Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

On the Complexity of Consistent Query Answering in the Presence of Simple Ontologies

Meghyn Bienvenu
Laboratoire de Recherche en Informatique
CNRS & Université Paris Sud, France

Abstract

Consistent query answering is a standard approach for producing meaningful query answers when data is inconsistent. Recent work on consistent query answering in the presence of ontologies has shown this problem to be intractable in data complexity even for ontologies expressed in lightweight description logics. In order to better understand the source of this intractability, we investigate the complexity of consistent query answering for simple ontologies consisting only of class subsumption and class disjointness axioms. We show that for conjunctive queries with at most one quantified variable, the problem is first-order expressible; for queries with at most two quantified variables, the problem has polynomial data complexity but may not be first-order expressible; and for three quantified variables, the problem may become co-NP-hard in data complexity. For queries having at most two quantified variables, we further identify a necessary and sufficient condition for first-order expressibility. In order to be able to handle arbitrary conjunctive queries, we propose a novel inconsistency-tolerant semantics and show that under this semantics, first-order expressibility is always guaranteed. We conclude by extending our positive results to $DL\text{-}Lite$ ontologies without inverse.

1 Introduction

In recent years, there has been growing interest in ontology-based data access, in which the semantic information provided by the ontology is exploited when querying data. Much of the work in this area has focused on ontologies formulated using description logics (DLs). The $DL\text{-}Lite$ family of DLs (Calvanese et al. 2007; Artale et al. 2009)) is considered especially well-suited for such applications due to the fact that query answering can be performed by first incorporating the relevant information from the ontology into the query, and then posing the modified query to the bare data. This property, known as first-order rewritability, means that query answering over $DL\text{-}Lite$ ontologies has very low data complexity, which is key to scalability.

An important issue which arises in ontology-based data access is how to handle data which is inconsistent with the ontology. Ideally, one would like to restore consistency by identifying and correcting the errors in the data (using e.g. techniques for debugging or revising DL knowledge bases, cf. (Schlobach et al. 2007; Qi and Du 2009; Wang, Wang, and Topor 2010) and references therein). However, such an approach presupposes the ability to modify the data and the necessary domain knowledge to determine which part of the data is erroneous. When these conditions are not met (e.g. in information integration applications), a sensible strategy is to adopt an inconsistency-tolerant semantics which allows reasonable answers to be obtained despite the inconsistencies.

The related problem of querying databases which violate integrity constraints has long been studied in the database community (cf. (Arenas, Bertossi, and Chomicki 1999) and the survey (Chomicki 2007)), under the name of *consistent query answering*. The semantics is based upon the notion of a repair, which is a database which satisfies the integrity constraints and is as similar as possible to the original database. Consistent query answering corresponds to evaluating the query in each of the repairs, and then intersecting the results. This semantics is easily adapted to the setting of ontology-based data access, by defining repairs as the inclusion-maximal subsets of the data which are consistent with the ontology.

Consistent query answering for the $DL\text{-}Lite$ family of lightweight DLs was investigated in (Lembo et al. 2010; 2011). The obtained complexity results are rather disheartening: the problem was shown in (Lembo et al. 2010) to be co-NP-hard in data complexity, even for the restricted case of instance queries. Similarly discouraging results were recently obtained in (Rosati 2011) for another prominent lightweight DL $\mathcal{EL}_\bot$ (Baader, Brandt, and Lutz 2005). In fact, we will see in Example 5 that if we consider conjunctive queries, only a single concept disjointness axiom is required to obtain co-NP-hard data complexity.

In the database community, negative complexity results spurred a line of research (Fuxman and Miller 2005; Grieco et al. 2005; Wijsen 2010) aimed at identifying cases where consistent query answering is feasible, and in particular, can be done using first-order query rewriting techniques. The idea is to use targeted polynomial-time procedures whenever possible, and to reserve generic methods with worst-case exponential behavior for difficult cases (see (Grieco et al. 2005) for some experimental results supporting such an approach). A similar investigation for $DL\text{-}Lite$ ontologies was initiated in (Bienvenu 2011), where general conditions

were identified for proving either first-order expressibility or co-NP-hardness of consistent query answering. However, that work considered only instance queries.

The main objective of the present work is to gain a better understanding of what makes consistent conjunctive query answering in the presence of ontologies so difficult. To this end, we conduct a fine-grained complexity analysis which aims to characterize the complexity of consistent query answering based on the properties of the ontology and the query. We focus on simple ontologies, consisting of class subsumption ($A_1 \sqsubseteq A_2$) and class disjointness ($A_1 \sqsubseteq \neg A_2$) axioms, since the problem is already far from trivial for this case. We identify the number of quantified variables in the query as an important factor in determining the complexity of consistent query answering. Specifically, we show that consistent query answering is always first-order expressible for conjunctive queries with at most one quantified variable; the problem has polynomial data complexity (but is not necessarily first-order expressible) when there are two quantified variables; and it may become co-NP-hard starting from three quantified variables. For queries having at most two quantified variables, we further identify a necessary and sufficient condition for first-order expressibility.

To obtain positive results for arbitrary conjunctive queries, we propose a novel inconsistency-tolerant semantics which is a sound approximation of the consistent query answering semantics (and a finer approximation than the approximate semantics proposed in (Lembo et al. 2010)). We show that under this semantics, first-order expressibility of consistent query answering is guaranteed for all conjunctive queries. Finally, in order to treat more expressive ontologies, and to demonstrate the applicability of our techniques, we show how our positive results can be extended to handle $DL\text{-}Lite$ ontologies without inverse roles.

Note that full proofs have been omitted for lack of space but can be found in an appendix available at `http://www.lri.fr/~meghyn/publications`.

2 Preliminaries

Syntax All the ontology languages considered in this paper are fragments of $DL\text{-}Lite_{core}$ (Calvanese et al. 2007; Artale et al. 2009), which we will henceforth abbreviate to $DL\text{-}Lite$. We recall that $DL\text{-}Lite$ knowledge bases (KBs) are built up from a set $\mathsf{N_I}$ of *individuals*, a set $\mathsf{N_C}$ of *atomic concepts*, and a set $\mathsf{N_R}$ of *atomic roles*. Complex concept and role expressions are constructed as follows:

$$B \rightarrow A \mid \exists R \qquad C \rightarrow B \mid \neg B \qquad P \rightarrow R \mid R^-$$

where $A \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$. A *TBox* is a finite set of *inclusions* of the form $B \sqsubseteq C$ (with B, C as above). An *ABox* is a finite set of *assertions* of the form $A(a)$ ($A \in \mathsf{N_C}$) or $R(a, b)$ ($R \in \mathsf{N_R}$), where $a, b \in \mathsf{N_I}$. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individuals in $\mathcal{A}$. A KB consists of a TBox and an ABox.

Semantics An *interpretation* is $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set and $\cdot^\mathcal{I}$ maps each $a \in \mathsf{N_I}$ to $a^\mathcal{I} \in \Delta^\mathcal{I}$, each $A \in \mathsf{N_C}$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, and each $P \in \mathsf{N_R}$ to $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is straightforwardly extended to general concepts and roles, e.g. $(\neg A)^\mathcal{I} = \Delta^\mathcal{I} \setminus A^\mathcal{I}$ and $(\exists S)^\mathcal{I} = \{c \mid \exists d : (c, d) \in S^\mathcal{I}\}$. $\mathcal{I}$ satisfies $G \sqsubseteq H$ if $G^\mathcal{I} \subseteq H^\mathcal{I}$; it satisfies $A(a)$ (resp. $P(a, b)$) if $a^\mathcal{I} \in A^\mathcal{I}$ (resp. $(a^\mathcal{I}, b^\mathcal{I}) \in P^\mathcal{I}$). We write $\mathcal{I} \models \alpha$ if $\mathcal{I}$ satisfies inclusion/assertion α. $\mathcal{I}$ is a *model* of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ satisfies all inclusions in $\mathcal{T}$ and assertions in $\mathcal{A}$. We say a KB $\mathcal{K}$ is *consistent* if it has a model, and that $\mathcal{K}$ *entails* an inclusion/assertion α, written $\mathcal{K} \models \alpha$, if every model of $\mathcal{K}$ is a model of α. The *closure of an ABox $\mathcal{A}$ w.r.t. TBox $\mathcal{T}$*, written $cl_\mathcal{T}(\mathcal{A})$, is the set of assertions which are entailed from $\mathcal{T}, \mathcal{A}$.

In what follows, it will prove useful to extend the notions of satisfaction and entailment to sets of concepts. We will say that a set of concepts $\{C_1, \ldots, C_n\}$ is consistent w.r.t. a TBox $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ and an element $e \in \Delta^\mathcal{I}$ such that $e \in C_i$ for every $1 \leq i \leq n$. Entailment of a concept from a set of concepts is defined in the obvious way: $\mathcal{T} \models S \sqsubseteq D$ if and only if for every model $\mathcal{I}$ of $\mathcal{T}$, we have $\cap_{C \in S} C^\mathcal{I} \subseteq D^\mathcal{I}$.

Queries A *(first-order) query* is a formula of first-order logic with equality, whose atoms are of the form $A(t)$ ($A \in \mathsf{N_C}$), $R(t, t')$ ($R \in \mathsf{N_R}$), or $t = t'$ with t, t' *terms*, i.e., variables or individuals. *Conjunctive queries* (CQs) have the form $\exists \vec{y}\, \psi$, where $\vec{y}$ denotes a tuple of variables, and ψ is a conjunction of atoms of the forms $A(t)$ or $R(t, t')$. *Instance queries* are queries consisting of a single atom with no variables (i.e. ABox assertions). Free variables in queries are called *answer variables*, whereas bound variables are called *quantified variables*. We use $\mathsf{terms}(q)$ to denote the set of terms appearing in a query q.

A *Boolean query* is a query with no answer variables. For a Boolean query q, we write $\mathcal{I} \models q$ when q holds in the interpretation $\mathcal{I}$, and $\mathcal{K} \models q$ when $\mathcal{I} \models q$ for all models $\mathcal{I}$ of $\mathcal{K}$. For a non-Boolean query q with answer variables $v_1, \ldots, v_k$, a tuple of individuals $(a_1, \ldots, a_k)$ is said to be a certain answer for q w.r.t. $\mathcal{K}$ just in the case that $\mathcal{K} \models q[a_1, \ldots, a_k]$, where $q[a_1, \ldots, a_k]$ is the Boolean query obtained by replacing each v_i by a_i. Thus, conjunctive query answering is straightforwardly reduced to entailment of Boolean CQs.

First-order Rewritability It is well-known (cf. (Calvanese et al. 2007; Artale et al. 2009)) that for every $DL\text{-}Lite$ TBox $\mathcal{T}$ and CQ q, we can find a first-order query q' such that for every ABox $\mathcal{A}$ and tuple of individuals $\vec{a}$, we have $\mathcal{T}, \mathcal{A} \models q[\vec{a}]$ if and only if $\mathcal{I}_\mathcal{A} \models q'[\vec{a}]$, where $\mathcal{I}_\mathcal{A}$ denotes the interpretation which has domain $\mathsf{Ind}(\mathcal{A})$ and makes true precisely the assertions in $\mathcal{A}$.

3 Consistent Query Answering

In this section, we formally recall the consistent query answering semantics, present some simple examples which illustrate the difficulty of the problem, and introduce the main problem which will be studied in this paper. For readability, throughout the paper, we will formulate our definitions and results in terms of Boolean CQs, but they can be straightforwardly extended to general CQs.

Consistent query answering relies on the notion of a repair, defined as follows:

Definition 1. *A* repair *of an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ is an*

inclusion-maximal subset $\mathcal{B}$ of $\mathcal{A}$ consistent with $\mathcal{T}$. We use $Rep_{\mathcal{T}}(\mathcal{A})$ to denote the set of repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$.

Consistent query answering can be seen as performing standard query answering on each of the repairs and intersecting the answers. For Boolean queries, the formal definition is as follows:

Definition 2. *A query q is said to be* consistently entailed *from a DL KB $(\mathcal{T}, \mathcal{A})$, written $\mathcal{T}, \mathcal{A} \models_{cons} q$, if $\mathcal{T}, \mathcal{B} \models q$ for every repair $\mathcal{B} \in Rep_{\mathcal{T}}(\mathcal{A})$.*

Just as with standard query entailment, we can ask whether consistent query entailment can be tested by rewriting the query and evaluating it over the data.

Definition 3. *A first-order query q' is a* consistent rewriting *of a Boolean query q w.r.t. a TBox $\mathcal{T}$ if for every ABox $\mathcal{A}$, we have $\mathcal{T}, \mathcal{A} \models_{cons} q$ iff $\mathcal{I}_{\mathcal{A}} \models q'$.*

As mentioned in the introduction, it was shown in (Lembo et al. 2010) that consistent instance checking in $DL\text{-}Lite$ is co-NP-hard in data complexity, which means in particular that consistent rewritings need not exist. We present the reduction in the following example.

Example 4. *Consider an instance $\varphi = c_1 \wedge \ldots \wedge c_m$ of UNSAT, where each c_i is a propositional clause. Let $v_1, \ldots, v_k$ be the propositional variables appearing in φ. We define the $DL\text{-}Lite_{core}$ knowledge base $(\mathcal{T}, \mathcal{A})$ as follows:*

$$\begin{aligned}\mathcal{T} &= \{\, \exists P^- \sqsubseteq \neg\exists N^-,\ \exists P \sqsubseteq \neg\exists U^-,\\ &\quad\ \exists N \sqsubseteq \neg\exists U^-, \exists U \sqsubseteq A \,\}\\ \mathcal{A} &= \{\, U(a, c_i) \mid 1 \le i \le m \,\} \cup\\ &\quad\ \{\, P(c_i, v_j) \mid v_j \in c_i \} \cup \{\, N(c_i, v_j) \mid \neg v_j \in c_i \}\end{aligned}$$

It is not hard to verify that φ is unsatisfiable if and only if $\mathcal{T}, \mathcal{A} \models_{cons} A(a)$. The basic idea is that, because of the inclusion $\exists P^- \sqsubseteq \neg\exists N^-$, each repair corresponds to a valuation of the variables, with v_j assigned true if it has an incoming P-edge in the repair. If a clause c_i is not satisfied by the valuation encoded by the repair, then the individual c_i will have no outgoing P- or N-edges, and hence it will retain its incoming U-edge, causing A to be entailed at a.

The preceding reduction makes crucial use of inverse roles, and indeed, we will show in Section 7 that consistent instance checking is first-order expressible for DL-Lite ontologies without inverse. However, in the case of conjunctive queries, the absence of inverses does not guarantee tractability. Indeed, the next example shows that only a single concept disjointness axiom can yield co-NP-hardness.

Example 5. *We use a variant of UNSAT, called 2+2UNSAT, proved co-NP-hard in (Donini et al. 1994), in which each clause has 2 positive and 2 negative literals, where literals involve either regular variables or the truth constants $\texttt{true}$ and $\texttt{false}$. Consider an instance $\varphi = c_1 \wedge \ldots \wedge c_m$ of 2+2-UNSAT over $v_1, \ldots, v_k$, $\texttt{true}$, and $\texttt{false}$. Let $\mathcal{T} = \{T \sqsubseteq \neg F\}$, and define $\mathcal{A}$ as follows:*

$$\begin{aligned}&\{\, P_1(c_i, u), P_2(c_i, x), N_1(c_i, y), N_2(c_i, z) \mid\\ &\qquad c_i = u \vee x \vee \neg y \vee \neg z, 1 \le i \le m\}\\ &\cup\ \{\, T(v_j), F(v_j) \mid 1 \le j \le k \,\} \cup \{T(\texttt{true}), F(\texttt{false})\}\end{aligned}$$

Then one can show that φ is unsatisfiable just in the case that $(\mathcal{T}, \mathcal{A})$ consistently entails the following query:

$$\begin{aligned}\exists x, y_1, \ldots, y_4\ & P_1(x, y_1) \wedge F(y_1) \wedge P_2(x, y_2) \wedge F(y_2)\\ & \wedge N_1(x, y_3) \wedge T(y_3) \wedge N_2(x, y_4) \wedge T(y_4)\end{aligned}$$

Essentially, $T \sqsubseteq \neg F$ forces the choice of a truth value for each variable, so the repairs of $\mathcal{A}$ correspond exactly to the set of valuations. Importantly, there is only one way to avoid satisfying a 2+2-clause: the first two variables must be assigned false and the last two variables must be assigned true. The existence of such a configuration is checked by q.

We remark that the query in the preceding reduction has quite a simple structure, its only notable property being the use of several quantified variables.

The aim of this paper is to gain a better understanding of what makes consistent conjunctive query answering so difficult (and conversely, what can make it easy). To this end, we will consider the following decision problem:

$$\textsc{Certain}(q, \mathcal{T}) = \{\mathcal{A} \mid \mathcal{T}, \mathcal{A} \models_{cons} q\}$$

and analyze its complexity in terms of the properties of the pair $(q, \mathcal{T})$. We will investigate in particular the impact of limiting the number of quantified variables in q.

In the next three sections, we focus on *simple ontologies*, consisting of axioms of the forms $A_1 \sqsubseteq A_2$ and $A_1 \sqsubseteq \neg A_2$ where $A_1, A_2 \in \mathsf{N_C}$. As Example 5 demonstrates, the problem is already non-trivial in this case. All obtained lower bounds clearly transfer to richer ontologies, and we will show in Section 7 that positive results can also be extended to $DL\text{-}Lite$ ontologies without inverse roles.

4 Tractability for Queries with at Most Two Quantified Variables

In this section, we investigate the complexity of consistent query answering in the presence of simple ontologies for CQs having at most two quantified variables. We show this problem has tractable data complexity, and we provide necessary and sufficient conditions for FO-expressibility.

We begin with queries with at most one quantified variable, showing that a consistent rewriting always exists.

Theorem 6. *Let $\mathcal{T}$ be a simple ontology, and let q be a Boolean CQ with at most one quantified variable. Then* CERTAIN*$(q, \mathcal{T})$ is first-order expressible.*

Proof Sketch. We show how to construct the desired consistent rewriting of q in the case where q has a single quantified variable x. First, for each $t \in \mathsf{terms}(q)$, we set $C_t = \{A \mid A(t) \in q\}$, and we let Σ_t be the set of all $S \subseteq \mathsf{N_C}$ such that every maximal subset $U \subseteq S$ consistent with $\mathcal{T}$ is such that $\mathcal{T} \models U \sqsubseteq C_t$. Intuitively, Σ_t defines the possible circumstances under which the conjunction of concepts in C_t is consistently entailed. We can express this condition with the first-order formula ψ_t:

$$\psi_t = \bigvee_{S \in \Sigma_t} \Big(\bigwedge_{A \in S} A(t) \wedge \bigwedge_{A \in \mathsf{N_C} \setminus S} \neg A(t) \Big)$$

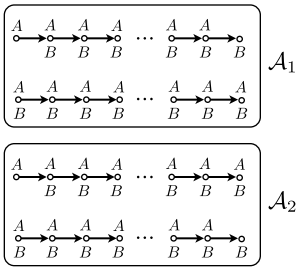

Figure 1: ABoxes for Example 7. Arrows indicate the role R, and each of the four R-chains has length exceeding 2^k.

Now using the ψ_t, we construct q':

$$q' = \exists x \bigwedge_{R(t,t') \in q} R(t,t') \wedge \bigwedge_{t \in \mathsf{terms}(q)} \psi_t$$

It can be shown that q' is indeed a consistent rewriting of q w.r.t. $\mathcal{T}$. To see why this is so, it is helpful to remark that the repairs of $(\mathcal{T}, \mathcal{A})$ contain precisely the role assertions in $\mathcal{A}$, together with a maximal subset of concept assertions consistent with $\mathcal{T}$ for each individual. □

The next example shows that Theorem 6 cannot be extended to the class of queries with two quantified variables.

Example 7. *Consider $q = \exists xy\, A(x) \wedge R(x,y) \wedge B(y)$ and $\mathcal{T} = \{A \sqsubseteq \neg B\}$. Suppose for a contradiction that q' is a consistent rewriting of q w.r.t. $\mathcal{T}$, and let k be the quantifier rank of q'. In Figure 1, we give two ABoxes $\mathcal{A}_1$ and $\mathcal{A}_2$, each consisting of two R-chains of length $> 2^k$. It can be verified that q is consistently entailed from $\mathcal{T}, \mathcal{A}_1$. This is because in every repair, the upper chain will have A at one end, B at the other, and either an A or B at all interior points; every such configuration makes q true somewhere along the chain. On the other hand, we can construct a repair for $\mathcal{T}, \mathcal{A}_2$ which does not entail q by always preferring A on the top chain and B on the bottom chain. It follows that the interpretation $\mathcal{I}_{\mathcal{A}_1}$ satisfies q', whereas $\mathcal{I}_{\mathcal{A}_2}$ does not. However, one can show using standard tools from finite model theory (cf. Ch. 3-4 of (Libkin 2004)) that no formula of quantifier rank k can distinguish $\mathcal{I}_{\mathcal{A}_1}$ and $\mathcal{I}_{\mathcal{A}_2}$, yielding the desired contradiction.*

We can generalize the preceding example to obtain sufficient conditions for the inexistence of a consistent rewriting.

Theorem 8. *Let $\mathcal{T}$ be a simple ontology, and let q be a Boolean CQ with two quantified variables x, y. Assume that there do not exist CQs q_1 and q_2, each with less than two quantified variables, such that $q \equiv q_1 \wedge q_2$. Denote by C_x (resp. C_y) the set of concepts A such that $A(x) \in q$ (resp. $A(y) \in q$). Then* CERTAIN$(q, \mathcal{T})$ *is* not *first-order expressible if there exists $S \subseteq \mathsf{N_C}$ such that:*

- *for $v \in \{x, y\}$, there is a maximal subset $D_v \subseteq S$ consistent with $\mathcal{T}$ s.t. $\mathcal{T} \not\models D_v \sqsubseteq C_v$*
- *for every maximal subset $D \subseteq S$ consistent with $\mathcal{T}$, either $\mathcal{T} \models D \sqsubseteq C_x$ or $\mathcal{T} \models D \sqsubseteq C_y$*

Sketch. The proof generalizes the argument outlined in Example 7. Instead of having a single role connecting successive elements in the chains, we establish the required relational structure for each pair of successive points. We then substitute the set D_y for A, the set D_x for B, and the set S for $\{A, B\}$. The properties of S ensure that if S is asserted at some individual, then we can block the satisfaction of C_x using D_y, and we can block C_y using D_x, but we can never simultaneously block both C_x and C_y. The assumption that q cannot be rewritten as a conjunction of queries with less than two quantified variables is used in the proof of $\mathcal{T}, \mathcal{A}_2 \not\models_{cons} q$ to show that the only possible matches of q involve successive chain elements (and not constants from the query). To show $\mathcal{I}_{\mathcal{A}_1}$ and $\mathcal{I}_{\mathcal{A}_2}$ cannot be distinguished, we use Ehrenfeucht-Fraïssé games, rather than Hanf locality, since the latter is inapplicable when there is a role atom containing a constant and a quantified variable. □

The following theorem shows that whenever the conditions of Theorem 8 are not met, a consistent rewriting exists.

Theorem 9. *Let $\mathcal{T}$ be a simple ontology, and let q be a Boolean CQ with two quantified variables. Then* CERTAIN$(q, \mathcal{T})$ *is first-order expressible if q is equivalent to a conjunction $q_1 \wedge q_2$ of CQs q_1, q_2 each with at most one quantified variable, or if there is no set S satisfying the conditions of Theorem 8.*

Proof Sketch. First suppose q is equivalent to $q_1 \wedge q_2$, where q_1 and q_2 both have at most one quantified variable. Then we can apply Theorem 6 to obtain consistent rewritings q_1' and q_2' of q_1 and q_2 respectively. We can show that $q_1' \wedge q_2'$ is a consistent rewriting for $q_1 \wedge q_2$, hence for q. Thus, the interesting case is when there is no such equivalent query, nor any set S satisfying the conditions of Theorem 8. Intuitively, the inexistence of such a set S ensures that if at some individual, one can block C_x, and one can block C_y, then it is possible to simultaneously block C_x and C_y (compare this to Example 7 in which blocking A causes B to hold, and vice-versa). This property is key, as it allows different potential query matches to be treated independently. □

Together, Theorems 8 and 9 provide a necessary and sufficient condition for the existence of a consistent rewriting. We now reconsider $\mathcal{T}$ and q from Example 7 and outline a polynomial-time method for solving CERTAIN$(q, \mathcal{T})$.

Example 10. *Suppose we have an ABox $\mathcal{A}$, and we wish to decide if $\mathcal{T}, \mathcal{A} \models_{cons} q$, for $\mathcal{T} = \{A \sqsubseteq \neg B\}$ and $q = \exists xy\, A(x) \wedge R(x,y) \wedge B(y)$. The basic idea is to try to construct a repair which does not entail q. We start by iteratively applying the following rules until neither rule is applicable: (1) if $R(a,b), A(a), B(a), B(b) \in \mathcal{A}$ but $A(b) \notin \mathcal{A}$, then delete $A(a)$ from $\mathcal{A}$, and (2) if $R(a,b), A(a), A(b), B(b) \in \mathcal{A}$ but $B(a) \notin \mathcal{A}$, then delete $B(b)$. Note that since the size of $\mathcal{A}$ decreases with every rule application, we will stop after a polynomial number of iterations. Once finished, we check whether there are a, b such that $A(a), R(a,b), B(b) \in \mathcal{A}$, $B(a) \notin \mathcal{A}$, and $A(b) \notin \mathcal{A}$. If so, we return 'yes' (to indicate $\mathcal{T}, \mathcal{A} \models_{cons} q$), and otherwise, we output no' (for $\mathcal{T}, \mathcal{A} \not\models_{cons} q$). Note that in the latter case, for all pairs a, b*

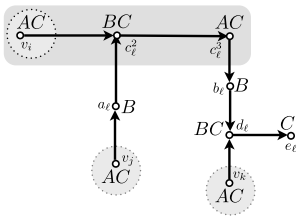

Figure 2: Abox $\mathcal{A}_{c_\ell}$ for clause $c_\ell = \neg v_i \vee \neg v_j \vee \neg v_k$.

with $A(a), R(a,b), B(b) \in \mathcal{A}$, we have both $B(a)$ and $A(b)$. Thus, we can choose to always keep A, thereby blocking all remaining potential matches.

By carefully generalizing the ideas outlined in Example 10, we obtain a tractability result which covers all queries having at most two quantified variables.

Theorem 11. *Let $\mathcal{T}$ be a simple ontology, and let q be a CQ with at most 2 quantified variables. Then* CERTAIN*$(q, \mathcal{T})$ is polynomial in data complexity.*

5 Improved co-NP Lower Bound: Three Quantified Variables Suffice

The objective of this section is to show that the tractability result we obtained for queries with at most two quantified variables cannot be extended further to the class of conjunctive queries with three quantified variables. We will do this by establishing co-NP-hardness for a specific conjunctive query with three quantified variables, thereby improving the lower bound sketched in Example 5. Specifically, we will reduce 3SAT to CERTAIN$(q, \mathcal{T})$ where:

$$\mathcal{T} = \{A \sqsubseteq \neg B, A \sqsubseteq \neg C, B \sqsubseteq \neg C\}$$

$$q = \exists x, y, z \; A(x) \wedge R(x,y) \wedge B(y) \wedge R(y,z) \wedge C(z).$$

The first component of the reduction is a mechanism for choosing truth values for the variables. For this, we create an ABox $\mathcal{A}_{v_i} = \{A(v_i), C(v_i)\}$ for each variable v_i. It is easy to see that there are two repairs for $\mathcal{A}_{v_i}$ w.r.t. $\mathcal{T}$: $\{A(v_i)\}$ and $\{C(v_i)\}$. We will interpret the choice of $A(v_i)$ as assigning true to v_i, and the presence of $C(v_i)$ to mean that v_i is false.

Next we need some way of verifying whether a clause is satisfied by the valuation associated with a repair of $\cup_i \mathcal{A}_{v_i}$. To this end, we create an ABox $\mathcal{A}_{c_\ell}$ for each clause c_ℓ; the ABox $\mathcal{A}_\varphi$ encoding φ will then simply be the union of the ABoxes $\mathcal{A}_{v_i}$ and $\mathcal{A}_{c_\ell}$. The precise definition of the ABox $\mathcal{A}_{c_\ell}$ is a bit delicate and depends on the polarity of the literals in c_ℓ. Figure 2 presents a pictorial representation of $\mathcal{A}_{c_\ell}$ for the case where $c_\ell = \neg v_i \vee \neg v_j \vee \neg v_k$ (the ABoxes $\mathcal{A}_{v_i}$, $\mathcal{A}_{v_j}$, and $\mathcal{A}_{v_k}$ are also displayed).

Let us now see how the ABox $\mathcal{A}_{c_\ell}$ pictured in Figure 2 can be used to test the satisfaction of c_ℓ. First suppose that we have a repair $\mathcal{B}$ of $\mathcal{A}_\varphi$ which contains $A(v_i), A(v_j)$, and $A(v_k)$, i.e. the valuation associated with the repair does not satisfy c_ℓ. We claim that this implies that q holds. Suppose for a contradiction that q is not entailed from $\mathcal{B}, \mathcal{T}$. We first note that by maximality of repairs, $\mathcal{B}$ must contain all of the assertions $A(v_j), R(v_j, a_\ell), B(a_\ell)$, and $R(a_\ell, c_\ell^2)$. It follows that including $C(c_\ell^2)$ in $\mathcal{B}$ would cause q to hold, which means we must choose to include $B(c_\ell^2)$ instead. Using similar reasoning, we can see that in order to avoid satisfying q, we must have $C(d_\ell)$ in $\mathcal{B}$ rather than $B(d_\ell)$, which in turn forces us to select $C(c_\ell^3)$ to block $A(c_\ell^3)$. However, this is a contradiction, since we have identified a match for q in $\mathcal{B}$ with $x = v_i, y = c_\ell^2, z = c_\ell^3$. The above argument (once extended to the other possible forms of $\mathcal{A}_{c_\ell}$) is the key to showing that the unsatisfiability of φ implies $\mathcal{T}, \mathcal{A}_\varphi \models q$.

Conversely, it can be proven that if one of c_ℓ's literals is made true by the valuation, then it is possible to repair $\mathcal{A}_{c_\ell}$ in such a way that a match for q is avoided. For example, consider again $\mathcal{A}_{c_\ell}$ from Figure 2, and suppose that the second literal v_j is satisfied. It follows that $C(v_j) \in \mathcal{B}$, hence $A(v_j) \notin \mathcal{B}$, which means we can keep $C(c_\ell^2)$ rather than $B(c_\ell^2)$, thereby blocking the match at $(v_i, c_\ell^2, c_\ell^3)$. By showing this property holds for the different forms of $\mathcal{A}_{c_\ell}$, and by further arguing that we can combine "q-avoiding" repairs of the $\mathcal{A}_{c_\ell}$ without inducing a match for q, we can prove that the satisfiability of φ implies $\mathcal{T}, \mathcal{A}_\varphi \not\models q$. We thus have:

Theorem 12. CERTAIN*$(q, \mathcal{T})$ is co-NP-hard in data complexity for $\mathcal{T} = \{A \sqsubseteq \neg B, A \sqsubseteq \neg C, B \sqsubseteq \neg C\}$ and $q = \exists x, y, z \; A(x) \wedge R(x,y) \wedge B(y) \wedge R(y,z) \wedge C(z)$.*

6 Tractability through Approximation

The positive results from Section 4 give us a polynomial algorithm for consistent query answering in the presence of simple ontologies, but only for CQs with at most two quantified variables. In order to be able to handle all queries, we explore in this section alternative inconsistency-tolerant semantics which are sound approximations of the consistent query answering semantics[1].

One possibility is to adopt the IAR semantics from (Lembo et al. 2010). We recall that this semantics (denoted by $\models_{IAR}$) can be seen as evaluating queries against the ABox corresponding to the *intersection of the repairs*. Conjunctive query answering under IAR semantics was shown in (Lembo et al. 2011) to be tractable for general CQs in the presence of DL-Lite ontologies (and *a fortiori* simple ontologies) using query rewriting.

To obtain a finer approximation of the consistent query answering semantics, we propose a new inconsistency-tolerant semantics which corresponds to closing repairs with respect to the TBox before intersecting them:

Definition 13. *A Boolean query q is said to be entailed from $(\mathcal{T}, \mathcal{A})$ under ICR semantics ("intersection of closed repairs"), written $\mathcal{T}, \mathcal{A} \models_{ICR} q$, if $\mathcal{T}, \mathcal{D} \models q$, where $\mathcal{D} = \bigcap_{\mathcal{B} \in Rep_\mathcal{T}(\mathcal{A})} cl_\mathcal{T}(\mathcal{B})$.*

The following theorem, which is easy to prove, establishes the relationship among the three semantics.

Theorem 14. *For every Boolean CQ q and TBox $\mathcal{T}$:*

$$\mathcal{T}, \mathcal{A} \models_{IAR} q \quad \Rightarrow \quad \mathcal{T}, \mathcal{A} \models_{ICR} q \quad \Rightarrow \quad \mathcal{T}, \mathcal{A} \models_{cons} q$$

The reverse implications do not hold.

[1] We recall that a semantics $\models_1$ is said to be a *sound approximation* of a semantics $\models_2$ if $\mathcal{K} \models_1 \alpha \Rightarrow \mathcal{K} \models_2 \alpha$ for all $\mathcal{K}, \alpha$.

The next example illustrates how the ICR semantics can preserve information lost by the IAR semantics:

Example 15. *Let $\mathcal{T} = \{A \sqsubseteq C, B \sqsubseteq C, A \sqsubseteq \neg B\}$ and $\mathcal{A} = \{A(a), B(a)\}$. Then $C(a)$ is entailed from $(\mathcal{T}, \mathcal{A})$ under ICR semantics, but not under IAR semantics.*

Finally, we show that under ICR semantics, we can answer any CQ in polynomial time using query rewriting.

Theorem 16. *Let $\mathcal{T}$ be a simple ontology and q a Boolean CQ. Then there exists a first-order query q' such that for every ABox $\mathcal{A}$: $\mathcal{T}, \mathcal{A} \models_{ICR} q$ if and only if $\mathcal{I}_\mathcal{A} \models q'$.*

Proof Sketch. We first compute, using standard techniques, a union of conjunctive queries q' such that for every $\mathcal{A}$, we have $\mathcal{T}, \mathcal{A} \models q$ if and only if $\mathcal{I}_\mathcal{A} \models q'$. Next we use Theorem 6 to find a consistent rewriting $\varphi_{A(t)}$ of each concept atom $A(t) \in q'$, and we let ψ be the first-order query obtained by replacing each occurrence of $A(t)$ in q' by $\varphi_{A(t)}$. It can be shown that the query ψ is such that $\mathcal{T}, \mathcal{A} \models_{ICR} q$ if and only if $\mathcal{I}_\mathcal{A} \models \psi$. □

7 Extension to Inverse-free DL-$Lite$

In this section, we show how the techniques we developed for simple ontologies can be used to extend our positive results to DL-$Lite$ ontologies which do not contain inverse roles (we will use DL-$Lite^{no-}$ to refer to this logic).

Our first result shows that the analogues of Theorems 6 and 11 hold for DL-$Lite^{no-}$ ontologies. The main technical difficulty in adapting the proofs of Theorems 6 and 11 is that role assertions may now be contradicted, which means repairs need not have the same set of role assertions as the original ABox.

Theorem 17. *Consider a DL-$Lite^{no-}$ ontology $\mathcal{T}$, and a Boolean CQ q with at most two quantified variables. Then* CERTAIN*$(q, \mathcal{T})$ is polynomial in data complexity, and first-order expressible if there is at most one quantified variable.*

We can also extend the general first-order expressibility result for the new ICR semantics (Theorem 16) to the class of DL-$Lite^{no-}$ ontologies.

Theorem 18. *Let $\mathcal{T}$ be a DL-$Lite^{no-}$ ontology, and let q be a Boolean CQ. Then there exists a first-order query q' such that for every ABox $\mathcal{A}$: $\mathcal{T}, \mathcal{A} \models_{ICR} q$ if and only if $\mathcal{I}_\mathcal{A} \models q'$.*

Proof Sketch. We apply the same strategy as for Theorem 16, except now we must also replace each role atom $R(t, t')$ in q' by its consistent rewriting. □

As noted earlier, consistent query answering in (full) DL-$Lite$ is co-NP-hard in data complexity even for instance queries, which means that neither of the preceding theorems can be extended to the class of DL-$Lite$ ontologies.

8 Related Work

The principal inspiration for the present paper comes from a line of research in the database community (Fuxman and Miller 2005; Wijsen 2010; Kolaitis and Pema 2012) aimed at deciding for a given set of integrity constraints (typically, functional dependencies) and a given CQ, whether the associated consistent query answering problem is first-order expressible, tractable, or intractable. Although a full characterization has proved elusive, there have been some important recent advances. Notably, a necessary and sufficient condition for first-order expressibility was obtained in (Wijsen 2010) for functional dependencies and the class of acyclic conjunctive queries without self-join. Very recently, a P-co-NP dichotomy was shown for the same setting (Kolaitis and Pema 2012), although only for queries with at most two atoms. Despite the strong similarities in motivation, the setting we consider differs significantly from the one studied by the database community, since we adopt an open world semantics, use different types of constraints, and exploit different restrictions on the query to gain tractability.

We next discuss the relationship with prior work on inconsistency-tolerant query answering for DLs. In (Lembo et al. 2010), four different inconsistency-tolerant query answering semantics (AR, IAR, CAR, ICAR) were studied for DL-$Lite$, and it was shown that CQ answering is co-NP-hard in data complexity for AR and CAR semantics, and first-order expressible for the IAR and ICAR semantics (Lembo et al. 2011). The AR semantics corresponds exactly to the consistent query answering semantics we investigated in this paper, and the IAR semantics is the sound approximation we discussed in Section 6. The CAR and ICAR semantics are defined analogously to the AR and IAR semantics, except that they work on the closure of the input ABox. While this idea is similar to our ICR semantics, in which we close the repairs, the semantics have quite different properties. Indeed, unlike the ICR semantics, the CAR and ICAR semantics are not sound approximations of the consistent query answering semantics. For example, given $\mathcal{T} = \{A \sqsubseteq \neg B, A \sqsubseteq C\}$ and $\mathcal{A} = \{A(a), B(a)\}$, the query $C(a)$ is entailed under CAR and ICAR semantics, but $\mathcal{T}, \mathcal{A} \not\models_{cons} C(a)$. The paraconsistent approach to querying inconsistent DL-$Lite$ knowledge bases recently proposed in (Zhou et al. 2012) also differs considerably from our own, as it does not guarantee that the query result is consistent with the TBox. For example, given $\mathcal{T} = \{A \sqsubseteq \neg B\}$ and $\mathcal{A} = \{A(a), B(a)\}$, both $A(a)$ and $B(a)$ are entailed.

Finally, we note that is a large literature devoted to other forms of inconsistency-handling in description logics, including debugging and revision of DL knowledge bases, both of which aim to modify a KB so as to restore consistency. In general, such methods have the advantage of allowing for the use of standard querying algorithms, which typically have lower complexity than inconsistency-tolerant querying algorithms. However, these approaches are not well-suited to all applications, first, because they presuppose the ability to modify the KB, and second, because they usually require some extra information to decide among the many different ways of restoring consistency. For ontology debugging (Schlobach et al. 2007; Nikitina, Rudolph, and Glimm 2011), this extra information takes the form of a domain expert who is available to answer questions or to decide which modifications should be made. For belief revision operators, the extra information typically takes the form of preferences encoded by an inci-

sion or selection function (Ribeiro and Wassermann 2009). A recently proposed model-based revision operator for DL-Lite$_{bool}$ KBs (Wang, Wang, and Topor 2010) does not require any extra input, but the result of the revision may not be representable as a DL-Lite KB, and approximation can lead to significant loss of information. Finally, it is relevant to note that most prior work on debugging and revision for DLs either focuses uniquely on the TBox, or treats ABox and TBox statements equally, whereas we consider a setting in which the TBox is considered reliable and inconsistencies arise due to errors in the ABox.

9 Conclusion and Future Work

The detailed complexity analysis we conducted for consistent query answering in the presence of simple ontologies provides further insight into the negative complexity results obtained in (Lembo et al. 2010; Rosati 2011), by making clear how little is needed to obtain first-order inexpressibility or intractability. Our investigation also yielded some positive results, including the identification of novel tractable cases, such as inverse-free $DL\text{-}Lite$ ontologies coupled with CQs with at most two quantified variables (or coupled with arbitary CQs, under the new ICR semantics).

There are several natural directions for future work. First, it would be interesting to explore how far we can push our positive results. We expect that adding Horn inclusions and positive role inclusions should be unproblematic, but role disjointness axioms will be more challenging. In order to handle functional roles, we might try to combine our positive results with those which have been obtained for relational databases under functional dependencies (Wijsen 2010). It would also be interesting to try to build upon the results in this paper in order to obtain a criterion for first-order expressibility (or tractability) which applies to all conjunctive queries, regardless of the number of quantified variables.

Finally, we view the present work as a useful starting point in the development of sound but incomplete consistent query answering algorithms for popular lightweight DLs like (full) $DL\text{-}Lite$ and $\mathcal{EL}_\bot$. For example, our results could be extended to identify some CQ-TBox pairs in these richer logics for which consistent query answering is tractable. Another idea would be to use the new ICR semantics to lift tractability results for IQs (like those from (Bienvenu 2011)) to classes of CQs.

10 Acknowledgements

The author would like to thank one of the anonymous reviewers for a careful reading of the proofs, and Université Paris Sud for their financial support (Attractivité grant).

References

Arenas, M.; Bertossi, L. E.; and Chomicki, J. 1999. Consistent query answers in inconsistent databases. In *Proc. of PODS*, 68–79.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The DL-Lite family and relations. *Journal of Artificial Intelligence Research* 36:1–69.

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI*, 364–369.

Bienvenu, M. 2011. First-order expressibility results for queries over inconsistent DL-Lite knowledge bases. In *Proc. of DL*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning* 39(3):385–429.

Chomicki, J. 2007. Consistent query answering: Five easy pieces. In *Proc. of ICDT*, 1–17.

Donini, F. M.; Lenzerini, M.; Nardi, D.; and Schaerf, A. 1994. Deduction in concept languages: From subsumption to instance checking. *Journal of Logic and Computation* 4(4):423–452.

Fuxman, A., and Miller, R. J. 2005. First-order query rewriting for inconsistent databases. In *Proc. of ICDT*, 337–351.

Grieco, L.; Lembo, D.; Rosati, R.; and Ruzzi, M. 2005. Consistent query answering under key and exclusion dependencies: algorithms and experiments. In *Proc. of CIKM*, 792–799.

Kolaitis, P. G., and Pema, E. 2012. A dichotomy in the complexity of consistent query answering for queries with two atoms. *Information Processing Letters* 112(3):77–85.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2010. Inconsistency-tolerant semantics for description logics. In *Proc. of RR*, 103–117.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. Query rewriting for inconsistent DL-Lite ontologies. 155–169.

Libkin, L. 2004. *Elements of Finite Model Theory*. Springer.

Nikitina, N.; Rudolph, S.; and Glimm, B. 2011. Reasoning-supported interactive revision of knowledge bases. In *Proc. of IJCAI*, 1027–1032.

Qi, G., and Du, J. 2009. Model-based revision operators for terminologies in description logics. In *Proc. of IJCAI*, 891–897.

Ribeiro, M. M., and Wassermann, R. 2009. Base revision for ontology debugging. *Journal of Logic and Computation* 19(5):721–743.

Rosati, R. 2011. On the complexity of dealing with inconsistency in description logic ontologies. In *Proc. of IJCAI*, 1057–1062.

Schlobach, S.; Huang, Z.; Cornet, R.; and van Harmelen, F. 2007. Debugging incoherent terminologies. *Journal of Automated Reasoning* 39(3):317–349.

Wang, Z.; Wang, K.; and Topor, R. W. 2010. A new approach to knowledge base revision in DL-Lite. In *Proc. of AAAI*.

Wijsen, J. 2010. On the first-order expressibility of computing certain answers to conjunctive queries over uncertain databases. In *Proc. of PODS*, 179–190.

Zhou, L.; Huang, H.; Ma, Y.; Qi, G.; Huang, Z.; and Qu, Y. 2012. Paraconsistent query answering over DL-Lite ontologies. *Journal of Web Intelligence and Agent Systems* 19–31.