Inconsistency-Tolerant Conjunctive Query Answering for Simple Ontologies

Meghyn Bienvenu

LRI - CNRS & Université Paris-Sud, France
www.lri.fr/∼meghyn/ meghyn@lri.fr

1 Introduction

In recent years, there has been growing interest in using description logic (DL) ontologies to query instance data. An important issue which arises in this setting is how to handle the case in which the data (ABox) is inconsistent with the ontology (TBox). 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. [13]). However, such an approach requires 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), an alternative is to adopt an inconsistency-tolerant semantics in order to obtain reasonable answers despite the inconsistencies.

The related problem of querying databases which violate integrity constraints has long been studied in the database community (cf. [1] and the survey [6]), 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-Lite* family of lightweight DLs was investigated in [10, 11]. The obtained complexity results are rather disheartening: the problem was shown in [10] to be co-NP-hard in data complexity, even for instance queries; this contrasts sharply with the very low AC_0 data complexity for (plain) conjunctive query answering in *DL-Lite*. Similarly discouraging results were recently obtained in [14] for another prominent lightweight DL $\mathcal{EL}_\bot$ [3]. In fact, we will see in Example 1 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 [8, 9, 15] 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 [9] for some experimental results supporting such an approach).

A similar investigation for *DL-Lite* ontologies was initiated in [4], where general conditions were identified for proving either first-order expressibility or coNP-hardness of consistent query answering for a given TBox and instance query.

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 conjunctive 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 coNP-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 [10]). 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 $DL\text{-}Lite_{core}$ ontologies without inverse roles.

Full proofs can be found in a long version available on the author's website.

2 Preliminaries

Syntax. All the ontology languages considered in this paper are fragments of $DL\text{-}Lite_{core}$ [5, 2]. We recall that $DL\text{-}Lite_{core}$ 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:

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

where $A \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$. A *TBox* is a finite set of *inclusions* of the form $B \sqsubseteq C$ (B, C as above). An *ABox* is a finite set of *(ABox) 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. An interpretation $\mathcal{I}$ 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.

We say that a set of concepts $\{C_1, \ldots, C_n\}$ is consistent w.r.t. a TBox $\mathcal{T}$ if there is 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 $\exists \boldsymbol{y}\, \psi$, where $\boldsymbol{y}$ 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 Calvanese et al. [5] proved that for every$DL\text{-}Lite_{core}$ TBox $\mathcal{T}$ and CQ q, there exists a first-order query q' such that for every ABox $\mathcal{A}$ and tuple $\boldsymbol{a}$: $\mathcal{T}, \mathcal{A} \models q[\boldsymbol{a}] \Leftrightarrow \mathcal{I}_{\mathcal{A}} \models q'[\boldsymbol{a}]$, where $\mathcal{I}_{\mathcal{A}}$ denotes the interpretation with domain $\mathsf{Ind}(\mathcal{A})$ that makes true precisely the assertions in $\mathcal{A}$.

3 Consistent Query Answering for Description Logics

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, we will formulate our definitions and results in terms of Boolean CQs, but they can be straightforwardly extended to general CQs.

The key notion underlying consistent query answering semantics is that of a repair of an ABox $\mathcal{A}$, which is an ABox which is consistent with the TBox and as similar as possible to $\mathcal{A}$. In this paper, we follow common practice and use subset inclusion to compare ABoxes.

Definition 1. *A* repair *of a DL 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 Section 1, consistent query answering in *DL-Lite*$_{core}$ is co-NP-hard in data complexity, even for instance queries [10], which means in particular that consistent rewritings need not exist. All known reductions make crucial use of inverse roles, and indeed, we will show in Section 7 that consistent instance checking is first-order expressible for *DL-Lite*$_{core}$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 coNP-hardness.

Example 1. We use a variant of UNSAT, called 2+2UNSAT, proved coNP-hard in [7], in which each clause has 2 positive and 2 negative literals, where literals involve either regular variables or the truth constants `true` and `false`. Consider an instance $\varphi = c_1 \land \ldots \land c_m$ of 2+2-UNSAT over $v_1, \ldots, v_k$, `true`, and `false`. Let $\mathcal{T} = \{T \sqsubseteq \neg F\}$, and define $\mathcal{A}$ as follows:

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

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

$$\exists x y_1 ... \, y_4 P_1(x, y_1) \land F(y_1) \land P_2(x, y_2) \land F(y_2) \land N_1(x, y_3) \land T(y_3) \land N_2(x, y_4) \land T(y_4)$$

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 does not have a particularly complicated structure (in particular, it is tree-shaped). Its only notable property is that it has several quantified variables.

In this paper, we aim 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:

$$\textsc{ConsEnt}(q, \mathcal{T})\text{: Is } \mathcal{A} \text{ such that } \mathcal{T}, \mathcal{A} \models_{cons} q?$$

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

In the next three sections, we focus on *simple ontologies*, consisting of inclusions 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 1 demonstrates, the problem is already non-trivial in this case. All obtained lower bounds transfer to richer ontologies, and we will show in Section 7 that positive results can also be extended to *DL-Lite*$_{core}$ 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 1. *Let $\mathcal{T}$ be a simple ontology, and let q be a Boolean CQ with at most one quantified variable. Then* $\textsc{ConsEnt}(q, \mathcal{T})$ *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:

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

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 1 cannot be extended to the class of queries with two quantified variables.

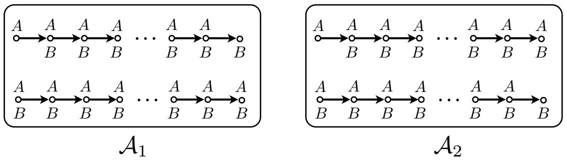

Fig. 1: ABoxes for Example 2. Arrows indicate the role R, and each of the four R-chains has length exceeding 2^k.

Example 2. 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 Fig. 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 [12]) 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 2. *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* ConsEnt*$(q, \mathcal{T})$ 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 $\mathcal{T} \models D \sqsubseteq C_y$*

Proof (Sketch). The proof generalizes the argument outlined in Example 2. 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 2 are not met, a consistent rewriting exists.

Theorem 3. *Let $\mathcal{T}$ be a simple ontology, and let q be a Boolean CQ with two quantified variables x, y. Then* $\textsc{ConsEnt}(q, \mathcal{T})$ *is first-order expressible if q is equivalent to a CQ with at most one quantified variable, or if there is no set S satisfying the conditions of Theorem 2.*

Proof (Sketch). When q is equivalent to a query q' with at most one quantified variable, then Theorem 1 yields a consistent rewriting of q', and hence of q. Thus, the interesting case is when there is no such equivalent query, nor any set S satisfying the conditions of Theorem 2. 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 2 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 2 and 3 provide a necessary and sufficient condition for the existence of a consistent rewriting. We now reconsider $\mathcal{T}$ and q from Example 2 and outline a polynomial-time method for solving $\textsc{ConsEnt}(q, \mathcal{T})$.

Example 3. 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 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 3, we obtain a tractability result which covers all queries having at most two quantified variables.

Theorem 4. *Let $\mathcal{T}$ be a simple ontology, and let q be a CQ with at most 2 quantified variables. Then* $\textsc{ConsEnt}(q, \mathcal{T})$ *is polynomial in data complexity.*

5 An Improved coNP Lower Bound

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

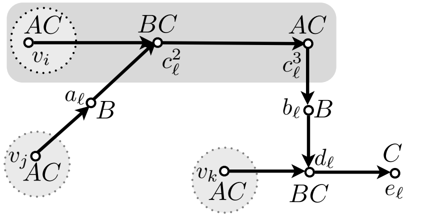

Fig. 2: Abox $\mathcal{A}_{c_\ell}$ for clause $c_\ell = \neg v_i \vee \neg v_j \vee \neg v_k$

to the class of conjunctive queries with three quantified variables. We will do this by establishing coNP-hardness for a specific conjunctive query with three quantified variables, thereby improving the lower bound sketched in Example 1. Specifically, we will reduce 3SAT to $\textsc{ConsEnt}(q, \mathcal{T})$ 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 Fig. 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 $\mathcal{T}, \mathcal{B}$. 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 5. *ConsEnt$(q, \mathcal{T})$ is coNP-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.

One option is to adopt the IAR semantics from [10]. 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 [11] 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. In the following definition, we use $cl_{\mathcal{T}}(\mathcal{B})$ to denote the set of assertions entailed from $\mathcal{T}, \mathcal{B}$.

Definition 4. *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 6. *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.

The next example illustrates the difference between IAR and ICR semantics:

Example 4. 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 conjunctive query in polynomial time using query rewriting.

Theorem 7. *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 $\mathcal{A}$: $\mathcal{T}, \mathcal{A} \models_{ICR} q$ iff $\mathcal{I}_\mathcal{A} \models q'$.*

Proof (Sketch). We first compute, using standard techniques, a union of conjunctive queries φ such that for every $\mathcal{A}$, we have $\mathcal{T}, \mathcal{A} \models q$ if and only if $\mathcal{I}_\mathcal{A} \models \varphi$. Next we use Theorem 1 to find a consistent rewriting $\psi_{A(t)}$ of each concept atom $A(t) \in \varphi$, and we let q' be the first-order query obtained by replacing each occurrence of $A(t)$ in φ by $\psi_{A(t)}$. It can be shown that the query q' is such that $\mathcal{T}, \mathcal{A} \models_{ICR} q$ if and only if $\mathcal{I}_\mathcal{A} \models q'$.

7 Extension to Inverse-Free *DL-Lite*$_{core}$

In this section, we show how the techniques we developed for simple ontologies can be used to extend our positive results to *DL-Lite*$_{core}$ 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 1 and 4 hold for *DL-Lite*$^{no-}$ ontologies. The main technical difficulty in adapting the proofs of Theorems 1 and 4 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 8. *Consider a DL-Lite*$^{no-}$ *ontology $\mathcal{T}$, and a Boolean CQ q with at most two quantified variables. Then* ConsEnt*$(q, \mathcal{T})$ 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 7) to the class of *DL-Lite*$^{no-}$ ontologies.

Theorem 9. *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 $\mathcal{I}_\mathcal{A} \models q'$.*

As noted earlier, consistent query answering in (full) *DL-Lite*$_{core}$ is coNP-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*$_{core}$ ontologies.

8 Conclusion and Future Work

The detailed complexity analysis we conducted for consistent query answering in the presence of simple ontologies provides further insight into previously obtained negative complexity results [10, 14], 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-Lite*$_{core}$ 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 [15]. 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) $DL\text{-}Lite_{core}$ 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 is to use the new ICR semantics to lift tractability results for IQs (like those in [4]) to classes of CQs.

References

1. Arenas, M., Bertossi, L.E., Chomicki, J.: Consistent query answers in inconsistent databases. In: Proc. of PODS. pp. 68–79. ACM Press (1999)
2. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. Journal of Artificial Intelligence Research 36, 1–69 (2009)
3. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of IJCAI. pp. 364–369 (2005)
4. Bienvenu, M.: First-order expressibility results for queries over inconsistent DL-Lite knowledge bases. In: Proc. of DL (2011)
5. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. Journal of Automated Reasoning 39(3), 385–429 (2007)
6. Chomicki, J.: Consistent query answering: Five easy pieces. In: Proc. of ICDT. pp. 1–17 (2007)
7. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Deduction in concept languages: From subsumption to instance checking. Journal of Logic and Computation 4(4), 423–452 (1994)
8. Fuxman, A., Miller, R.J.: First-order query rewriting for inconsistent databases. In: Proc. of ICDT. pp. 337–351 (2005)
9. Grieco, L., Lembo, D., Rosati, R., Ruzzi, M.: Consistent query answering under key and exclusion dependencies: algorithms and experiments. In: Proc. of CIKM. pp. 792–799 (2005)
10. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Proc. of RR. pp. 103–117 (2010)
11. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Query rewriting for inconsistent DL-Lite ontologies. In: Proc. of RR. pp. 155–169 (2011)
12. Libkin, L.: Elements of Finite Model Theory. Springer (2004)
13. Nikitina, N., Rudolph, S., Glimm, B.: Reasoning-supported interactive revision of knowledge bases. In: Proc. of IJCAI. pp. 1027–1032 (2011)
14. Rosati, R.: On the complexity of dealing with inconsistency in description logic ontologies. In: Proc. of IJCAI. pp. 1057–1062 (2011)
15. Wijsen, J.: On the first-order expressibility of computing certain answers to conjunctive queries over uncertain databases. In: Proc. of PODS. pp. 179–190 (2010)