

# 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 in the database (i.e., data that do not satisfy the integrity constraints defined for 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 queries 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 equivalent to evaluating queries 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, and *roles*, denoting binary relations between (instances of) concepts. Complex concept and role expressions are constructed starting from a set of atomic concepts and roles by 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 stores 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. In this paper we assume that the ABox assertions are always *atomic*, i.e., they correspond to ground atoms.

Formally, a DL knowledge base  $\mathcal{K}$  is a pair  $\langle \mathcal{T}, \mathcal{A} \rangle$ , where  $\mathcal{T}$  is a TBox and  $\mathcal{A}$  is a ABox, i.e., both include only constructs formed according to the syntax for concept expressions, role expressions, and other assertions allowed by the specific DL. The alphabet of  $\mathcal{K}$  is given by the union of the predicate symbols occurring in  $\mathcal{T}$  and  $\mathcal{A}$  and the constant symbols occurring in  $\mathcal{A}$ . The standard semantics of a DL knowledge base is given in terms of first-order (FOL) interpretations (cf. [1]). We denote with  $Mod(\mathcal{K})$  the set of models of  $\mathcal{K}$ , i.e., the set of FOL interpretations that satisfy the assertions in  $\mathcal{K}$  (the notion of satisfaction depends

on the specific DL language in which  $\mathcal{K}$  is specified). As usual, a KB  $\mathcal{K}$  entails a FOL sentence  $\phi$ , denoted  $\mathcal{K} \models \phi$ , if  $\phi^{\mathcal{I}}$  is true in every  $\mathcal{I} \in \text{Mod}(\mathcal{K})$ .

We now consider the DL  $DL\text{-}Lite_{\mathcal{A}}$ , a member of the  $DL\text{-}Lite$  family [4], which is a family of tractable DLs particularly suited for dealing with KBs with very large ABoxes, and which is at the basis of OWL 2 QL, one of the profile of OWL 2, the official ontology specification language of the World-Wide-Web Consortium (W3C).  $DL\text{-}Lite_{\mathcal{A}}$  distinguishes concepts from *value-domains*, which denote sets of (data) values, and roles from *attributes*, which denote binary relations between objects and values. Concepts, roles, attributes, and value-domains in this DL are formed according to the following syntax:

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

In such rules,  $A$ ,  $P$ , and  $U$  respectively denote an atomic concept name, an atomic role name and an attribute name and  $P^-$  denotes the inverse of an atomic role. Furthermore,  $\delta(U)$  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;  $\top_D$  is the universal value-domain;  $T_1, \dots, T_n$  are  $n$  pairwise disjoint unbounded value-domains.

A  $DL\text{-}Lite_{\mathcal{A}}$  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

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

From left to right, the first four assertions respectively denote inclusions between concepts, roles, value-domains, and attributes. In turn, the last two assertions denote functionality on roles and on attributes. In fact, in  $DL\text{-}Lite_{\mathcal{A}}$  TBoxes we further impose that roles and attributes occurring in functionality assertions cannot be specialized (i.e., they cannot occur in the right-hand side of inclusions).

A  $DL\text{-}Lite_{\mathcal{A}}$  ABox  $\mathcal{A}$  is a finite set of assertions of the forms  $A(a)$ ,  $P(a, b)$ , and  $U(a, v)$ , where  $A$ ,  $P$ , and  $U$  are as above,  $a$  and  $b$  are object constants from an alphabet  $\Gamma_O$ , and  $v$  is a value constant from an alphabet  $\Gamma_V$ . As usual in OWL, the set of object constants is disjoint from that of value constants, that is,  $\Gamma_O \cap \Gamma_V = \emptyset$ .

*Example 1.* Let us consider a simple  $DL\text{-}Lite_{\mathcal{A}}$  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:

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

In words,  $\mathcal{T}$  specifies that drivers and mechanics are team members, but drivers cannot be mechanics, and vice-versa. Moreover, the role *drives* has *Driver* as

domain and Car as range, and it is also functional, i.e., every driver cannot drive more than one car. As ABox we assume to have the set  $\mathcal{A} = \{\text{Driver}(\text{felipe}), \text{TeamMember}(\text{felipe}), \text{drives}(\text{felipe}, \text{ferrari})\}$ . The ABox asserts that *felipe* is both a driver and a team member, and that he drives a *ferrari*.

The semantics of a *DL-Lite<sub>A</sub>* KB is given in terms of FOL interpretations  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ .  $\Delta^{\mathcal{I}}$  is a non-empty domain such that  $\Delta^{\mathcal{I}} = \Delta_V \cup \Delta_O^{\mathcal{I}}$ , where  $\Delta_O^{\mathcal{I}}$  is the domain used to interpret object constants in  $\Gamma_O$ , and  $\Delta_V$  is the fixed domain (disjoint from  $\Delta_O^{\mathcal{I}}$ ) used to interpret data values.  $\cdot^{\mathcal{I}}$  is an interpretation function defined as follows:

$$\begin{array}{ll} A^{\mathcal{I}} & \subseteq \Delta_O^{\mathcal{I}} & P^{\mathcal{I}} & \subseteq \Delta_O^{\mathcal{I}} \times \Delta_O^{\mathcal{I}} \\ (\delta(U))^{\mathcal{I}} & = \{ o \mid \exists v. (o, v) \in U^{\mathcal{I}} \} & (P^-)^{\mathcal{I}} & = \{ (o, o') \mid (o', o) \in P^{\mathcal{I}} \} \\ (\exists Q)^{\mathcal{I}} & = \{ o \mid \exists o'. (o, o') \in Q^{\mathcal{I}} \} & (\neg Q)^{\mathcal{I}} & = (\Delta_O^{\mathcal{I}} \times \Delta_O^{\mathcal{I}}) \setminus Q^{\mathcal{I}} \\ (\neg B)^{\mathcal{I}} & = \Delta_O^{\mathcal{I}} \setminus B^{\mathcal{I}} & U^{\mathcal{I}} & \subseteq \Delta_O^{\mathcal{I}} \times \Delta_V \\ \top_D^{\mathcal{I}} & = \Delta_V & (\neg U)^{\mathcal{I}} & = (\Delta_O^{\mathcal{I}} \times \Delta_V) \setminus U^{\mathcal{I}} \\ (\rho(U))^{\mathcal{I}} & = \{ v \mid \exists o. (o, v) \in U^{\mathcal{I}} \} \end{array}$$

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.,  $Q^{\mathcal{I}} \subseteq R^{\mathcal{I}}$ ). Furthermore, a role functionality assertion (funct  $Q$ ) is satisfied by  $\mathcal{I}$  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. As for the semantics of ABox assertions, we say that  $\mathcal{I}$  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 establishing whether a *DL-Lite<sub>A</sub>* KB entails a *boolean union of conjunctive queries (UCQ)*, i.e., a first order sentence of the form  $\exists \mathbf{y}_1. \text{conj}_1(\mathbf{y}_1) \vee \dots \vee \exists \mathbf{y}_n. \text{conj}_n(\mathbf{y}_n)$ , where  $\mathbf{y}_1, \dots, \mathbf{y}_n$  are terms (i.e., constants or variables), and each  $\text{conj}_i(\mathbf{y}_i)$  is a conjunction of atoms of the form  $A(z)$ ,  $P(z, z')$  and  $U(z, z')$  where  $A$  is a concept name,  $P$  is a role name 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]).

### 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, and 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.  $\text{Mod}(\langle \mathcal{T}, \mathcal{A}' \rangle) \neq \emptyset$
3. *there does not exist  $\mathcal{A}''$  such that  $\mathcal{A}' \subset \mathcal{A}'' \subseteq \mathcal{A}$  and  $\text{Mod}(\langle \mathcal{T}, \mathcal{A}'' \rangle) \neq \emptyset$*

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

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  $\mathcal{A}' \in \text{AR-Rep}(\mathcal{K})$  such that  $\mathcal{I} \models \langle \mathcal{T}, \mathcal{A}' \rangle$ . The set of ABox repair models is denoted by  $\text{AR-Mod}(\mathcal{K})$ .*

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

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

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

This means that *felipe* is a team member and that he is both a driver and a mechanic, moreover this means that *felipe* drives a *ferrari* and that a *ferrari* is a car. It is easy to see that  $\mathcal{K}$  is unsatisfiable, since *felipe* is both a driver and a mechanic.

The set  $\text{AR-Rep}(\mathcal{K}')$  is constituted by the set of  $\mathcal{T}$ -consistent ABoxes:

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

$$AR\text{-}rep_2 = \{\text{Mechanic}(felipe), \text{TeamMember}(felipe)\}.$$

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

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 that we are considering, since each ABox repair stays as close as possible to the original ABox, it has the characteristic to be “syntax-dependent”, in the sense showed in the following example.

*Example 3 (Syntax-dependence of the AR-semantics).* 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:

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

Notice that  $\mathcal{A}''$  can be obtained by adding  $\text{Car}(ferrari)$  to  $\mathcal{A}'$ . Since  $\text{Car}(ferrari)$  is entailed by the KB  $\langle \mathcal{T}, \{\text{drives}(felipe, ferrari)\} \rangle$ , 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  $AR\text{-}Rep(\mathcal{K}'')$  is formed by:

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

$$AR\text{-}rep_4 = \{\text{Mechanic}(felipe), \text{TeamMember}(felipe), \text{Car}(ferrari)\}.$$

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

Depending on the particular setting analyzed and the specific application at hand, the above behavior can be considered non-acceptable. This motivates the definition of a new semantics that does not present such a characteristic. To achieve this aim, the repairs of the new semantics that we propose take into account not only the assertions explicitly included in the ABox, but also those that are implied by the TBox and each 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 ground atoms that can be built over the alphabet of  $\mathcal{K}$ . Then we define the *consistent logical consequences of  $\mathcal{K}$*  as the set  $clc(\mathcal{K}) = \{\alpha \mid \alpha \in HB(\mathcal{K}) \text{ and there exists } S \subseteq \mathcal{A} \text{ such that } Mod(\langle \mathcal{T}, S \rangle) \neq \emptyset \text{ and } \langle \mathcal{T}, S \rangle \models \alpha\}$ . 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.  $\text{Mod}(\langle \mathcal{T}, \mathcal{A}' \rangle) \neq \emptyset$
2.  $\mathcal{A}' \subseteq \text{clc}(\mathcal{K})$
3. there does not exist  $\mathcal{A}'' \subseteq \text{clc}(\mathcal{K})$  such that
  - (a)  $\text{Mod}(\langle \mathcal{T}, \mathcal{A}'' \rangle) \neq \emptyset$ , 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  $\text{CAR-Rep}(\mathcal{T}, \mathcal{A})$ .

Intuitively, a CAR-repair is a subset of  $\text{clc}(\mathcal{K})$  consistent with  $\mathcal{T}$  that “maximally preserves” the ABox  $\mathcal{A}$ . In particular, condition 3 states that we prefer  $\mathcal{A}'$  to any other  $\mathcal{A}_R \subseteq \text{clc}(\mathcal{K})$  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  $\text{clc}(\mathcal{K})$  as possible.

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

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

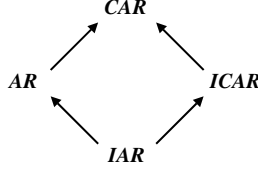
$$\begin{aligned} \text{CAR-rep}_1 &= \{\text{Driver}(\textit{felipe}), \text{drives}(\textit{felipe}, \textit{ferrari}), \text{TeamMember}(\textit{felipe}), \\ &\quad \text{Car}(\textit{ferrari})\}; \\ \text{CAR-rep}_2 &= \{\text{Mechanic}(\textit{felipe}), \text{TeamMember}(\textit{felipe}), \text{Car}(\textit{ferrari})\}. \end{aligned}$$

It follows that both  $\mathcal{K}'$  and  $\mathcal{K}''$  CAR-entails the ground sentence  $\text{Car}(\textit{ferrari})$ , 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 CAR-semantics that are not entailed under AR-semantics. Conversely, we can show that the AR-semantics is a sound approximation of the CAR-semantics, i.e., for any KB  $\mathcal{K}$   $\text{CAR-Mod}(\mathcal{K}) \subseteq \text{AR-Mod}(\mathcal{K})$ . In other words, the logical consequences of  $\mathcal{K}$  under AR-semantics are contained in the logical consequences of  $\mathcal{K}$  under CAR-semantics, as stated by the following theorem.

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

As we will see in the next section, entailment of a union of conjunctive queries from a KB  $\mathcal{K}$  is intractable both under AR-semantics and CAR-semantics. Since this can be an obstacle in the practical use of such semantics, we introduce here two approximations of the AR-semantics and the CAR-semantics, respectively, 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 repairs the intersection of the AR-repairs and of the CAR-repairs,



**Fig. 1.** Partial order over the inconsistency tolerant semantics

respectively. This actually corresponds to follow in our setting the approach WIDTIO (When you are in doubt throw it out), widely adopted 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 a set  $\mathcal{A}'$  of membership assertions such that  $\mathcal{A}' = \bigcap_{\mathcal{A}_i \in AR\text{-Rep}(\mathcal{K})} \mathcal{A}_i$ . The (singleton) set of *IAR*-repairs for  $\mathcal{K}$  is denoted by  $IAR\text{-Rep}(\mathcal{K})$ .

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 a set  $\mathcal{A}'$  of membership assertions such that  $\mathcal{A}' = \bigcap_{\mathcal{A}_i \in CAR\text{-Rep}(\mathcal{K})} \mathcal{A}_i$ . The (singleton) set of *ICAR*-repairs for  $\mathcal{K}$  is denoted by  $ICAR\text{-Rep}(\mathcal{K})$ .

The sets  $IAR\text{-Mod}(\mathcal{K})$  and  $ICAR\text{-Mod}(\mathcal{K})$  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 (IAR Semantics and ICAR Semantics).* Consider the KB  $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$  presented in Example 2. Then  $IAR\text{-Rep}(\mathcal{K}')$  is the singleton formed by the ABox  $IAR\text{-rep} = AR\text{-rep}_1 \cap AR\text{-rep}_2 = \{\text{TeamMember}(felipe)\}$ . In turn, referring to Example 4,  $ICAR\text{-Rep}(\mathcal{K}')$  is the singleton formed by the ABox  $ICAR\text{-rep}_1 = \{\text{TeamMember}(felipe), \text{Car}(ferrari)\} = CAR\text{-rep}_1 \cap CAR\text{-rep}_2$ .

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 does not always hold. For instance, the sentence  $\text{Driver}(felipe)$  is entailed by  $\mathcal{K} = \langle \mathcal{T}, \{\text{drives}(felipe, ferrari), \text{drives}(felipe, mcLaren)\} \rangle$ , where  $\mathcal{T}$  is the TBox of Example 1, under *AR*-semantics, but it is not entailed under *IAR*-semantics.

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

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

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

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<sub>core</sub>*. We recall that *DL-Lite<sub>core</sub>* is the less expressive logic in the *DL-Lite* family, more precisely 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<sub>core</sub>* KB and let  $\alpha$  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 3-colorability to (the complement of) instance checking.

Let  $G = (V, E)$  be an undirected graph with set of vertices  $V$  and the set of edges  $E \subset V \times V$ . Let  $\ell$  be a labeling over the edges in  $E$ , i.e.,  $\ell : E \rightarrow L$  is a bijective function and  $L$  is a set of symbols disjoint from  $V$ .

We define the *DL-Lite<sub>core</sub>* KB  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  in which  $\mathcal{T}$  is the TBox  $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3 \cup \mathcal{T}_4 \cup \mathcal{T}_5$ , where:

$$\begin{aligned}
\mathcal{T}_1 &= \{\exists B_1 \sqsubseteq \neg \exists B_2, \exists G_1 \sqsubseteq \neg \exists G_2, \exists R_1 \sqsubseteq \neg \exists R_2\} \\
\mathcal{T}_2 &= \{\exists B_1^- \sqsubseteq \neg \exists G_1^-, \exists B_1^- \sqsubseteq \neg \exists G_2^-, \exists B_1^- \sqsubseteq \neg \exists R_1^-, \exists B_1^- \sqsubseteq \neg \exists R_2^-, \\
&\quad \exists B_2^- \sqsubseteq \neg \exists G_1^-, \exists B_2^- \sqsubseteq \neg \exists G_2^-, \exists B_2^- \sqsubseteq \neg \exists R_1^-, \exists B_2^- \sqsubseteq \neg \exists R_2^-, \\
&\quad \exists G_1^- \sqsubseteq \neg \exists B_1^-, \exists G_1^- \sqsubseteq \neg \exists B_2^-, \exists G_1^- \sqsubseteq \neg \exists R_1^-, \exists G_1^- \sqsubseteq \neg \exists R_2^-, \\
&\quad \exists G_2^- \sqsubseteq \neg \exists B_1^-, \exists G_2^- \sqsubseteq \neg \exists B_2^-, \exists G_2^- \sqsubseteq \neg \exists R_1^-, \exists G_2^- \sqsubseteq \neg \exists R_2^-, \\
&\quad \exists R_1^- \sqsubseteq \neg \exists B_1^-, \exists R_1^- \sqsubseteq \neg \exists B_2^-, \exists R_1^- \sqsubseteq \neg \exists G_1^-, \exists R_1^- \sqsubseteq \neg \exists G_2^-, \\
&\quad \exists R_2^- \sqsubseteq \neg \exists B_1^-, \exists R_2^- \sqsubseteq \neg \exists B_2^-, \exists R_2^- \sqsubseteq \neg \exists G_1^-, \exists R_2^- \sqsubseteq \neg \exists G_2^-\} \\
\mathcal{T}_3 &= \{\exists B_1 \sqsubseteq \neg E_1, \exists G_1 \sqsubseteq \neg E_1, \exists R_1 \sqsubseteq \neg E_1, \\
&\quad \exists B_2 \sqsubseteq \neg E_2, \exists G_2 \sqsubseteq \neg E_2, \exists R_2 \sqsubseteq \neg E_2\} \\
\mathcal{T}_4 &= \{\exists S_1 \sqsubseteq E_1, \exists S_2 \sqsubseteq E_2\} \\
\mathcal{T}_5 &= \{\exists S_1^- \sqsubseteq C, \exists S_2^- \sqsubseteq C\}
\end{aligned}$$

and  $\mathcal{A}$  is the following ABox:

$$\begin{aligned}
&\{B_1(e, v_1), B_2(e, v_2), G_1(e, v_1), G_2(e, v_2), R_1(e, v_1), R_2(e, v_2), \\
&\quad E_1(e), E_2(e), S_1(e, f), S_2(e, f) \mid \langle v_1, v_2 \rangle \in E \text{ and } e = \ell(\langle v_1, v_2 \rangle)\}
\end{aligned}$$

We prove that  $\mathcal{K} \not\models_{AR} C(f)$  iff  $G$  is 3-colorable.

If  $G$  is 3-colorable, then let  $\mathcal{S}$  be the subset of  $\mathcal{A}$  corresponding to a 3-coloring  $col$  (where  $col$  is a function  $col : V \rightarrow \{R, G, B\}$ ), i.e.,

$$\begin{aligned}
\mathcal{S} &= \{COL_1(e, v) \mid v \in V \text{ and } COL = col(v) \text{ and } e = \langle v, v' \rangle \in E\} \cup \\
&\quad \{COL_2(e, v) \mid v \in V \text{ and } COL = col(v) \text{ and } e = \langle v', v \rangle \in E\}
\end{aligned}$$

Since  $\mathcal{S}$  does not contain any fact of the form  $E_1(e)$  or  $E_2(e)$  or  $S_1(e, f)$  or  $S_2(e, f)$ , it follows that  $\mathcal{S}$  is consistent with  $\mathcal{T}_3 \cup \mathcal{T}_4 \cup \mathcal{T}_5$ . Then, it is immediate to verify that, since  $col$  is a 3-coloring of  $G$ ,  $\mathcal{S}$  is consistent with  $\mathcal{T}_1 \cup \mathcal{T}_2$ : in fact, by definition of  $\mathcal{S}$ , since  $col$  is a 3-coloring, there is no edge  $e$  whose associated vertices have the same color, hence  $\mathcal{S}$  is consistent with  $\mathcal{T}_1$ ; moreover, every vertex has only one color, i.e., in  $\mathcal{S}$  for every  $v$  the facts in which  $v$  occurs are either (i) all of the form  $B_1(e, v)$  or  $B_2(e', v)$ ; or (ii) they are all of the form  $G_1(e, v)$  or  $G_2(e', v)$ ; or (iii) they are all of the form  $R_1(e, v)$  or  $R_2(e', v)$ . Hence,  $\mathcal{S}$  is consistent with  $\mathcal{T}_2$ . Therefore,  $\mathcal{S}$  is consistent with  $\mathcal{T}$ . Moreover,  $\mathcal{S}$  is a maximal subset of  $\mathcal{A}$  consistent with  $\mathcal{T}$ , since no fact of the form  $E_1(e)$  or  $E_2(e)$  can be added to  $\mathcal{S}$  without making the set of facts inconsistent with  $\mathcal{T}_3$ , hence no fact of the form  $S_1(e, f)$  or  $S_2(e, f)$  can be added to  $\mathcal{S}$  without making the set of facts inconsistent with  $\mathcal{T}_4$ . Then, since no fact of the form  $S_1(e, f)$  or  $S_2(e, f)$  is in  $\mathcal{S}$ , it follows that  $\langle \mathcal{T}, \mathcal{S} \rangle \not\models C(f)$ , consequently  $\mathcal{K} \not\models_{AR} C(f)$ .

Conversely, suppose  $\mathcal{K} \models_{AR} C(f)$ . Then, for every maximal subset  $\mathcal{S}$  of  $\mathcal{A}$  consistent with  $\mathcal{T}$ ,  $\langle \mathcal{T}, \mathcal{S} \rangle \models C(f)$ , which immediately implies that there is a fact of the form  $S_1(e, f)$  or  $S_2(e, f)$  in every such  $\mathcal{S}$  (because the only way to force  $C(f)$  to hold in all models of  $\langle \mathcal{T}, \mathcal{S} \rangle$  is through the inclusions in  $\mathcal{T}_5$  and the presence in  $\mathcal{S}$  of a fact of the form  $S_1(e, f)$  or  $S_2(e, f)$ ). But we have proved above that, if  $G$  is 3-colorable, then there exists a maximal subset of  $\mathcal{A}$  consistent with

$\mathcal{T}$  which does not contain any fact of the form  $S_1(e, f)$  or  $S_2(e, f)$ . Therefore,  $G$  is not 3-colorable.  $\square$

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 inclusion assertions with negation in the right-hand side.

Then, 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 is obtained by a slight modification of the reduction from 3-colorability in the proof of Theorem 3.  $\square$

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 on 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* that computes 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:** IAR-repair of  $\mathcal{K}$   
**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 of ABox assertions  $\mathcal{D}$  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  $DL\text{-Lite}_{\mathcal{A}}$ : every ABox assertion  $\alpha$  that does not belong to at least one  $AR$ -repair of  $\mathcal{K}$  satisfies one of the following conditions: (i)  $\alpha$  is such that the KB  $\langle \mathcal{T}, \{\alpha\} \rangle$  is unsatisfiable; (ii)  $\alpha$  is such that there exists another ABox assertion  $\alpha'$  such that the KB  $\langle \mathcal{T}, \{\alpha, \alpha'\} \rangle$  is unsatisfiable and  $\alpha'$  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  $\alpha$  that does not belong to at least one  $AR$ -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  $DL\text{-Lite}_{\mathcal{A}}$  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  $Compute\text{-}IAR\text{-}Repair(\mathcal{K})$  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  $Compute\text{-}IAR\text{-}Repair(\mathcal{K})$  runs in polynomial time and from tractability of UCQ entailment in  $DL\text{-Lite}_{\mathcal{A}}$  [13], the claim follows.  $\square$

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

**Algorithm**  $Compute\text{-}ICAR\text{-}Repair(\mathcal{K})$

**input:**  $DL\text{-Lite}_{\mathcal{A}}$  KB  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

**output:**  $ICAR$ -repair of  $\mathcal{K}$

**begin**

**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  $Compute\text{-}IAR\text{-}Repair$ . The main differences are the following: (i) the algorithm  $Compute\text{-}ICAR\text{-}Repair$  returns (and operates on) a subset of  $clc(\mathcal{K})$ , while the algorithm  $Compute\text{-}IAR\text{-}Repair$  returns a subset of the original ABox  $\mathcal{A}$ ; (ii) differently from the algorithm  $Compute\text{-}IAR\text{-}Repair$ , the algorithm  $Compute\text{-}ICAR\text{-}Repair$  does not need to eliminate ABox assertions  $\alpha$  such that  $\langle \mathcal{T}, \{\alpha\} \rangle$  is unsatisfiable, since such facts cannot occur in  $clc(\mathcal{K})$ .

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

**Theorem 6.** *Let  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  be a  $DL\text{-Lite}_{\mathcal{A}}$  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  $\alpha_1, \alpha_2 \in \text{clc}(\langle \mathcal{T}, \mathcal{A} \rangle)$  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  $\beta_1, \beta_2$  always exist). Now, it is immediate to verify that  $\langle \mathcal{T}, \{\beta_1, \beta_2\} \rangle$  is unsatisfiable. Moreover, by definition of  $\text{clc}(\langle \mathcal{T}, \mathcal{A} \rangle)$  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  $\beta_1$ , and hence  $\alpha_1$  since  $\mathcal{A}'$  is deductively closed. Consequently,  $\alpha_2$  cannot belong to the  $\mathcal{A}'$  (since  $\langle \mathcal{T}, \mathcal{A}' \rangle$  must be satisfiable, and hence  $\alpha_2$  does not belong to the intersection of all the *CAR*-repairs of  $\mathcal{K}$ . In the same way, from (ii) we derive that  $\alpha_1$  does not belong to the intersection of all the *CAR*-repairs of  $\mathcal{K}$ .

Then, we prove that, when  $\mathcal{K}$  is a *DL-Lite<sub>A</sub>* KB,  $\text{clc}(\mathcal{K})$  can be computed in polynomial time as follows: assume  $\mathcal{A} = \{\alpha_1, \dots, \alpha_n\}$ . Then,  $\text{clc}(\mathcal{K})$  consists of the union of the instance assertions derived from the  $n$  KBs  $\langle \mathcal{T}, \{\alpha_1\} \rangle, \dots, \langle \mathcal{T}, \{\alpha_n\} \rangle$ .

Finally, by definition of *ICAR*-semantics, we have that, for every UCQ  $Q$ ,  $\mathcal{K} \models_{\text{ICAR}} Q$  iff  $\langle \mathcal{T}, \mathcal{A}' \rangle \models Q$  where  $\mathcal{A}'$  is the *ICAR*-repair of  $\mathcal{K}$ . From the fact that the algorithm *Compute-ICAR-Repair*( $\mathcal{K}$ ) runs in polynomial time and from tractability of UCQ entailment in *DL-Lite<sub>A</sub>*, the thesis follows.  $\square$

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<sub>A</sub>* KB and let  $\alpha$  be an ABox assertion. Then,  $\mathcal{K} \models_{\text{CAR}} \alpha$  iff  $\mathcal{K} \models_{\text{ICAR}} \alpha$ .*

*Proof.*  $\mathcal{K} \models_{\text{CAR}} \alpha$  if  $\mathcal{K} \models_{\text{ICAR}} \alpha$  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  $\mathcal{K} \models_{\text{CAR}} \alpha$  iff  $\alpha$  belongs to the intersection of all the *CAR*-repairs of  $\langle \mathcal{T}, \mathcal{A} \rangle$ , i.e., to the *ICAR*-repair of  $\mathcal{K}$ .  $\square$

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<sub>A</sub>* KB and let  $\alpha$  be an ABox assertion. Deciding whether  $\mathcal{K} \models_{\text{CAR}} \alpha$  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.

	<i>AR</i> semantics	<i>CAR</i> semantics	<i>IAR</i> semantics	<i>ICAR</i> semantics
instance checking	coNP-complete	in PTIME	in PTIME	in PTIME
UCQ entailment	coNP-complete	coNP-complete	in PTIME	in PTIME

**Fig. 2.** Data complexity of UCQ entailment over  $DL-Lite_A$  KBs under inconsistency-tolerant semantics.

## 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.

## 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. Cali, 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.