On the Complexity of Dealing with Inconsistency in Description Logic Ontologies

Riccardo Rosati
Dipartimento di Informatica e Sistemistica
Sapienza Università di Roma, Italy

Abstract

We study the problem of dealing with inconsistency in Description Logic (DL) ontologies. We consider inconsistency-tolerant semantics recently proposed in the literature, called *AR-semantics* and *CAR-semantics*, which are based on repairing (i.e., modifying) in a minimal way the extensional knowledge (ABox) while keeping the intensional knowledge (TBox) untouched. We study instance checking and conjunctive query entailment under the above inconsistency-tolerant semantics for a wide spectrum of DLs, ranging from tractable ones ($\mathcal{EL}$) to very expressive ones ($\mathcal{SHIQ}$), showing that reasoning under the above semantics is inherently intractable, even for very simple DLs. To the aim of overcoming such a high computational complexity of reasoning, we study sound approximations of the above semantics. Surprisingly, our computational analysis shows that reasoning under the approximated semantics is intractable even for tractable DLs. Finally, we identify suitable language restrictions of such DLs allowing for tractable reasoning under inconsistency-tolerant semantics.

1 Introduction

In this paper we study the problem of dealing with inconsistency in Description Logic (DL) ontologies. This problem is becoming of increasing importance in practical applications of ontologies. In fact, the size of ontologies used by real applications is scaling up, and ontologies are increasingly merged and integrated into larger ontologies: the probability of introducing inconsistency in these activities is consequently getting higher, and dealing with inconsistency is becoming a practical issue in ontology-based systems (see e.g. [Hogan *et al.*, 2010]).

A traditional approach (analogous to data cleaning in databases) is to manually (or semi-automatically) repair the inconsistency, by suitably modifying the ontology. This might be not be practical or feasible in many contexts, e.g., when dealing with (very) large ontologies, or integrating different ontologies. An alternative approach is to define *inconsistency-tolerant* semantics, which, differently from the classical FOL semantics of DLs, are able to derive meaningful conclusions from inconsistent ontologies, and can be the formal basis for an automated treatment of inconsistency.

The inconsistency-tolerant semantics for DL ontologies that we consider are based on repairing (i.e., modifying) the extensional knowledge (i.e., the ABox $\mathcal{A}$) while keeping the intensional knowledge (i.e., the TBox $\mathcal{T}$) untouched. Specifically, such semantics are based on the notion of *ABox 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 classical 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.

In particular, we consider inconsistency-tolerant semantics recently proposed in [Lembo *et al.*, 2010], called *AR-semantics* and *CAR-semantics*, for which reasoning has been studied in the context of the Description Logics of the *DL-Lite* family. The notion of ABox repair in the AR-semantics is a very simple and natural one: a repair is a maximal subset of the ABox that is consistent with the TBox. The CAR-semantics is a variant of the AR-semantics that is based on a notion of "equivalence under consistency" of ABoxes inconsistent with respect to a given TBox.

We study the main forms of extensional reasoning in DL ontologies (instance checking and conjunctive query entailment) under the above inconsistency-tolerant semantics for a wide spectrum of DLs, ranging from tractable ones ($\mathcal{EL}$) to very expressive ones ($\mathcal{SHIQ}$). Our results confirm the results obtained in [Lembo *et al.*, 2010] for *DL-Lite* and show that reasoning under the above semantics is inherently intractable, even for very simple DLs. So, to the aim of overcoming the high computational complexity of reasoning, we also consider *sound approximations* of the previous semantics. In particular, we focus on semantics (called IAR-semantics and $ICAR$-semantics in [Lembo *et al.*, 2010]) that consider as a repair, in each of the previous semantics, the ABox corresponding to the intersection of all the repairs. Surprisingly, our computational analysis shows that, differently from the case of *DL-Lite*, reasoning under the approximated semantics is in general intractable even for tractable DLs.

Finally, we study tractable cases, i.e., suitable restrictions of such DLs that allow for tractable reasoning under inconsistency-tolerant semantics. In particular, we identify a fragment of the DL $\mathcal{EL}_\bot$ for which reasoning under an inconsistency-tolerant semantics (in particular, the IAR-semantics) is computationally not harder than reasoning un-

DL	concept and role expressions	TBox axioms
$\mathcal{EL}_\bot$	$C ::= A \mid \bot \mid C_1 \sqcap C_2 \mid \exists P.C$ $R ::= P$	$C_1 \sqsubseteq C_2$
$\mathcal{ALC}$	$C ::= A \mid C_1 \sqcap C_2 \mid \neg C \mid \exists P.C$ $R ::= P$	$C_1 \sqsubseteq C_2$
$\mathcal{SHIQ}$	$C ::= A \mid \neg C \mid C_1 \sqcap C_2 \mid (\geq n\, R\, C)$ $R ::= P \mid P^-$	$C_1 \sqsubseteq C_2$ $R_1 \sqsubseteq R_2$ Trans(R)

Figure 1: Abstract syntax of the DLs studied in the paper.

der the standard DL semantics.

2 Preliminaries

The DLs mainly considered in this paper are the following:

- $\mathcal{EL}$ [Baader *et al.*, 2005], a prominent tractable DL. Here, we consider $\mathcal{EL}_\bot$, a slight extension of $\mathcal{EL}$ allowing for the empty concept $\bot$ (and hence for inconsistency in KBs);
- $\mathcal{ALC}$, a very well-known DL which corresponds to multimodal logic K_n [Baader *et al.*, 2003];
- $\mathcal{SHIQ}$ [Glimm *et al.*, 2008], a very expressive DL which constitutes the basis of the OWL family of DLs adopted as standard languages for ontology specification in the Semantic Web.

In every DL, *concept expressions* and *role expressions* can be constructed starting from concept and role names. Such expressions are used to define *axioms*. Typical axioms are concept inclusions, role inclusions, and instance assertions. A *concept inclusion* is an expression of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are concept expressions. Similarly, a *role inclusion* is an expression of the form $R_1 \sqsubseteq R_2$, where R_1 and R_2 are role expressions. An *instance (ABox) assertion* is an expression of the form $A(a)$ or $P(a, b)$, where A is a concept name, P is a role name, and a, b are constant (individual) names. We do not consider complex concept and role expressions in instance assertions. A *DL knowledge base (KB)* is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$, called the *TBox*, is a set of concept and role inclusions, and $\mathcal{A}$, called the *ABox*, is a set of instance assertions.

The syntax of $\mathcal{EL}_\bot$, $\mathcal{ALC}$ and $\mathcal{SHIQ}$ is summarized in Figure 1, in which the symbol A denotes a concept name and the symbol P denotes a role name (in addition to concept and role inclusions, $\mathcal{SHIQ}$ also allows for TBox axioms of the form Trans(R), which state transitivity of the role R).

The semantics of DLs can be given through the well-known translation ρ_{fol} of DL knowledge bases into FOL theories with counting quantifiers (see [Baader *et al.*, 2003]). An interpretation of $\mathcal{K}$ is a classical FOL interpretation for $\rho_{fol}(\mathcal{K})$. A *model* of a DL KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a FOL model of $\rho_{fol}(\mathcal{K})$. We say that $\mathcal{K}$ is *satisfiable* if $\mathcal{K}$ has a model. We say that $\mathcal{K}$ entails a FOL sentence ϕ (and write $\mathcal{K} \models \phi$) if ϕ is satisfied in every model of $\mathcal{K}$.

In the following, we are interested in particular in *UCQ entailment*, i.e., the problem of deciding whether a KB entails a *(Boolean) union of conjunctive queries (UCQ)*, i.e., a FOL sentence of the form $\exists \vec{y_1}.conj_1(\vec{y_1}) \vee \cdots \vee \exists \vec{y_n}.conj_n(\vec{y_n})$ where $\vec{y_1}, \ldots, \vec{y_n}$ are terms (i.e., constants or variables), and each $conj_i(\vec{y_i})$ is a conjunction of atoms of the form $A(z)$ and $P(z, z')$, where A ia a concept name, P is a role name and z, z' are terms. *Instance checking (IC)* is a restricted form of UCQ entailment, corresponding to the case when the UCQ is an ABox assertion (i.e., a ground atom). Notice that all the results we achieve about Boolean UCQ entailment can be easily extended in the standard way to the presence of free variables in queries (see e.g. [Glimm *et al.*, 2008]).

In this paper, we will consider *data complexity* (i.e., the complexity with respect to the size of the ABox) and *combined complexity* (i.e., the complexity with respect to the size of the whole input) of UCQ entailment and instance checking. Besides the usual complexity classes NP, coNP, Π_2^p, EXPTIME, 2-EXPTIME, we will mention the following classes:

- $\Delta_2^p[\mathcal{O}(\log n)]$ (resp., $\Delta_3^p[\mathcal{O}(\log n)]$) is the class of problems that can be solved in polynomial time using $\mathcal{O}(\log n)$ calls to an NP-oracle (resp., a Σ_2^p-oracle) (see e.g. [Gottlob, 1995]);
- DP is the class of problems corresponding to the conjunction of a problem in Σ_2^p and a problem in Π_2^p. An example of a problem complete for this class is 3-CNF-SAT+UNSAT: given two 3-CNF formulas ϕ_1, ϕ_2, decide whether ϕ_1 is satisfiable and ϕ_2 is unsatisfiable. Notice that DP $=$ BH$_2$(NP), i.e., DP is the second level of the Boolean hierarchy over NP;
- BH$_2(\Sigma_2^p)$ is the class of problems corresponding to the conjunction of a problem in Σ_2^p and a problem in Π_2^p. An example of a problem complete for this class is 2-QBF-SAT+UNSAT: given two 2-QBF formulas ϕ_1, ϕ_2, decide whether ϕ_1 is not valid and ϕ_2 is valid.

Finally, the following table recalls known results on the complexity of instance checking and UCQ entailment under standard semantics in the DLs considered in this paper (all problems are complete w.r.t. the class reported).

DL (problem)	data complexity	combined complexity
$\mathcal{EL}, \mathcal{EL}_\bot$ (IC)	PTIME	PTIME
$\mathcal{EL}, \mathcal{EL}_\bot$ (UCQ)	PTIME	NP
$\mathcal{ALC}$ (IC)	coNP	EXPTIME
$\mathcal{ALC}$ (UCQ)	coNP	EXPTIME
$\mathcal{SHIQ}$ (IC)	coNP	EXPTIME
$\mathcal{SHIQ}$ (UCQ)	coNP	2-EXPTIME

3 Inconsistency-tolerant semantics

In this section we recall the inconsistency-tolerant semantics for DL ontologies defined in [Lembo *et al.*, 2010].[1] 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.

AR-semantics The first notion of repair that we consider, called *AR-repair*, is a very natural one: a repair is a maximal subset of the ABox that is consistent with the TBox. Thus, an AR-repair is obtained by throwing away from $\mathcal{A}$ a minimal set of assertions to make it consistent with $\mathcal{T}$.

Definition 1 Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB. An *AR-repair* of $\mathcal{K}$ is a set $\mathcal{A}'$ of membership assertions such that: (i)$\mathcal{A}' \subseteq \mathcal{A}$;

[1]Due to space limitations, we refer the reader to [Lembo *et al.*, 2010] for introductory examples illustrating these semantics.

(ii) $\langle \mathcal{T}, \mathcal{A}' \rangle$ is satisfiable; (iii) there does not exist $\mathcal{A}''$ such that $\mathcal{A}' \subset \mathcal{A}'' \subseteq \mathcal{A}$ and $\langle \mathcal{T}, \mathcal{A}'' \rangle$ is satisfiable. The set of AR-repairs for $\mathcal{K}$ is denoted by *AR-Rep*$(\mathcal{K})$. Moreover, we say that a first-order sentence ϕ is AR*-entailed* by $\mathcal{K}$, written $\mathcal{K} \models_{AR} \phi$, if $\langle \mathcal{T}, \mathcal{A}' \rangle \models \phi$ for every $\mathcal{A}' \in$ *AR-Rep*$(\mathcal{K})$.

CAR-semantics We start by formally introducing a notion of "equivalence under consistency" for inconsistent KBs.

Given a KB $\mathcal{K}$, let $\mathcal{S}_K$ denote the signature of $\mathcal{K}$, i.e., the set of concept, role, and individual names occurring in $\mathcal{K}$. Given a signature $\mathcal{S}$, we denote with $HB(\mathcal{S})$ the *Herbrand Base of* $\mathcal{S}$, i.e. the set of ABox assertions (ground atoms) that can be built over the signature $\mathcal{S}$. Then, given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, we define the *consistent logical consequences of* $\mathcal{K}$ as the set $clc(\mathcal{K}) = \{\alpha \mid \alpha \in HB(\mathcal{S}_K)$ and there exists $\mathcal{A}' \subseteq \mathcal{A}$ such that $\langle \mathcal{T}, \mathcal{A}' \rangle$ is satisfiable and $\langle \mathcal{T}, \mathcal{A}' \rangle \models \alpha\}$.

Finally, we say that two KBs $\langle \mathcal{T}, \mathcal{A} \rangle$ and $\langle \mathcal{T}, \mathcal{A}' \rangle$ are *consistently equivalent (C-equivalent)* if $clc(\langle \mathcal{T}, \mathcal{A} \rangle) = clc(\langle \mathcal{T}, \mathcal{A}' \rangle)$.

We argue that the notion of C-equivalence is very reasonable in settings in which the ABox (or at least a part of it) has been "closed" with respect to the TBox, e.g., when (some or all) the ABox assertions that are entailed by the ABox and the TBox have been added to the original ABox. This may happen, for example, when the ABox is obtained by integrating different (and locally consistent) sources, since some of these sources might have been locally closed with respect to some TBox axioms.

In settings where C-equivalence makes sense, the AR-semantics is not suited to handle inconsistency. In fact, we would expect two C-equivalent ontologies to produce the same logical consequences under inconsistency-tolerant semantics. Unfortunately, the AR-semantics does not have this property. A simple example is the following: let $\mathcal{T} = \{\mathit{student} \sqsubseteq \mathit{young},\ \mathit{student} \sqcap \mathit{worker} \sqsubseteq \bot\}$ and let $\mathcal{A} = \{\mathit{student}(mary), \mathit{worker}(mary)\}$, $\mathcal{A}' = \{\mathit{student}(mary), \mathit{worker}(mary), \mathit{young}(mary)\}$. It is immediate to verify that $clc(\mathcal{K}) = clc(\mathcal{K}') = \mathcal{A}'$, thus $\mathcal{K}$ and $\mathcal{K}'$ are C-equivalent, however $\mathcal{K}' \models_{AR} young(mary)$ while $\mathcal{K} \not\models_{AR} young(mary)$.

To overcome the above problem, the CAR-semantics has been defined in [Lembo *et al.*, 2010], through a modification of the AR-semantics.[2]

Definition 2 Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB. A CAR*-repair* for $\mathcal{K}$ is a set $\mathcal{A}'$ of membership assertions such that $\mathcal{A}'$ is an AR-repair of $\langle \mathcal{T}, clc(\mathcal{K}) \rangle$. The set of CAR-repairs for $\mathcal{K}$ is denoted by *CAR-Rep*$(\mathcal{T}, \mathcal{A})$. Moreover, we say that a first-order sentence ϕ is CAR*-entailed* by $\mathcal{K}$, written $\mathcal{K} \models_{CAR} \phi$, if $\langle \mathcal{T}, \mathcal{A}' \rangle \models \phi$ for every $\mathcal{A}' \in$ *CAR-Rep*$(\mathcal{K})$.

Going back to the previous example, it is immediate to see that, since $\mathcal{K}$ and $\mathcal{K}'$ are C-equivalent, the set of CAR-repairs (and hence the set of CAR-models) of $\mathcal{K}$ and $\mathcal{K}'$ coincide.

As the above example shows, there are sentences entailed by a KB under CAR-semantics that are not entailed under AR-semantics. Conversely, it is shown in [Lembo *et al.*, 2010] that the AR-semantics is a sound approximation of the CAR-semantics, i.e., for every KB $\mathcal{K}$ and every FOL sentence ϕ, $\mathcal{K} \models_{AR} \phi$ implies $\mathcal{K} \models_{CAR} \phi$.

IAR-semantics and ICAR-semantics We then recall the sound approximations of the AR-semantics and the CAR-semantics, called IAR-semantics and $ICAR$-semantics, respectively [Lembo *et al.*, 2010].

Definition 3 Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB. The IAR*-repair* (respectively, $ICAR$*-repair*) for $\mathcal{K}$, denoted by *IAR-Rep*$(\mathcal{K})$ (respectively, *ICAR-Rep*$(\mathcal{K})$) is defined as *IAR-Rep*$(\mathcal{K}) = \bigcap_{\mathcal{A}' \in \text{AR-Rep}(\mathcal{K})} \mathcal{A}'$ (respectively, *ICAR-Rep*$(\mathcal{K}) = \bigcap_{\mathcal{A}' \in \text{CAR-Rep}(\mathcal{K})} \mathcal{A}'$). Moreover, we say that a first-order sentence ϕ is IAR*-entailed* (resp., $ICAR$-entailed) by $\mathcal{K}$, and we write $\mathcal{K} \models_{IAR} \phi$ (resp., $\mathcal{K} \models_{ICAR} \phi$), if $\langle \mathcal{T}, \text{IAR-Rep}(\mathcal{K}) \rangle \models \phi$ (resp., $\langle \mathcal{T}, \text{ICAR-Rep}(\mathcal{K}) \rangle \models \phi$).

It is immediate to see [Lembo *et al.*, 2010] 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, while the converse in general does not hold.

4 Lower bounds

We start by considering data complexity. First, we provide lower bounds for instance checking in $\mathcal{EL}_\bot$.[3]

Theorem 1 *Let $\mathcal{K}$ be an $\mathcal{EL}_\bot$ KB and let α be an ABox assertion. Then: (i) deciding whether $\mathcal{K} \models_{AR} \alpha$ is coNP-hard w.r.t. data complexity; (ii) deciding whether $\mathcal{K} \models_{IAR} \alpha$ is coNP-hard w.r.t. data complexity; (iii) deciding whether $\mathcal{K} \models_{CAR} \alpha$ is DP-hard w.r.t. data complexity.*

Then, we consider UCQ entailment, still in $\mathcal{EL}_\bot$.

Theorem 2 *Let $\mathcal{K}$ be an $\mathcal{EL}_\bot$ KB and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{CAR} Q$ is $\Delta_2^p[\mathcal{O}(\log n)]$-hard w.r.t. data complexity; (ii) deciding whether $\mathcal{K} \models_{ICAR} Q$ is $\Delta_2^p[\mathcal{O}(\log n)]$-hard w.r.t. data complexity.*

We then provide a lower bound for instance checking in $\mathcal{ALC}$.

Theorem 3 *Let $\mathcal{K}$ be an $\mathcal{ALC}$ KB and let α be an ABox assertion. Then: (i) deciding whether $\mathcal{K} \models_{AR} \alpha$ is Π_2^p-hard w.r.t. data complexity; (ii) deciding whether $\mathcal{K} \models_{IAR} \alpha$ is Π_2^p-hard w.r.t. data complexity; (iii) deciding whether $\mathcal{K} \models_{CAR} \alpha$ is $BH_2(\Sigma_2^p)$-hard w.r.t. data complexity.*

We now turn our attention to UCQ entailment in $\mathcal{ALC}$.

Theorem 4 *Let $\mathcal{K}$ be an $\mathcal{ALC}$ KB and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{CAR} Q$ is $\Delta_3^p[\mathcal{O}(\log n)]$-hard w.r.t. data complexity; (ii) deciding whether $\mathcal{K} \models_{ICAR} Q$ is $\Delta_3^p[\mathcal{O}(\log n)]$-hard w.r.t. data complexity.*

Finally, we consider combined complexity and provide lower bounds for UCQ entailment in $\mathcal{EL}_\bot$.

Theorem 5 *Let $\mathcal{K}$ be an $\mathcal{EL}_\bot$ KB and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{IAR} Q$ is $\Delta_2^p[\mathcal{O}(\log n)]$-hard w.r.t. combined complexity; (ii) deciding whether $\mathcal{K} \models_{AR} Q$ is Π_2^p-hard w.r.t. combined complexity; (iii) deciding whether $\mathcal{K} \models_{CAR} Q$ is Π_2^p-hard w.r.t. combined complexity.*

[2]The definition provided here of the CAR-semantics is a slight simplification of the one appearing in [Lembo *et al.*, 2010]: this modification, however, does not affect any of the computational results presented in [Lembo *et al.*, 2010].

[3]Due to lack of space, in the present version of the paper we omit the proofs of lower bounds.

5 Upper bounds

We start by providing upper bounds for the combined complexity of instance checking in $\mathcal{EL}_\perp$.

Theorem 6 *Let $\mathcal{K}$ be an $\mathcal{EL}_\perp$ KB and let α be an ABox assertion. Then: (i) deciding whether $\mathcal{K} \models_{AR} \alpha$ is in coNP w.r.t. combined complexity; (ii) deciding whether $\mathcal{K} \models_{IAR} \alpha$ is in coNP w.r.t. combined complexity; (iii) deciding whether $\mathcal{K} \models_{CAR} \alpha$ is in DP w.r.t. combined complexity.*

Proof. For case (i), we define the following Algorithm AR1:

ALGORITHM AR1 (decides $\langle \mathcal{T}, \mathcal{A} \rangle \models_{AR} Q$ with Q UCQ)
if there exists $\mathcal{A}' \subseteq \mathcal{A}$ such that
1) $\langle \mathcal{T}, \mathcal{A}' \rangle$ satisfiable **and**
2) $\mathcal{A}'$ is maximal $\mathcal{T}$-consistent subset of $\mathcal{A}$ **and**
3) $\langle \mathcal{T}, \mathcal{A}' \rangle \not\models Q$
then return false **else return** true

It is easy to verify that the above algorithm is correct for every DL, i.e., it returns true iff $\langle \mathcal{T}, \mathcal{A} \rangle \models_{AR} Q$, and that, in the case of $\mathcal{EL}_\perp$ and when Q is an ABox assertion, the algorithm runs in coNP.

For case (ii), we define the following Algorithm IAR1:

ALGORITHM IAR1 (decides $\langle \mathcal{T}, \mathcal{A} \rangle \models_{IAR} Q$ with Q UCQ)
if there exist $\mathcal{A}' \subseteq \mathcal{A}$ (let $\mathcal{A} - \mathcal{A}' = \{\alpha_1, \ldots, \alpha_n\}$),
$\mathcal{A}_1 \subseteq \mathcal{A}, \ldots, \mathcal{A}_n \subseteq \mathcal{A}$ such that
1) $\langle \mathcal{T}, \mathcal{A}' \rangle \not\models Q$ **and**
2) for each i such that $1 \leq i \leq n$,
$\alpha_i \in \mathcal{A}_i$ and $\mathcal{A}_i$ is a minimal $\mathcal{T}$-inconsistent subset of $\mathcal{A}$
then return false **else return** true

It is easy to verify that the above algorithm is correct for every DL, i.e., it returns true iff $\langle \mathcal{T}, \mathcal{A} \rangle \models_{IAR} Q$, and that, in the case of $\mathcal{EL}_\perp$ and when Q is an ABox assertion, the algorithm runs in coNP.

For case (iii), we define the following algorithm ICAR1:

ALGORITHM ICAR1 (decides $\langle \mathcal{T}, \mathcal{A} \rangle \models_{ICAR} \alpha$)
if (a) for each $\mathcal{A}' \subseteq \mathcal{A}$, either $\langle \mathcal{T}, \mathcal{A}' \rangle$ unsatisfiable or $\langle \mathcal{T}, \mathcal{A}' \rangle \not\models \alpha$
or (b) there exist $\mathcal{A}'' = \{\alpha_1, \ldots, \alpha_n\} \subseteq HB(\mathcal{T}, \mathcal{A})$,
$A_1 \subseteq \mathcal{A}, \ldots, \mathcal{A}_n \subseteq \mathcal{A}$ such that
1) $\alpha \in \mathcal{A}''$ **and** 2) $\langle \mathcal{T}, \mathcal{A}'' \rangle$ unsatisfiable **and**
3) for every i s.t. $1 \leq i \leq n$
3.1) $\langle \mathcal{T}, \mathcal{A}'' - \{\alpha_i\} \rangle$ satisfiable **and**
3.2) $\langle \mathcal{T}, \mathcal{A}_i \rangle$ satisfiable **and** 3.3) $\langle \mathcal{T}, \mathcal{A}_i \rangle \models \alpha_i$
then return false **else return** true

It is easy to verify that the above algorithm is correct for every DL, i.e., it returns true iff $\langle \mathcal{T}, \mathcal{A} \rangle \models_{ICAR} \alpha$, and that, in the case of $\mathcal{EL}_\perp$, the algorithm runs in DP. □

We now turn our attention to both data and combined complexity of UCQ entailment in $\mathcal{EL}_\perp$.

Theorem 7 *Let $\mathcal{K}$ be an $\mathcal{EL}_\perp$ KB and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{AR} Q$ is in coNP w.r.t. data complexity; (ii) deciding whether $\mathcal{K} \models_{IAR} Q$ is in coNP w.r.t. data complexity; (iii) deciding whether $\mathcal{K} \models_{CAR} Q$ is in $\Delta_2^p[\mathcal{O}(\log n)]$ w.r.t. data complexity.*

Proof. Case (i) follows from the correctness of Algorithm AR1 defined in the proof of Theorem 6. Case (ii) follows from the correctness of Algorithm IAR1 defined in the proof of Theorem 6. Finally, case (iii) follows from the correctness (for every DL) of the following Algorithm CAR1:

ALGORITHM CAR1 (decides $\langle \mathcal{T}, \mathcal{A} \rangle \models_{CAR} Q$ with Q UCQ)
$\mathcal{A}' = \emptyset$;
for each $\alpha \in HB(\mathcal{T}, \mathcal{A})$ **do**
if there exists $\mathcal{A}'' \subseteq \mathcal{A}$
such that $\langle \mathcal{T}, \mathcal{A}'' \rangle$ satisfiable **and** $\langle \mathcal{T}, \mathcal{A}'' \rangle \models \alpha$
then $\mathcal{A}' = \mathcal{A}' \cup \{\alpha\}$;
if $\langle \mathcal{T}, \mathcal{A}' \rangle \models_{AR} Q$ **then return** true **else return** false

When the input KB is in $\mathcal{EL}_\perp$, the above algorithm can be reduced to an NP tree [Gottlob, 1995], which proves the $\Delta_2^p[\mathcal{O}(\log n)]$ upper bound. □

Theorem 8 *Let $\mathcal{K}$ be an $\mathcal{EL}_\perp$ KB and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{AR} Q$ is in Π_2^p w.r.t. combined complexity; (ii) deciding whether $\mathcal{K} \models_{CAR} Q$ is in Π_2^p w.r.t. combined complexity; (iii) deciding whether $\mathcal{K} \models_{IAR} Q$ is in $\Delta_2^p[\mathcal{O}(\log n)]$ w.r.t. combined complexity; (iv) deciding whether $\mathcal{K} \models_{ICAR} Q$ is in $\Delta_2^p[\mathcal{O}(\log n)]$ w.r.t. combined complexity.*

Proof. Case (i) follows from the correctness of Algorithm AR1 defined in the proof of Theorem 6. Case (ii) follows from the correctness (for every DL) of the following Algorithm CAR2:

ALGORITHM CAR2 (decides $\langle \mathcal{T}, \mathcal{A} \rangle \models_{CAR} Q$ with Q UCQ)
if there exists $\mathcal{A}' \subset HB(\mathcal{T}, \mathcal{A})$ such that
1) for each $\alpha \in \mathcal{A}'$
$\alpha \in clc(\mathcal{T}, \mathcal{A})$ **and**
2) for each $\alpha \in HB(\mathcal{T}, \mathcal{A}) - \mathcal{A}'$
$\alpha \notin clc(\mathcal{T}, \mathcal{A})$ **or** $\langle \mathcal{T}, \mathcal{A}' \cup \{\alpha\} \rangle$ unsatisfiable **and**
3) $\langle \mathcal{T}, \mathcal{A}' \rangle \not\models Q$
then return false **else return** true

and from the fact that deciding $\alpha \in clc(\mathcal{T}, \mathcal{A})$ (with α ABox assertion) can be decided in NP. Case (iii) follows from the correcness (for every DL) of the following Algorithm IAR2:

ALGORITHM IAR2 (decides $\langle \mathcal{T}, \mathcal{A} \rangle \models_{IAR} Q$ with Q UCQ)
for each $\alpha \in \mathcal{A}$ **do**
if α does not belong to any minimal $\mathcal{T}$-inconsistent subset of $\mathcal{A}$
then $\mathcal{A}' = \mathcal{A}' \cup \{\alpha\}$;
if $\langle \mathcal{T}, \mathcal{A}' \rangle \models Q$ **then return** true **else return** false

Finally, Case (iv) follows from the correctness (for every DL) of the following Algorithm ICAR2:

ALGORITHM ICAR2 (decides $\langle \mathcal{T}, \mathcal{A} \rangle \models_{ICAR} Q$ with Q UCQ)
$\mathcal{A}' = \emptyset$;
for each $\alpha \in HB(\mathcal{T}, \mathcal{A})$ **do**
if there exists $\mathcal{A}'' \subseteq \mathcal{A}$
such that $\langle \mathcal{T}, \mathcal{A}'' \rangle$ satisfiable and $\langle \mathcal{T}, \mathcal{A}'' \rangle \models \alpha$
then $\mathcal{A}' = \mathcal{A}' \cup \{\alpha\}$;
if $\langle \mathcal{T}, \mathcal{A}' \rangle \models_{IAR} Q$ **then return** true **else return** false

and from the fact that, when the input KB is in $\mathcal{EL}_\perp$, the above algorithm can be reduced to an NP tree [Gottlob, 1995], which proves the $\Delta_2^p[\mathcal{O}(\log n)]$ upper bound. □

We then consider the combined complexity of UCQ entailment in $\mathcal{ALC}$.

Theorem 9 *Let $\mathcal{K}$ be an $\mathcal{ALC}$ KB and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{AR} Q$ is in* EXPTIME *w.r.t. combined complexity; (ii) deciding whether $\mathcal{K} \models_{CAR} Q$ is in* EXPTIME *w.r.t. combined complexity; (iii) deciding whether $\mathcal{K} \models_{IAR} Q$ is in* EXPTIME *w.r.t. combined complexity; (iv) deciding whether $\mathcal{K} \models_{ICAR} Q$ is in* EXPTIME *w.r.t. combined complexity.*

Proof. Case (i) follows from the correctness of Algorithm AR1 defined in the proof of Theorem 6. Case (ii) follows from the correctness of Algorithm CAR1 defined in the proof of Theorem 7. Case (iii) follows from the correctness of Algorithm IAR2 defined in the proof of Theorem 8. Finally, case (iv) follows from the correctness of Algorithm ICAR2 defined in the proof of Theorem 8. □

Then, we provide upper bounds for the data complexity of both instance checking and UCQ entailment in $\mathcal{SHIQ}$.

Theorem 10 *Let $\mathcal{K}$ be an $\mathcal{SHIQ}$ KB and let α be an ABox assertion. Then, deciding whether $\mathcal{K} \models_{CAR} \alpha$ is in $BH_2(\Sigma_2^p)$ w.r.t. data complexity.*

Proof. The proof follows from the correctness of Algorithm ICAR1 defined in the proof of Theorem 6. □

Theorem 11 *Let $\mathcal{K}$ be a $\mathcal{SHIQ}$ KB and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{AR} Q$ is in Π_2^p w.r.t. data complexity; (ii) deciding whether $\mathcal{K} \models_{IAR} Q$ is in Π_2^p w.r.t. data complexity; (iii) deciding whether $\mathcal{K} \models_{CAR} Q$ is in $\Delta_3^p[\mathcal{O}(\log n)]$ w.r.t. data complexity; (iv) deciding whether $\mathcal{K} \models_{ICAR} Q$ is in $\Delta_3^p[\mathcal{O}(\log n)]$ w.r.t. data complexity.*

Proof. Case (i) follows from the correctness of Algorithm AR1 defined in the proof of Theorem 6. Case (ii) follows from the correctness of Algorithm IAR2 defined in the proof of Theorem 8. Case (iii) follows from the correctness of Algorithm CAR1 defined in the proof of Theorem 7. It can be proved that, when the input KB is in $\mathcal{SHIQ}$, Algorithm CAR1 can be reduced to a Σ_2^p tree, which proves the $\Delta_3^p[\mathcal{O}(\log n)]$ upper bound. Analogously, case (iv) follows from the correctness of Algorithm ICAR2 defined in Theorem 8: when the input KB is in $\mathcal{SHIQ}$, Algorithm ICAR2 can be reduced to a Σ_2^p tree, which proves the $\Delta_3^p[\mathcal{O}(\log n)]$ upper bound. □

Finally, we provide upper bounds for the combined complexity of instance checking and UCQ entailment in $\mathcal{SHIQ}$.

Theorem 12 *Let $\mathcal{K}$ be a $\mathcal{SHIQ}$ KB and let α be an ABox assertion. Then: (i) deciding whether $\mathcal{K} \models_{AR} \alpha$ is in* EXPTIME *w.r.t. combined complexity; (ii) deciding whether $\mathcal{K} \models_{CAR} \alpha$ is in* EXPTIME *w.r.t. combined complexity; (iii) deciding whether $\mathcal{K} \models_{IAR} \alpha$ is in* EXPTIME *w.r.t. combined complexity; (iv) deciding whether $\mathcal{K} \models_{ICAR} \alpha$ is in* EXPTIME *w.r.t. combined complexity.*

Proof. Case (i) follows from the correctness of Algorithm AR1 defined in the proof of Theorem 6. Case (ii) follows from the correctness of Algorithm CAR1 defined in the proof of Theorem 7. Case (iii) follows from the correctness of Algorithm IAR2 defined in the proof of Theorem 8. Finally, case (iv) follows from the correctness of Algorithm ICAR2 defined in the proof of Theorem 8. □

Theorem 13 *Let $\mathcal{K}$ be a $\mathcal{SHIQ}$ KB and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{AR} Q$ is in 2-*EXPTIME *w.r.t. combined complexity; (ii) deciding whether $\mathcal{K} \models_{CAR} Q$ is in 2-*EXPTIME *w.r.t. combined complexity; (iii) deciding whether $\mathcal{K} \models_{IAR} Q$ is in 2-*EXPTIME *w.r.t. combined complexity; (iv) deciding whether $\mathcal{K} \models_{ICAR} Q$ is in 2-*EXPTIME *w.r.t. combined complexity.*

Proof. Case (i) follows from Algorithm AR1 defined in the proof of Theorem 6. Case (ii) follows from Algorithm CAR1 defined in the proof of Theorem 7. Case (iii) follows from Algorithm IAR2 defined in the proof of Theorem 8. Finally, case (iv) follows from Algorithm ICAR2 defined in the proof of Theorem 8. □

The theorems presented in the last two sections provide a complete picture of the complexity of reasoning in the DLs considered in this paper under the four inconsistency-tolerant semantics. A summary of the complexity results obtained is reported in Figure 2. All the results displayed are completeness results: all the lower bounds are implied by theorems 1–5 and by the lower bounds for reasoning under standard DL semantics reported in the preliminaries, while all the upper bounds are implied by theorems 6–13.

6 Tractable cases

We now consider a language restriction of $\mathcal{EL}_\bot$ whose aim is to allow for tractable reasoning under inconsistency-tolerant semantics, in particular, under the IAR-semantics. The condition is on the form of empty concept assertions, i.e., concept inclusions whose right-hand side is the empty concept $\bot$. Formally, an *empty concept assertion in $\mathcal{EL}_\bot$* is an inclusion of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq \bot$.

Given an $\mathcal{EL}_\bot$ TBox $\mathcal{T}$, we say that a concept name N is *relevant for empty concepts in $\mathcal{T}$* if there exists an empty concept assertion δ such that N occurs in δ and $\mathcal{T} \models \delta$ and $\mathcal{T} \not\models \delta[\top/N]$, where $\delta[\top/N]$ is the empty concept assertion obtained from δ by replacing every occurrence of N with $\top$. Intuitively, concept names relevant for empty concepts in a TBox $\mathcal{T}$ are those involved in the empty concept assertions entailed by $\mathcal{T}$. The above definition filters out "redundant" (i.e., non-minimal) empty concept assertions.

Then, we introduce the notion of (non)-recursive concept name in $\mathcal{T}$. A concept name N is *recursive in $\mathcal{T}$* if $\mathcal{T}$ entails an inclusion of the form $C \sqsubseteq N$, where C contains at least an occurrence of N nested into an existentially quantified concept expression, and there exists no concept C' such that $\mathcal{T}$ entails $C' \sqsubseteq N$ and C' is more specific than C in $\mathcal{T}$ (i.e., $\mathcal{T}$ entails $C' \sqsubseteq C$ but not vice versa). Otherwise, we say that N is non-recursive in $\mathcal{T}$. We call *$\mathcal{EL}_{\bot nr}$ ($\mathcal{EL}_\bot$ with non-recursive empty concepts) KB* every $\mathcal{EL}_\bot$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ such that every concept name that is relevant for empty concepts in $\mathcal{T}$ is non-recursive in $\mathcal{T}$. It is easy to see that deciding whether an $\mathcal{EL}_\bot$ KB is an $\mathcal{EL}_{\bot nr}$ KB can be done in time polynomial w.r.t. the size of the TBox.

The language restriction imposed in $\mathcal{EL}_{\bot nr}$ KBs allows us to prove the following property, which is crucial for lowering the complexity of reasoning under the IAR-semantics.

Lemma 1 *Let $\langle \mathcal{T}, \mathcal{A} \rangle$ be an $\mathcal{EL}_{\bot nr}$ KB. The maximum cardinality of a minimal $\mathcal{T}$-inconsistent subset of $\mathcal{A}$ is bounded by the number of concept expressions occurring in $\mathcal{T}$.*

We are now ready to state the complexity of reasoning in $\mathcal{EL}_{\bot nr}$ under IAR-semantics.

Theorem 14 *Let $\mathcal{K}$ be an $\mathcal{EL}_{\bot nr}$ KB, let α be an ABox assertion, and let Q be a UCQ. Then: (i) deciding whether $\mathcal{K} \models_{IAR} \alpha$ is* PTIME*-complete w.r.t. data complexity; (ii)*

	data complexity				combined complexity			
semantics	AR	CAR	IAR	$ICAR$	AR	CAR	IAR	$ICAR$
$\mathcal{EL}_\perp$ (IC)	coNP	DP	coNP	DP	coNP	DP	coNP	DP
$\mathcal{EL}_\perp$ (UCQ)	coNP	$\Delta_2^p[\mathcal{O}(\log n)]$	coNP	$\Delta_2^p[\mathcal{O}(\log n)]$	Π_2^p	Π_2^p	$\Delta_2^p[\mathcal{O}(\log n)]$	$\Delta_2^p[\mathcal{O}(\log n)]$
$\mathcal{ALC}$ (IC)	Π_2^p	$\mathrm{BH}_2(\Sigma_2^p)$	Π_2^p	$\mathrm{BH}_2(\Sigma_2^p)$	EXPTIME	EXPTIME	EXPTIME	EXPTIME
$\mathcal{ALC}$ (UCQ)	Π_2^p	$\Delta_3^p[\mathcal{O}(\log n)]$	Π_2^p	$\Delta_3^p[\mathcal{O}(\log n)]$	EXPTIME	EXPTIME	EXPTIME	EXPTIME
$\mathcal{SHIQ}$ (IC)	Π_2^p	$\mathrm{BH}_2(\Sigma_2^p)$	Π_2^p	$\mathrm{BH}_2(\Sigma_2^p)$	EXPTIME	EXPTIME	EXPTIME	EXPTIME
$\mathcal{SHIQ}$ (UCQ)	Π_2^p	$\Delta_3^p[\mathcal{O}(\log n)]$	Π_2^p	$\Delta_3^p[\mathcal{O}(\log n)]$	2-EXPTIME	2-EXPTIME	2-EXPTIME	2-EXPTIME

Figure 2: Complexity of UCQ entailment over DL KBs under inconsistency-tolerant semantics.

deciding whether $\mathcal{K} \models_{IAR} \alpha$ *is* PTIME-*complete w.r.t. combined complexity; (iii) deciding whether* $\mathcal{K} \models_{IAR} Q$ *is* PTIME-*complete w.r.t. data complexity; (iv) deciding whether* $\mathcal{K} \models_{IAR} Q$ *is NP-complete w.r.t. combined complexity.*

Proof. For all the four cases, the upper bounds are an immediate consequence of Lemma 1 and of the correctness of Algorithm IAR2 in the proof of Theorem 8 (observe that Lemma 1 implies that there are only polynomially many potential minimal $\mathcal{T}$-inconsistent subsets w.r.t. the size of the ABox), while the lower bounds are immediately implied by the complexity of reasoning under standard DL semantics in $\mathcal{EL}$ (notice that $\mathcal{EL}$ is a sublogic of $\mathcal{EL}_{\perp nr}$). □

The above theorem shows the nice computational behaviour of reasoning under IAR semantics in $\mathcal{EL}_{\perp nr}$, which is not harder than reasoning under standard DL semantics. This case seems very important not only from the theoretical viewpoint, but also for practical applications, since it identifies a DL allowing for tractable automated treatment of inconsistency. Conversely, it can be shown that reasoning under the $ICAR$ semantics is still intractable in $\mathcal{EL}_{\perp nr}$. Informally, the reason is that the restriction imposed by $\mathcal{EL}_{\perp nr}$ only affects the cardinality of minimal $\mathcal{T}$-inconsistent subsets of $\mathcal{A}$, while it leaves unbounded the cardinality of the minimal subsets of $\mathcal{A}$ entailing an ABox assertion. Therefore, it is still NP-hard to decide whether an ABox assertion belongs to $clc(\mathcal{T}, \mathcal{A})$.

7 Related work and conclusions

Inconsistency-tolerance has been studied in various forms in several areas of Artificial Intelligence and databases. While several recent papers have dealt with different forms of inconsistency in DL ontologies (focusing especially on the TBox, see e.g. [Qi and Du, 2009]), the approach considered in this paper (based on instance-level repair only) is novel for DLs, and is actually inspired by the work on *consistent query answering (CQA)* in databases (see [Chomicki, 2007] for a survey). Moreover, the form of inconsistency-tolerance considered in this paper corresponds to a form of belief revision [Eiter and Gottlob, 1992]: more precisely, with respect to the knowledge base revision framework, the ABox corresponds to the initial knowledge, while the TBox represents the new information. Based on such a correspondence, the semantic studies on belief revision are indeed very relevant for the present setting (e.g., the IAR-semantics and $ICAR$-semantics are deeply connected to the well-known WIDTIO semantics of belief revision [Eiter and Gottlob, 1992]): however, the kind of theories and formulas considered in the DL setting have a very special form, which has never been considered in the research in belief revision and update. Therefore, (to the best of our knowledge) there are no available results about reasoning in the present setting.

The present work can be continued along several lines. For instance, it would be very interesting to see whether the present approach can be extended to handle more general forms of ABoxes (e.g., ABoxes with variables and/or non-atomic concept expressions). Also, it would be very important for practical purposes to identify further DL sublanguages allowing for tractable reasoning under the IAR and the ICAR semantics.

Acknowledgments This research has been partially supported by the EU funded Project ACSI (Artifact-Centric Service Interoperation).

References

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI 2005*.

[Chomicki, 2007] Jan Chomicki. Consistent query answering: Five easy pieces. In *Proc. of ICDT 2007*.

[Dantsin *et al.*, 2001] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3), 2001.

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

[Glimm *et al.*, 2008] Birte Glimm, Ian Horrocks, Carsten Lutz, and Uli Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *J. of Artificial Intelligence Research*, 31:151–198, 2008.

[Gottlob, 1995] Georg Gottlob. NP trees and Carnap's modal logic. *J. of the ACM*, 42(2):421–457, 1995.

[Hogan *et al.*, 2010] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. Weaving the pedantic web. In *Proc. of 3rd Int. Workshop on Linked Data on the Web (LDOW2010)*, 2010.

[Lembo *et al.*, 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proc. of RR 2010*, 2010.

[Qi and Du, 2009] Guilin Qi and Jianfeng Du. Model-based revision operators for terminologies in description logics. In *Proc. of IJCAI 2009*, pages 891–897, 2009.