need to remove from $\mathcal{A}$ both $\mathsf{Driver}(felipe)$, which is obvious, and $\mathsf{Driver}(felipe, ferrari)$, which, together with the TBox assertion $\exists\mathsf{drives} \sqsubseteq \mathsf{Driver}$, implies $\mathsf{Driver}(felipe)$.

The AR-semantics given above in fact coincides with the inconsistency-tolerant semantics for DL KBs presented in [10], and with the loosely-sound semantics studied in [3] 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 3. 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 2, 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:

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

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:*

1. *$\mathcal{A}' \subseteq clc(\mathcal{K})$,*
2. *$Mod(\langle \mathcal{T}, \mathcal{A}' \rangle) \neq \emptyset$,*
3. *there exist no $\mathcal{A}'' \subseteq clc(\mathcal{K})$ such that*
   (a) *$Mod(\langle \mathcal{T}, \mathcal{A}'' \rangle) \neq \emptyset$, and*
   (b) *it is either $\mathcal{A}'' \cap \mathcal{A} \supset \mathcal{A}' \cap \mathcal{A}$ or $\mathcal{A}'' \cap \mathcal{A} = \mathcal{A}' \cap \mathcal{A}$ and $\mathcal{A}'' \supset \mathcal{A}'$.*

The set of CAR-repairs for $\mathcal{K}$ is denoted by $CAR\text{-}Rep(\mathcal{T}, \mathcal{A})$.

Intuitively, a CAR-repair is a subset of $clc(\mathcal{K})$ consistent with $\mathcal{T}$ that "maximally preserves" the ABox $\mathcal{A}$. In particular, condition 3 states that we prefer $\mathcal{A}'$ to any other $\mathcal{A}_R \subseteq clc(\mathcal{K})$ consistent with $\mathcal{T}$ such that $\mathcal{A}_R \cap \mathcal{A} \subset \mathcal{A}' \cap \mathcal{A}$ (i.e., $\mathcal{A}_R$ maintains a smaller subset of $\mathcal{A}$ with respect to $\mathcal{A}'$). Then, among those $\mathcal{A}_R$ having the same intersection with $\mathcal{A}$, we prefer the ones that contain as much assertions of $clc(\mathcal{K})$ as possible.

The set of CAR-models of a KB $\mathcal{K}$, denoted $CAR\text{-}Mod(\mathcal{K})$, is defined analogously to AR-models (cf. Definition 2). Also, CAR-entailment, denoted $\models_{CAR}$, is analogous to AR-entailment (cf. Definition 3).

Example 4. Consider the two KBs $\mathcal{K}'$ and $\mathcal{K}''$ presented in the Example 2 and Example 3. It is easy to see that both $CAR\text{-}Rep(\mathcal{K}')$ and $CAR\text{-}Rep(\mathcal{K}'')$ are constituted by the two sets below:

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

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 3.

The above example shows also that there are sentences entailed by a KB under the CAR-semantics that are not entailed under the AR-semantics. Conversely, we can show that the AR-semantics is a sound approximation of the CAR-semantics, i.e., for any KB $\mathcal{K}$ $CAR\text{-}Mod(\mathcal{K}) \subseteq AR\text{-}Mod(\mathcal{K})$, implying that the logical consequences of $\mathcal{K}$ under the AR-semantics are contained in the logical consequences of $\mathcal{K}$ under the CAR-semantics, as stated by the following theorem.

Theorem 1. *Let $\mathcal{K}$ be a DL KB, and ϕ a first-order sentence. Then, $\mathcal{K} \models_{AR} \phi$ implies $\mathcal{K} \models_{CAR} \phi$.*

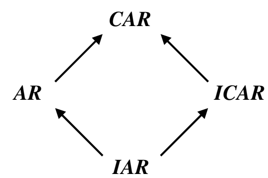

Fig. 1. Partial order over the inconsistency tolerant semantics

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 the area of belief revision and update [15, 6].

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).

Example 5. Consider the KB $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ presented in Example 2. 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 4, *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.

Furthermore, an analogous of Theorem 1 holds also for the "intersection" semantics.

Theorem 2. *Let $\mathcal{K}$ be a DL KB, and ϕ a first-order sentence. Then, $\mathcal{K} \models_{IAR} \alpha$ implies $\mathcal{K} \models_{ICAR} \alpha$.*

Also in this case one can easily see that the converse implication does not hold. It is sufficient to look again at Example 5, where $\mathsf{Car}(ferrari)$ is entailed by $\mathcal{K}'$ under the $ICAR$-semantics, but it is not entailed under the IAR-semantics.

From all the above results it follows that the AR-, CAR-, IAR-, and $ICAR$-semantics form a partial order, where the CAR-semantics is the upper bound, the IAR-semantics is the lower bound, whereas the $ICAR$-semantics and the AR-semantics are incomparable (see Figure 1). In other words, the IAR-semantics is a sound approximation of all the semantics, while the CAR-semantics is the one which is able to derive the largest set of conclusions from a KB. It can also easily be shown that the AR-semantics and the $ICAR$-semantics are incomparable.

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. We will also consider instance checking, which is a restricted form of UCQ entailment. As we said before, in our analysis we will focus on data complexity.

We start by considering the AR-semantics. It is known that UCQ entailment is intractable under this semantics [10]. Here, we strengthen this result, and show that instance checking under the AR-semantics is already coNP-hard 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 $C_1 \sqsubseteq C_2$, $C_1 \sqsubseteq \neg C_2$ (for more details, see [4]).

Theorem 3. *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.*

Proof. Membership in coNP follows from coNP-completeness of UCQ entailment under AR-semantics [10, Theorem 1].

We prove hardness with respect to coNP by reducing satisfiability of a 3-CNF formula to the complement of instance checking.

Let ϕ be a 3-CNF, i.e., a formula of the form $\phi = c_1 \wedge \ldots \wedge c_k$ where $c_i = \ell_1^i \vee \ell_2^i \vee \ell_3^i$ for every i such that $1 \leq i \leq k$ (every ℓ_j^i is a propositional literal). Let $a_1, \ldots, a_n$ be the propositional variable symbols occurring in ϕ.

We define the following TBox $\mathcal{T}$ (which does not depend on ϕ):

$\exists R^- \sqsubseteq \neg \exists LT_1^-$	$\exists R \sqsubseteq Unsat$	$\exists LT_2 \sqsubseteq \neg \exists LF_2$
$\exists R^- \sqsubseteq \neg \exists LF_1^-$	$\exists LT_1 \sqsubseteq \neg \exists LF_1$	$\exists LT_2 \sqsubseteq \neg \exists LF_3$
$\exists R^- \sqsubseteq \neg \exists LT_2^-$	$\exists LT_1 \sqsubseteq \neg \exists LF_2$	$\exists LF_2 \sqsubseteq \neg \exists LT_3$
$\exists R^- \sqsubseteq \neg \exists LF_2^-$	$\exists LT_1 \sqsubseteq \neg \exists LF_3$	$\exists LT_3 \sqsubseteq \neg \exists LF_3$
$\exists R^- \sqsubseteq \neg \exists LT_3^-$	$\exists LF_1 \sqsubseteq \neg \exists LT_2$	
$\exists R^- \sqsubseteq \neg \exists LF_3^-$	$\exists LF_1 \sqsubseteq \neg \exists LT_3$	

Then, we define the following ABox $\mathcal{A}_\phi$ (which depends on ϕ):

$$\{R(a,c_1),\ldots,R(a,c_k)\} \cup \bigcup_{i=1}^{k} \bigcup_{j=1}^{3} \{Polarity(\ell_j^i)(Atom(\ell_j^i), c_i)\}$$

where: (i) $Polarity(\ell_j^i) = LT_j$ if ℓ_j^i is a positive literal, while $Polarity(\ell_j^i) = LF_j$ if ℓ_j^i is a negative literal; (ii) $Atom(\ell_j^i)$ denotes the propositional variable symbol occurring in literal ℓ_j^i. For instance, if $\phi = (a_1 \vee \neg a_2 \vee a_3) \wedge (\neg a_3 \vee a_4 \vee \neg a_1)$, the ABox $\mathcal{A}_\phi$ is the following:

$$\{R(a,c_1), R(a,c_2)\} \cup$$
$$\{LT_1(a_1,c_1), LF_2(a_2,c_1), LT_3(a_3,c_1), LF_1(a_3,c_2), LT_2(a_4,c_2), LF_3(a_1,c_2)\}$$

We now prove that $\langle \mathcal{T}, \mathcal{A}_\phi \rangle \not\models_{AR} Unsat(a)$ iff ϕ is satisfiable.

First, suppose ϕ is satisfiable, and let $\mathcal{I}$ be a model for ϕ. Then, let $\mathcal{A}'$ be the following subset of $\mathcal{A}_\phi$:

$$\{LT_j(a_i,c_h) \mid LT_j(a_i,c_h) \in \mathcal{A}_\phi \text{ and } \mathcal{I} \models a_i\} \cup$$
$$\{LF_j(a_i,c_h) \mid LF_j(a_i,c_h) \in \mathcal{A}_\phi \text{ and } \mathcal{I} \not\models a_i\}$$

It is immediate to verify that $\mathcal{A}'$ is a maximal $\mathcal{T}$-consistent subset of $\mathcal{A}_\phi$: in particular, since $\mathcal{I} \models \phi$, if we add any $R(a,c_i)$ to $\mathcal{A}'$, the resulting ABox is inconsistent. Therefore, $\mathcal{A}'$ is an AR-repair of $\langle \mathcal{T}, \mathcal{A}_\phi \rangle$. Moreover, since no $R(a,c_i)$ belongs to $\mathcal{A}'$, it follows that $\langle \mathcal{T}, \mathcal{A}' \rangle \not\models Unsat(a)$, which implies that $\langle \mathcal{T}, \mathcal{A}_\phi \rangle \not\models_{AR} Unsat(a)$.

Conversely, suppose $\langle \mathcal{T}, \mathcal{A}_\phi \rangle \not\models_{AR} Unsat(a)$. Then, there exists an AR-repair $\mathcal{A}'$ of $\langle \mathcal{T}, \mathcal{A}_\phi \rangle$ such that $\langle \mathcal{T}, \mathcal{A}' \rangle \not\models Unsat(a)$. Of course, no $R(a,c_i)$ belongs to $\mathcal{A}'$. Now, let $\mathcal{I}$ be the propositional interpretation defined as follows: for every a_i, if there exists a fact of the form $LT_j(a_i,c_h) \in \mathcal{A}'$, then $\mathcal{I} \models a_i$, otherwise $\mathcal{I} \not\models a_i$. It is easy to see that, since no $R(a,c_h)$ belongs to $\mathcal{A}'$, for every c_h either there exists a fact of the form $LT_j(a_i,c_h)$ in $\mathcal{A}'$ or there exists a fact of the form $LF_j(a_i,c_h)$ in $\mathcal{A}'$. In both cases, we get $\mathcal{I} \models c_h$. Consequently, $\mathcal{I} \models \phi$. □

Theorem 3 corrects a wrong result presented in [10, Theorem 6], which asserts tractability of AR-entailment of ABox assertions from KBs specified in *DL-Lite*$_\mathcal{F}$, a superset of *DL-Lite*$_{core}$. It turns out that, while the algorithm presented in [10] (on which the above cited Theorem 6 was based) is actually unable to deal with general TBoxes, such a technique can be adapted to prove that AR-entailment of ABox assertions is tractable for *DL-Lite*$_\mathcal{A}$ KBs without TBox disjointness assertions.

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

Theorem 4. *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.*

Proof. The proof of coNP-hardness is obtained by a slight modification of the reduction from 3-CNF satisfiability in the proof of Theorem 3. To prove membership in coNP, we first prove that, when $\mathcal{K}$ is a *DL-Lite*$_\mathcal{A}$ KB, *clc*$(\mathcal{K})$ can be computed in polynomial time, which is an immediate consequence of the fact that

$$clc(\mathcal{K}) = \{\alpha \mid \alpha \in HB(\mathcal{K}) \text{ and } \alpha_i \in \mathcal{A} \text{ and } \langle \mathcal{T}, \{\alpha_i\} \rangle \models \alpha \text{ and } \langle \mathcal{T}, \{\alpha_i\} \rangle \text{ satisfiable}\}$$

Now, membership in coNP follows from the following facts: (i) $\langle \mathcal{T}, \mathcal{A} \rangle \models_{CAR} q$ iff $\langle \mathcal{T}, clc(\mathcal{T}, \mathcal{A}) \rangle \models_{AR} q$; (ii) $clc(\mathcal{T}, \mathcal{A})$ can be computed in polynomial time; (iii) Theorem 3. □

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 following algorithm *Compute-IAR-Repair* for computing the IAR-repair of a *DL-Lite*$_\mathcal{A}$ KB $\mathcal{K}$.

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

The algorithm is very simple: it computes a set $\mathcal{D}$ of ABox assertions in $\mathcal{A}$ which must be eliminated from the IAR-repair of $\mathcal{K}$.

Lemma 1. *Let $\mathcal{K}$ be a DL-Lite$_\mathcal{A}$ KB. Then, IAR-Rep$(\mathcal{K})$ = $\{$Compute-IAR-Repair$(\mathcal{K})\}$.*

Proof. (sketch) The proof is based on the following property, not difficult to verify, which is due to the form of the TBox assertions allowed in *DL-Lite*$_\mathcal{A}$: every ABox assertion α that does not belong to at least one AR-repair of $\mathcal{K}$ satisfies one of the following conditions: (i) α is such that the KB $\langle \mathcal{T}, \{\alpha\} \rangle$ is unsatisfiable; (ii) α is such that there exists another ABox assertion α' such that the KB $\langle \mathcal{T}, \{\alpha, \alpha'\} \rangle$ is unsatisfiable and α' does not satisfy the previous condition (i). Therefore, at the end of the execution of the algorithm, the set $\mathcal{D}$ contains every ABox assertion α that does not belong any AR-repair of $\mathcal{K}$. Hence $\mathcal{A} - \mathcal{D}$ is the IAR-repair of $\mathcal{K}$.

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

Theorem 5. *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.*

Proof. By Lemma 1, the ABox returned by *Compute-IAR-Repair*$(\mathcal{K})$ is the IAR-repair of $\mathcal{K}$. Then, by definition of IAR-semantics, we have that, for every UCQ Q, $\mathcal{K} \models_{IAR} Q$ iff $\langle \mathcal{T}, \mathcal{A}' \rangle \models Q$ where $\mathcal{A}'$ is the IAR-repair of $\mathcal{K}$. From the fact that the algorithm *Compute-IAR-Repair*$(\mathcal{K})$ runs in polynomial time and from tractability of UCQ entailment in *DL-Lite*$_{\mathcal{A}}$ [13], the claim follows. □

We now turn our attention to the $ICAR$-semantics and present the algorithm *Compute-ICAR-Repair* for computing the $ICAR$-repair of a *DL-Lite*$_{\mathcal{A}}$ KB $\mathcal{K}$.

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

The algorithm is analogous to the previous algorithm *Compute-IAR-Repair*. The main differences are the following: (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 6. *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.*

Proof. First, we prove that the ABox returned by *Compute-ICAR-Repair*$(\mathcal{K})$ is the $ICAR$-repair of $\mathcal{K}$. This follows from the following property: let $\alpha_1, \alpha_2 \in$ *clc*$(\langle \mathcal{T}, \mathcal{A} \rangle)$ and let $\langle \mathcal{T}, \{\alpha_1, \alpha_2\} \rangle$ be unsatisfiable. Then, let $\beta_1 \in \mathcal{A}$ be such that $\langle \mathcal{T}, \{\beta_1\} \rangle \models \alpha_1$, and let $\beta_2 \in \mathcal{A}$ be such $\langle \mathcal{T}, \{\beta_2\} \rangle \models \alpha_2$ (such facts β_1, β_2 always exist). Now, it is immediate to verify that $\langle \mathcal{T}, \{\beta_1, \beta_2\} \rangle$ is unsatisfiable. Moreover, by definition of *clc*$(\langle \mathcal{T}, \mathcal{A} \rangle)$ we have that (i) $\langle \mathcal{T}, \{\beta_1\} \rangle$ is satisfiable, and (ii) $\langle \mathcal{T}, \{\beta_2\} \rangle$ is satisfiable. Now, (i) immediately implies that there exists a CAR-repair $\mathcal{A}'$ of $\mathcal{K}$ that contains β_1, and hence α_1 since $\mathcal{A}'$ is deductively closed. Consequently, α_2 cannot belong to the $\mathcal{A}'$ (since $\langle \mathcal{T}, \mathcal{A}' \rangle$ must be satisfiable, and hence α_2 does not belong to the intersection of all the CAR-repairs of $\mathcal{K}$. In the same way, from (ii) we derive that α_1 does not belong to the intersection of all the CAR-repairs of $\mathcal{K}$.

Now, as shown in the proof of Theorem 4, *clc*$(\mathcal{K})$ can be computed in polynomial time, which implies that the algorithm *Compute-ICAR-Repair*$(\mathcal{K})$ runs in polynomial time. Moreover, by definition of $ICAR$-semantics, we have that, for every UCQ

	AR semantics	CAR semantics	IAR semantics	$ICAR$ semantics
instance checking	coNP-complete	in PTIME	in PTIME	in PTIME
UCQ entailment	coNP-complete [10]	coNP-complete	in PTIME	in PTIME

Fig. 2. Data complexity of UCQ entailment over *DL-Lite*$_{\mathcal{A}}$ KBs under inconsistency-tolerant semantics.

Q, $\mathcal{K} \models_{ICAR} Q$ iff $\langle \mathcal{T}, \mathcal{A}' \rangle \models Q$ where $\mathcal{A}'$ is the $ICAR$-repair of $\mathcal{K}$. Hence, from tractability of UCQ entailment in *DL-Lite*$_{\mathcal{A}}$, the thesis follows. □

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

Lemma 2. *Let $\mathcal{K}$ be a DL-Lite$_{\mathcal{A}}$ KB, and let α be an ABox assertion. Then, $\mathcal{K} \models_{CAR} \alpha$ iff $\mathcal{K} \models_{ICAR} \alpha$.*

Proof. $\mathcal{K} \models_{CAR} \alpha$ if $\mathcal{K} \models_{ICAR} \alpha$ follows from the fact that the $ICAR$-semantics is a sound approximation of the CAR-semantics. As for the converse, since every CAR-repair is deductively closed, it follows that $\mathcal{K} \models_{CAR} \alpha$ iff α belongs to the intersection of all the CAR-repairs of $\langle \mathcal{T}, \mathcal{A} \rangle$, i.e., to the $ICAR$-repair of $\mathcal{K}$. □

The above property and Theorem 6 allow us to establish tractability of instance checking under the CAR-semantics.

Theorem 7. *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 2 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 3. In Figure 2 we summarize the complexity results presented in this section.

5 Conclusions

We have presented an investigation on inconsistency-tolerant reasoning in DLs, with special attention to the *DL-Lite* family. The techniques we have illustrated assume that the TBox is consistent, and therefore consider the case of inconsistencies arising between the TBox and the ABox.

Our approach to inconsistency-tolerance is inspired by the work done on consistent query answering in databases. Indeed, the AR-semantics presented in Section 3 is the direct application of the notion of repair to DL knowledge bases. Motivated by the intractability of inconsistency-tolerant query answering under such semantics, we have investigated several variants of the AR-semantics, with the goal of finding a good compromise between expressive power and complexity of query answering.

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. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
2. L. E. Bertossi, A. Hunter, and T. Schaub, editors. *Inconsistency Tolerance*, volume 3300 of *LNCS*. Springer, 2005.
3. 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.
4. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
5. J. Chomicki. Consistent query answering: Five easy pieces. In *Proc. of ICDT 2007*, volume 4353 of *LNCS*, pages 1–17. Springer, 2007.
6. T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artificial Intelligence*, 57:227–270, 1992.
7. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. In *Proc. of IJCAI 2007*, pages 399–404, 2007.
8. P. Haasa, F. van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In *Proc. of ISWC 2005*, 2005.
9. Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In *Proc. of IJCAI 2005*, pages 454–459, 2003.
10. D. Lembo and M. Ruzzi. Consistent query answering over description logic ontologies. In *Proc. of RR 2007*, 2007.
11. Y. Ma and P. Hitzler. Paraconsistent reasoning for owl 2. In *Proc. of RR 2009*, pages 197–211, 2009.
12. B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *Proc. of WWW 2005*, pages 633–640, 2005.
13. 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.
14. G. Qi and J. Du. Model-based revision operators for terminologies in description logics. In *Proc. of IJCAI 2009*, pages 891–897, 2009.
15. M. Winslett. *Updating Logical Databases*. Cambridge University Press, 1990.