

Consistent Query Answering over Description Logic Ontologies*

Domenico Lembo and Marco Ruzzi

Dipartimento di Informatica e Sistemistica,
Sapienza Università di Roma
lembo, ruzzi@dis.uniroma1.it

Abstract. Description Logics (DLs) have been widely used in the last years as formal language for specifying ontologies. In several contexts, as ontology-based data integration, it may frequently happen that data contradict the intensional knowledge provided by the ontology through which they are accessed, which therefore may result inconsistent. In this paper, we analyze the problem of consistent query answering (CQA) over DL ontologies, i.e., the problem of providing meaningful answers to queries posed over inconsistent ontologies. We provide inconsistency-tolerant semantics for DLs, and study the computational complexity of CQA under such semantics for the case of conjunctive queries posed over ontologies specified in *DL-Lite_F*, a DL specifically tailored to deal with large amounts of data. We show that the above problem is coNP-complete w.r.t. data complexity. We then study the problem of consistent instance checking for *DL-Lite_F* ontologies, i.e., the instance checking problem considered under our inconsistency-tolerant semantics, and we show that such problem is in PTIME w.r.t. data complexity.

1 Introduction

Description Logics (DLs) have been widely used in the last years as formal language for specifying ontologies, for their ability of combining modelling power and decidability of reasoning [10]. Recently, besides expressive DLs, which suffer from inherently worst-case exponential time behavior of reasoning [3], also DLs that allow for tractable reasoning have been proposed for ontology modelling [2, 7]. Such DLs are particularly suited for management of large amounts of data (e.g., from thousands to millions of instances). Then, a challenging use of them is represented by ontology-based data integration, an issue that has recently received a growing attention in the Semantic Web community [15]. Indeed, integrating data in the Semantic Web context mainly means accessing, collecting, and exchanging data distributed over the web through mediated schemas given in terms of ontologies (i.e., DL TBoxes).

Due to the dynamic nature of the setting described above, it may frequently happen that data contradict the intensional knowledge provided by the ontology through which they are accessed, especially in those cases in which the ontology provides a conceptual view of a number of autonomous data sources, heterogeneous and widely distributed.

* The present work is an extended abstract of [12].

In the above situation, ontologies may result inconsistent, and reasoning over them according to classical DLs may become meaningless, since whatever conclusion may be derived from an inconsistent theory. Then, besides handling inconsistency at the terminological/schema level, which has been a subject recently investigated for ontology-based applications [14, 11], the need arises in this context to deal with inconsistency at the instance/data level. In the present paper we study this problem.

The approach commonly adopted to solve data inconsistency is through data cleaning [5]. This approach is procedural, and is based on domain-specific transformation mechanisms applied to the data. One of its problems is incomplete information on how certain conflicts should be resolved. This typically happens in systems which are not tailored for business logic support at the enterprise level, like systems for information integration over the web. In the last years, an alternative declarative approach has been investigated in the area of consistent query answering (CQA) [1, 4, 6]. Such an approach relies on the notion of *repair* for a database instance that may violate integrity constraints (ICs) specified over its schema. Roughly speaking, a repair is a new database instance which satisfies the constraints in the schema and minimally differs from the original one. In general multiple repairs are possible. Then, CQA amounts to compute those answers to a user query that are in the evaluation of the query over each repair. It is well-known [6, 9] that CQA of Conjunctive Queries (CQs) expressed over database schemas with (even simple forms of) ICs is a coNP-complete problem in data complexity, i.e., the complexity measured only with respect to the size of the database instance.

In this paper, we study CQA over DL ontologies. In particular, we provide a new semantic characterization for DLs, based on the notion of repair. We focus on $DL-Lite_{\mathcal{F}}$, a DL of the *DL-Lite* family [7, 8]. The *DL-Lite* family comprises a series of DLs that are specifically tailored to deal with large amounts of data and reasoning over queries. While the expressive power of the DLs in the *DL-Lite* family is carefully controlled to maintain low the complexity of reasoning, such DLs are expressive enough to capture the main notions of both ontologies, and conceptual modelling formalisms used in databases and software engineering (i.e., ER and UML class diagrams). We study CQA for the class of union of conjunctive queries (UCQs), which is the most expressive class of queries for which decidability of query answering has been proved in DLs [13]. Notably, standard query answering of UCQs over $DL-Lite_{\mathcal{F}}$ can be solved by means of evaluation of suitable first-order logic queries over the underlying *DL-Lite* ABox considered as a flat relational database [7, 8]. This allows for using well established Relational Database Management System technology for reasoning over queries in such DLs.

The contributions of the present paper can then be summarized as follows.

- We provide an inconsistency-tolerant semantics for DLs, which relies on the notion of repair of a DL ontology, and allows for meaningful reasoning in the presence of inconsistency (Section 3);
- We study computational complexity of CQA for conjunctive queries expressed over $DL-Lite_{\mathcal{F}}$ ontologies, and show that such a problem is coNP-complete w.r.t. data complexity (Section 4);
- Towards identification of tractable cases of CQA for *DL-Lite*, we study consistent instance checking over $DL-Lite_{\mathcal{F}}$ ontologies, i.e., the instance checking problem

under our inconsistency-tolerant semantics. and show that such a problem is in PTIME w.r.t. data complexity (Section 5).

2 The Description Logic $DL-Lite_{\mathcal{F}}$

In this section we present the syntax and the semantics of $DL-Lite_{\mathcal{F}}$ [7]. $DL-Lite_{\mathcal{F}}$ concepts and roles are defined as follows:

$$B \longrightarrow A \mid \exists R \quad R \longrightarrow P \mid P^- \quad C \longrightarrow B \mid \neg B$$

where A denotes an atomic concept, P an atomic role, and P^- the inverse of the atomic role P . B denotes a *basic concept*, i.e., a concept that can be either an atomic concept or a concept of the form $\exists R$, and R denotes a *basic role*, i.e., a role that is either an atomic role or the inverse of an atomic role. Finally, C denotes a (*general*) *concept*, which can be a basic concept or its negation. Let B_1 and B_2 be basic concepts, we call *positive inclusions (PIs)* assertions of the form $B_1 \sqsubseteq B_2$, whereas we call *negative inclusions (NIs)* assertions of the form $B_1 \sqsubseteq \neg B_2$.

A DL *knowledge base (KB)* \mathcal{K} is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$ which represents the domain of interest in terms of two parts, a *TBox* \mathcal{T} , specifying the intensional knowledge, and an *ABox* \mathcal{A} , specifying extensional knowledge. A $DL-Lite_{\mathcal{F}}$ TBox is formed by: (i) a finite set of *concept inclusion assertions*, i.e., expressions of the form $B \sqsubseteq C$, meaning that all instances of the basic concept B are also instances of the generic concept C , and (ii) a finite set of *functionality assertions* on roles or on their inverses of the form (funct P) or (funct P^-), respectively, meaning that a relation P (resp. P^-) is functional. $DL-Lite_{\mathcal{F}}$ ABoxes are formed by a finite set of *membership assertions* on atomic concepts and atomic roles, of the form $A(a)$ and $P(a, b)$, stating respectively that the object denoted by the constant a is an instance of the atomic concept A and that the pair of objects denoted by the pair of constants (a, b) is an instance of the role P .

The semantics of a DL is given in terms of interpretations, where an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty interpretation domain $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ that assigns to each concept C a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and to each role R a binary relation $R^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$. In particular we have:

$$\begin{aligned} A^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} & (\exists R)^{\mathcal{I}} &= \{o \mid \exists o'. (o, o') \in R^{\mathcal{I}}\} \\ P^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} & (\neg B)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}} \\ (P^-)^{\mathcal{I}} &= \{(o_2, o_1) \mid (o_1, o_2) \in P^{\mathcal{I}}\} \end{aligned}$$

Furthermore, an interpretation \mathcal{I} is a model of a concept inclusion assertion $B \sqsubseteq C$, if $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$, and is \mathcal{I} is a model of an assertion (funct P) if $(o, o_1) \in P^{\mathcal{I}}$ and $(o, o_2) \in P^{\mathcal{I}}$ implies $o_1 = o_2$. Analogously for (funct P^-).

To specify the semantics of membership assertions, we extend the interpretation function to constants, by assigning to each constant a a *distinct* object $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Note that this implies that we enforce the *unique name assumption* on constants [3]. An interpretation \mathcal{I} is a model of a membership assertion $A(a)$, (resp., $P(a, b)$) if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (resp., $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$).

Given an (inclusion, functionality, or membership) assertion α , and an interpretation \mathcal{I} , we denote by $\mathcal{I} \models \alpha$ the fact that \mathcal{I} is a model of α , and also say that α is satisfied by

\mathcal{I} . Given a (finite) set of assertions κ , we denote by $\mathcal{I} \models \kappa$ the fact that \mathcal{I} is a model of every assertion in κ . A *model of a DL-Lite KB* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an interpretation \mathcal{I} such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. With a little abuse of notation, we also write $\mathcal{I} \models \mathcal{K}$. A KB is *satisfiable* if it has at least one model, otherwise it is *unsatisfiable*.

Example 1. Consider the atomic concepts *Cat*, *Dog*, *Pet* and *Person* and the roles *hasOwner* and *feeds*. The following TBox \mathcal{T} is an example of *DL-Lite_F* TBox:

$$\begin{array}{ll} Dog \sqsubseteq Pet & \exists hasOwner^- \sqsubseteq Person \\ Cat \sqsubseteq Pet & Cat \sqsubseteq \neg Dog \\ Pet \sqsubseteq \exists hasOwner & (\text{funct } hasOwner). \end{array}$$

In the TBox above we say that cats and dogs are pets, every pet has an owner, a cat is not a dog, the owner of an animal is a person, and that a pet cannot have more than one owner. Finally, we show a simple *DL-Lite* ABox \mathcal{A} : *Person*(John), *Dog*(Bruto), *hasOwner*(Tom, Leo). \square

A *union of conjunctive queries (UCQ)* q over a *DL-Lite_F* KB \mathcal{K} is an expression of the form

$$q(\mathbf{x}) \leftarrow \bigvee_{i=1, \dots, n} \exists \mathbf{y}_i. conj_i(\mathbf{x}, \mathbf{y}_i), \quad (1)$$

where each $conj_i(\mathbf{x}, \mathbf{y}_i)$ is a conjunction of atoms and equalities, with free variables \mathbf{x} and \mathbf{y}_i . Variables in \mathbf{x} are called *distinguished*, and the size of \mathbf{x} is called the *arity* of q . The right-hand side of the Formula (1) is called the *body* of q . Atoms in each $conj_i$ are of the form $A(z)$ or $P(z_1, z_2)$, where A and P are respectively an atomic concept and an atomic role of \mathcal{K} , and z, z_1, z_2 are either constants in \mathcal{K} or variables. A *Boolean UCQ* is a query with arity 0, written simply as a sentence of the form $\bigvee_{i=1, \dots, n} \exists \mathbf{y}_i. conj_i(\mathbf{y}_i)$. A UCQ with a single conjunction of atoms, i.e., with $n = 1$ in the Formula (1), is called *conjunctive query (CQ)*.

Let q be a Boolean UCQ over a *DL-Lite_F* KB \mathcal{K} . We say that q is *entailed* by \mathcal{K} , and write $\mathcal{K} \models q$, if, for every model \mathcal{M} of \mathcal{K} , $\mathcal{M} \models q$, where \models is the standard evaluation of first-order sentences in an interpretation. The *instance checking problem* corresponds to entailment of a Boolean ground CQ consisting of a single atom, i.e., a membership assertion of the form $A(a)$ or $P(a, b)$. Let q be a non-Boolean UCQ of arity n over \mathcal{K} , and let \mathbf{t} be an n -tuple of constants. We say that \mathbf{t} is a *certain answer* to q in \mathcal{K} if $\mathcal{K} \models q(\mathbf{t})$, where $q(\mathbf{t})$ is the sentence obtained from the body of q by replacing its distinguished variables by constants in \mathbf{t} . We denote by $Ans(q, \mathcal{K})$ the set of certain answers to q in \mathcal{K} .

Example 2. Let us consider the DL KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where the TBox \mathcal{T} and the ABox \mathcal{A} are as defined in Example 1. The following query q is a CQ over \mathcal{K} : $q(x) \leftarrow Person(x)$. It is easy to see that the set of certain answers to q in \mathcal{K} is $Ans(q, \mathcal{K}) = \{\text{John}, \text{Leo}\}$ where John can be directly derived from the membership assertions of \mathcal{A} , whereas Leo is implied by the inclusion assertion $\exists hasOwner^- \sqsubseteq Person$ and by the role membership assertion *hasOwner*(Tom, Leo). \square

3 Inconsistency-tolerant Semantics

Let us now consider the case in which a DL KB \mathcal{K} is unsatisfiable, i.e., \mathcal{K} does not have any model. As already said in the introduction, reasoning over such a \mathcal{K} is meaningless, since whatever consequence can be deduced from \mathcal{K} . In this section, we provide a new semantics for DL KB that is inconsistency-tolerant, i.e., it allows for “meaningful” reasoning over KBs that are unsatisfiable according to the classical first-order based semantics, as that considered in Section 2 for *DL-Lite_F*. In particular, our semantics is tolerant to the inconsistency that arises in a DL knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ in which a satisfiable TBox \mathcal{T} may be contradicted by the extensional assertions in the ABox \mathcal{A} , thus resulting in possibly unsatisfiable KBs. This situation frequently happens in those systems that provide access to data (possibly integrated from autonomous sources) through DL ontologies, as in Semantic Web applications.

Formally, let \mathcal{I} be an interpretation and let \mathcal{A} be an ABox. We denote by $Sat(\mathcal{I}, \mathcal{A})$ the set of membership assertions from \mathcal{A} that are satisfied in \mathcal{I} , i.e., $Sat(\mathcal{I}, \mathcal{A}) = \{\alpha \mid \alpha \in \mathcal{A} \text{ and } \mathcal{I} \models \alpha\}$.

Definition 1. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB and \mathcal{R} an interpretation. We say that \mathcal{R} is a repair of \mathcal{K} if: (i) \mathcal{R} is a model for \mathcal{T} ; (ii) there exists no interpretation \mathcal{R}' such that \mathcal{R}' is a model for \mathcal{T} and $Sat(\mathcal{R}', \mathcal{A}) \supset Sat(\mathcal{R}, \mathcal{A})$.

In the following, we denote by $Rep(\mathcal{K})$ the set of repairs of \mathcal{K} . It is easy to see that when a KB \mathcal{K} is satisfiable, repairs of \mathcal{K} coincide with models of \mathcal{K} . Also, when the TBox of \mathcal{K} is satisfiable, \mathcal{K} has always at least one repair.

Following the line of research of CQA [1, 4, 6], in our semantics, intensional knowledge specified by the TBox of a knowledge base is considered stronger than data, i.e., the extensional knowledge provided by the ABox. Indeed, a repair \mathcal{R} of a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an interpretation that needs to satisfy \mathcal{T} and that at the same time satisfies a maximal set \mathcal{A}_m of the membership assertions in \mathcal{A} , i.e., \mathcal{R} is a model of the knowledge base $\langle \mathcal{T}, \mathcal{A}_m \rangle$.

Let q be Boolean UCQ over a *DL-Lite* KB \mathcal{K} , we say that q is *consistently entailed* by \mathcal{K} , and write $\mathcal{K} \models_{cons} q$ if, for every $\mathcal{R} \in Rep(\mathcal{K})$, $\mathcal{R} \models q$. Then, given a non-Boolean UCQ q of arity n over \mathcal{K} , we say that an n -tuple \mathbf{t} of constants is a *consistent answer* to q in \mathcal{K} if $\mathcal{K} \models_{cons} q(\mathbf{t})$. We denote by $ConsAns(q, \mathcal{K})$ the set of consistent answers to q in \mathcal{K} . Furthermore, the *consistent instance checking* problem corresponds to consistent entailment of a membership assertion.

We finally notice that, when a DL knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a *DL-Lite* KB, \mathcal{K} may result unsatisfiable only if the ABox \mathcal{A} contradicts the intensional knowledge of the TBox \mathcal{T} . Indeed, it is possible to show that a *DL-Lite* TBox admits always at least one model. As a consequence, we have that our inconsistency-tolerant semantics ensures that every *DL-Lite* KB has always at least one repair.

Example 3. Let us consider again the *DL-Lite_F* TBox \mathcal{T} described in Example 1 and the ABox \mathcal{A}' containing the facts $Person(\text{John})$, $hasOwner(\text{Tom}, \text{John})$, and $hasOwner(\text{Tom}, \text{Leo})$. It is easy to see that the knowledge base $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ is unsatisfiable, since the functionality assertion on $hasOwner$ is violated. Then, each repair \mathcal{R} of \mathcal{K}' is such that $\mathcal{R} \models \mathcal{T}$, and either $Sat(\mathcal{R}, \mathcal{A}') = \{hasOwner(\text{Tom}, \text{John})\}$,

or $Sat(\mathcal{R}, \mathcal{A}') = \{hasOwner(\text{Tom}, \text{Leo})\}$. Let us now consider the Boolean CQ $q' = \exists y.hasOwner(\text{Tom}, y)$ over \mathcal{K}' , asking whether Tom is owned by someone. It is easy to see that q is consistently entailed by \mathcal{K}' . However, for the query $q'' = \{x \mid hasOwner(\text{Tom}, x)\}$, we have that $ConsAns(q'', \mathcal{K}') = \emptyset$. In words, we cannot establish who is the owner of Tom, but we can state that Tom has an owner. \square

4 Consistent Query Answering

In this section we consider the problem of CQA for CQs over $DL-Lite_{\mathcal{F}}$ KBs and show that such a problem is coNP-complete w.r.t. data complexity, i.e., the complexity measured w.r.t. the size of the ABox only.

Theorem 1. *Let \mathcal{K} be a $DL-Lite_{\mathcal{F}}$ KB, q a CQ of arity n over \mathcal{K} , and \mathbf{t} an n -tuple of constants. Then, the problem of establishing whether $\mathbf{t} \in ConsAns(q, \mathcal{K})$ is coNP-complete with respect to data complexity.*

Proof (sketch) CoNP hardness can be proved by a reduction from the 3-colorability problem, whereas membership in coNP follows from the results in [9].

The above result tells us that CQA over $DL-Lite_{\mathcal{F}}$ KBs is in general intractable w.r.t. data complexity, differently from the problem of (standard) query answering over $DL-Lite_{\mathcal{F}}$ KBs [7]. Notice that tractability of query answering (and of classical DL reasoning services) is a crucial property for DLs of the $DL-Lite$ family which makes them particularly suited for dealing with big amounts of data. Then, to preserve this nice behavior, we aim at identifying interesting cases in which CQA is tractable. As we will show in the next section, consistent instance checking is in fact tractable over $DL-Lite_{\mathcal{F}}$ KBs.

5 Consistent Instance Checking

We now address consistent instance checking over $DL-Lite_{\mathcal{F}}$ KBs, and show that such a problem is in PTIME w.r.t. data complexity. To this aim, we define a polynomial time algorithm that takes as input a membership assertion α and a $DL-Lite_{\mathcal{F}}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and returns *true* if α is consistently entailed by \mathcal{K} , *false* otherwise.

In the following we only sketch our algorithm and refer the reader to [12] for a more detailed description of it. We assume that input KB \mathcal{K} is closed w.r.t. logical implication of NIs by PIs and NIs specified in the TBox \mathcal{T} , i.e., we assume that all NIs logically implied by \mathcal{T} are explicitly asserted in \mathcal{T} . Furthermore, we assume that the ABox \mathcal{A} does not contain any membership assertion β such that the KB $\langle \mathcal{T}, \{\beta\} \rangle$ is inconsistent, i.e., \mathcal{A} does not contain membership assertions that not belong to any repair of \mathcal{K} . Notice that each $DL-Lite_{\mathcal{F}}$ KB can be transformed in a KB of the above form through simple pre-processing steps [12].

Our technique can be summarized in the two steps described below. In the first step, we only take care of the PIs in \mathcal{K} by means of the algorithm `PerfectRef` presented in [7]. Informally, such an algorithm takes a $DL-Lite_{\mathcal{F}}$ TBox \mathcal{T} and a UCQ q as input,

and reformulates q according to the PIs in \mathcal{T} , used as rewriting rules, iteratively applied from right to left to the atoms occurring in q , thus producing a new UCQ q_r over \mathcal{K} . In words, **PerfectRef** encodes in q_r the intensional knowledge of \mathcal{T} in such a way that the answers to q over \mathcal{K} correspond to the answers to q_r over the *DL-Lite_F* KB \mathcal{K}^- that is obtained from \mathcal{K} by removing all PIs in \mathcal{T} . We execute the algorithm **PerfectRef** with the membership assertion α and the TBox \mathcal{T} as input. Due to the particular form of the input query, we obtain as result a Boolean union of atoms q_r over \mathcal{K} , i.e., q_r is a CQ of the form $\bigvee_{i=1,\dots,n} \exists \mathbf{y}_i \cdot \text{conj}_i(\mathbf{y}_i)$, in which every conj_i consists of a single atom, denoted in the following by atom_i .

In the second step we take into account only NIs and functionalities in \mathcal{T} . To this aim, we use the algorithm **ConsAnswer** which takes as input the Boolean union of atoms q_r and the *DL-Lite_F* KB \mathcal{K}^- , and returns *true* if $\mathcal{K}^- \models_{\text{cons}} q_r$, *false* otherwise. To explain the algorithm **ConsAnswer** more in detail, we need two preliminary definitions. A \mathcal{K} -opponent to a membership assertion α is a membership assertion $\beta \in \mathcal{A}$ that together with α contradicts a functionality or a NI assertion of \mathcal{T} , i.e., the KB $\langle \mathcal{T}, \{\alpha, \beta\} \rangle$ is unsatisfiable. Then, let q be a Boolean union of atoms $\bigvee_{i=1,\dots,n} \exists \mathbf{y}_i \cdot \text{atom}_i(\mathbf{y}_i)$. A membership assertion γ is an *image* of q if there is an $i \in \{1, \dots, n\}$ such that there exists a substitution σ from the variables in $\text{atom}_i(\mathbf{y}_i)$ to constants in γ such that $\sigma(\text{atom}_i(\mathbf{y}_i)) = \gamma$. Roughly speaking, an image of q is a membership assertion γ such that q is entailed by the knowledge base constituted only by the assertion γ . With these notions in place, we can intuitively describe the behavior of **ConsAnswer**. The algorithm verifies whether there exists an image γ of the input query q_r that belongs to \mathcal{A} such that either (a) γ has no \mathcal{K} -opponents or (b) every \mathcal{K} -opponent β to γ is such that β has at least one \mathcal{K} -opponent β' which is not \mathcal{K} -opponent to γ and is in turn \mathcal{K} -opponent to a different image γ' . If the condition (a) succeeds, then the query q_r is consistently entailed by \mathcal{K} since every repair of \mathcal{K} satisfies the same image of q_r . As for condition (b), it ensures that if a repair \mathcal{R} does not satisfy the image γ , since it satisfies the \mathcal{K} -opponent β to γ , then \mathcal{R} satisfies another image γ' , whose satisfaction is guaranteed by the fact that \mathcal{R} does not satisfy β' .

It is possible to prove that the procedure described above is sound and complete w.r.t. finding a solution to the consistent instance checking problem, and that it runs in time polynomial in the size of the ABox \mathcal{A} . Therefore, we can give the following notable result.

Theorem 2. *Let \mathcal{K} be a *DL-Lite_F* KB and α a membership assertion. Then, the problem of establishing whether $\mathcal{K} \models_{\text{cons}} \alpha$ is in PTIME with respect to data complexity.*

6 Conclusions

The work in the present paper can be extended in several directions. We are currently developing a completely intensional technique for consistent instance checking, with the aim of reducing this problem to query evaluation over a database instance representing the ABox of the knowledge base. Such a technique would allow us to maintain reasoning at the intensional level, as it can be already done for standard query answering over *DL-Lite_F* KBs. We are also working in the direction of identifying other tractable

cases of consistent query answering over $DL\text{-}Lite_{\mathcal{F}}$ KBs. In this respect, we point out that results of the present paper immediately imply that consistent query answering of Boolean ground unions of conjunctive queries is tractable. The same analysis is being carried out over other DLs that allow for tractable reasoning [2]. Finally, we are also studying the problem of consistent query answering over more expressive DLs.

Acknowledgments. This research has been partially supported by FET project TONES (Thinking ONtologiES), funded by the EU under contract number FP6-7603, by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant, and by MIUR FIRB 2005 project “Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet” (TOCAL.IT). We finally thank Riccardo Rosati for fruitful discussions on the matter of the present paper, and his helpful comments.

References

1. M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proc. of PODS'99*, pages 68–79, 1999.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the \mathcal{EL} envelope. In *Proc. of IJCAI 2005*, pages 364–369, 2005.
3. 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.
4. L. E. Bertossi, A. Hunter, and T. Schaub, editors. *Inconsistency Tolerance*, volume 3300 of *LNCS*. Springer, 2005.
5. M. Bouzeghoub and M. Lenzerini. Introduction to the special issue on data extraction, cleaning, and reconciliation. *Information Systems*, 26(8):535–536, 2001.
6. 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.
7. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*, pages 602–607, 2005.
8. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR 2006*, pages 260–270, 2006.
9. L. Grieco, D. Lembo, M. Ruzzi, and R. Rosati. Consistent query answering under key and exclusion dependencies: Algorithms and experiments. In *Proc. of CIKM 2005*, pages 792–799, 2005.
10. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From \mathcal{SHIQ} and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.
11. Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In *Proc. of IJCAI 2003*, 2003.
12. D. Lembo and M. Ruzzi. Consistent query answering over description logic ontologies. In *Proc. of the 1st Int. Conf. on Web Reasoning and Rules System (RR 2007)*, 2007. To appear. Available at: <http://www.dis.uniroma1.it/~ruzzi/full.pdf>.
13. M. M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of answering unions of conjunctive queries in \mathcal{SHIQ} . In *Proc. of DL 2006*. CEUR Electronic Workshop Proceedings, <http://ceur-ws.org/Vol-189/>, 2006.
14. B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *Proc. of the 14th Int. World Wide Web Conf. (WWW 2005)*, 2005.
15. N. Shadbolt, W. Hall, and T. Berners-Lee. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.