Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Tractable Approximations of Consistent Query Answering for Robust Ontology-Based Data Access

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

Riccardo Rosati
DIAG
Sapienza Università di Roma, Italy

Abstract

A robust system for ontology-based data access should provide meaningful answers to queries even when the data conflicts with the ontology. This can be accomplished by adopting an inconsistency-tolerant semantics, with the consistent query answering (CQA) semantics being the most prominent example. Unfortunately, query answering under the CQA semantics has been shown to be computationally intractable, even when extremely simple ontology languages are considered. In this paper, we address this problem by proposing two new families of inconsistency-tolerant semantics which approximate the CQA semantics from above and from below and converge to it in the limit. We study the data complexity of conjunctive query answering under these new semantics, and show a general tractability result for all known first-order rewritable ontology languages. We also analyze the combined complexity of query answering for ontology languages of the *DL-Lite* family.

1 Introduction

In ontology-based data access (OBDA) [Poggi *et al.*, 2008], an ontology provides an abstract and formal representation of the domain of interest, which is used as a virtual schema when formulating queries over the data. Current research in OBDA mostly focuses on ontology specification languages for which conjunctive query answering is *first-order (FO) rewritable*. In a nutshell, FO-rewritability means that query answering can be performed by rewriting the input query into a first-order query which encodes the relevant knowledge from the ontology, and then evaluating the resulting query over the data. Among FO-rewritable ontology languages, description logics (DLs) of the *DL-Lite* family [Calvanese *et al.*, 2007; Artale *et al.*, 2009] have played an especially prominent role and notably served as the inspiration for the OWL 2 QL profile [1] of the OWL web ontology language.

In real-world applications involving large amounts of data and/or multiple data sources, chances are that the data will be inconsistent with the ontology. Standard OBDA querying algorithms are next to useless in such circumstances, since first-order logic semantics (upon which DLs and standard ontology languages are based) dictates that everything can be derived from a contradiction. Appropriate mechanisms for handling inconsistent data are thus critical to the success of OBDA in practice. Clearly, the best solution is to restore consistency by removing the pieces of data that are responsible for the inconsistencies. However, this strategy cannot always be applied, since the system may not have enough information to localize the errors, or may lack the authorization to modify the data (as is often the case in information integration applications). Thus, a robust OBDA system must be capable of providing meaningful answers to user queries in the presence of inconsistent data.

Recently, several approaches have pursued the idea of adopting an inconsistency-tolerant semantics for OBDA, taking inspiration from the work on *consistent query answering* in databases [Arenas *et al.*, 1999; Bertossi, 2011]. The most well-known and intuitive among such semantics, which we will call the *CQA semantics*, considers as a *repair* of a knowledge base (KB) consisting of an ontology $\mathcal{T}$ and a dataset $\mathcal{A}$, a maximal subset of $\mathcal{A}$ that is consistent with $\mathcal{T}$. Query answering under the CQA semantics then amounts to computing those answers that hold in every repair of the KB. Unfortunately, conjunctive query answering (as well as simpler forms of reasoning) under CQA semantics is computationally hard, even for extremely simple ontology languages for which reasoning under classical semantics is tractable [Lembo *et al.*, 2010; Bienvenu, 2012].

To overcome this computational problem, approximations of the CQA semantics have been recently proposed. In particular, [Lembo *et al.*, 2010; 2011] introduces a sound approximation (called IAR semantics) that evaluates queries over the intersection of all the repairs of the CQA semantics. It was shown that conjunctive query answering under this semantics is tractable (in particular, it is first-order rewritable) for logics of the *DL-Lite* family. However, the IAR semantics has the drawback that it often constitutes a very rough approximation of the CQA semantics, and desirable query answers may be missed. In an effort to obtain more answers than the IAR semantics, a family of parameterized inconsistency-tolerant semantics, called k-lazy consistent semantics, was proposed in [Lukasiewicz *et al.*, 2012a] and shown to converge in the limit to the CQA semantics. However, since the convergence

[1] http://www.w3.org/TR/owl2-profiles/

is not monotone in k, these semantics are not sound approximations of the CQA semantics. Moreover, these semantics do not retain the nice computational properties of the IAR semantics: the polynomial data complexity result shown for linear Datalog+/- ontologies only holds for atomic queries, and it follows from results in [Bienvenu, 2012] that conjunctive query answering under k-lazy consistent semantics is coNP-hard in data complexity, for every $k \geq 1$.

In this paper, we address the above issues and provide the following contributions:

(i) we propose two new families of inconsistency-tolerant semantics, called *k-defeater* and *k-support* semantics, that approximate the CQA semantics from above (complete approximations) and from below (sound approximations), respectively, and converge to the CQA semantics in the limit;

(ii) we study the data complexity of conjunctive query answering under the new semantics, and show a general tractability result for a broad class of ontology languages that includes all known first-order rewritable languages, in particular almost all DLs of the *DL-Lite* family and several rule-based languages of the Datalog+/- family [Calì *et al.*, 2011];

(iii) we analyze the combined complexity of instance checking and conjunctive query answering under the above semantics for ontology languages of the *DL-Lite* family.

The k-support and k-defeater semantics proposed in this paper provide the basis for a semantically grounded and computationally tractable approximation of the CQA semantics in OBDA systems. In particular, we envision a flexible, iterated execution of query q under both k-support and k-defeater semantics with increasing values of k, which stops as soon as the answers to q under both semantics coincide, or when the user is not interested in (or does not want to pay further computational cost for) an exact classification of the tuples that are answers to q under the CQA semantics.

2 Preliminaries

Ontologies and KBs An *ontology* $\mathcal{T}$ is a finite set of first-order logic sentences, and an *ontology (specification) language* $\mathcal{L}$ is a (typically infinite) set of first-order logic sentences. If $\mathcal{T} \subseteq \mathcal{L}$, then $\mathcal{T}$ is called an $\mathcal{L}$ *ontology*. A *knowledge base* (KB) is a pair consisting of an ontology $\mathcal{T}$ and a finite set $\mathcal{A}$ of ground facts. A KB $\langle \mathcal{T}, \mathcal{A} \rangle$ is said to be *consistent* if the first-order theory $\mathcal{T} \cup \mathcal{A}$ has a model. Otherwise, it is *inconsistent*, which we denote by $\langle \mathcal{T}, \mathcal{A} \rangle \models \bot$.

We are interested in the problem of answering instance queries and conjunctive queries over KBs. Without loss of generality, and for ease of exposition, we only consider Boolean queries (i.e. queries without free variables). A first-order (FO) query, or simply *query*, is a first-order sentence. An *instance query (IQ)* is a FO query consisting of a single ground fact. A *conjunctive query (CQ)* is a FO query of the form $\exists \mathbf{x}(\alpha_1 \wedge \ldots \wedge \alpha_n)$ where every α_i is an atom whose arguments are either constants or variables from $\mathbf{x}$. A query q is *entailed* by a KB $\mathcal{K}$ under classical semantics (denoted by $\mathcal{K} \models q$) if q is satisfied in every model of $\mathcal{K}$. The *instance checking problem* consists in deciding, for a KB $\mathcal{K}$ and IQ q, whether $\mathcal{K} \models q$. The *conjunctive query entailment problem* is defined analogously, but with q a CQ.

We introduce some terminology for referring to sets of facts which are responsible for inconsistency or query entailment. A set S of ground facts is called *$\mathcal{T}$-consistent* if $\langle \mathcal{T}, S \rangle \not\models \bot$. A *minimal $\mathcal{T}$-inconsistent subset* of $\mathcal{A}$ is any $S \subseteq \mathcal{A}$ such that $\langle \mathcal{T}, S \rangle \models \bot$ and every $S' \subsetneq S$ is $\mathcal{T}$-consistent. A set of facts $S \subseteq \mathcal{A}$ is said to be a *$\mathcal{T}$-support* for query q in $\mathcal{A}$ if S is $\mathcal{T}$-consistent and $\langle \mathcal{T}, S \rangle \models q$, and it is called a *minimal $\mathcal{T}$-support* for q in $\mathcal{A}$ if no proper subset of S is a $\mathcal{T}$-support for q in $\mathcal{A}$. We sometimes omit "for q" or "in $\mathcal{A}$", when these are understood.

Given a set of ground facts $\mathcal{A}$, we define $\mathcal{I}_\mathcal{A}$ as the interpretation isomorphic to $\mathcal{A}$, i.e., the interpretation defined over the domain of constants occurring in $\mathcal{A}$ and such that the interpretation of every relation R in $\mathcal{I}_\mathcal{A}$ is equal to the set $\{\vec{a} \mid R(\vec{a}) \in \mathcal{A}\}$.

***DL-Lite* ontology languages** We focus on DLs of the *DL-Lite* family [Calvanese *et al.*, 2007; Artale *et al.*, 2009] and recall the syntax and semantics of two specific dialects, called *DL-Lite*[2] and *DL-Lite*$_{Horn}$. A *DL-Lite* ontology consists of a finite set of inclusions $B \sqsubseteq C$, where B and C are defined according to the following syntax:

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

with A a concept name (unary relation) and P a role name (binary relation). In *DL-Lite*$_{Horn}$, inclusions take the form $B_1 \sqcap \ldots \sqcap B_n \sqsubseteq C$, with $B_1, \ldots, B_n$ and C as above.

The classical semantics of *DL-Lite* and *DL-Lite*$_{Horn}$ ontologies is obtained by translating inclusions into first-order sentences using the following function Φ:

$$\begin{array}{rcl} \Phi(A(x)) & = & A(x) \\ \Phi(\exists P(x)) & = & \exists y(P(x,y)) \\ \Phi(\exists P^-(x)) & = & \exists y(P(y,x)) \\ \Phi(\neg B(x)) & = & \neg \Phi(B(x)) \\ \Phi(B_1 \sqcap B_2(x)) & = & \Phi(B_1(x)) \wedge \Phi(B_2(x)) \\ \Phi(C \sqsubseteq D) & = & \forall x(\Phi(C(x)) \rightarrow \Phi(D(x)) \end{array}$$

The classical semantics of a *DL-Lite*$_{Horn}$ KB $\langle \mathcal{T}, \mathcal{A} \rangle$ (and in particular, the notions of model, consistency, and entailment) corresponds to the semantics of the first-order KB $\langle \Phi(\mathcal{T}), \mathcal{A} \rangle$. Note that when considering DL KBs, one typically assumes that the dataset $\mathcal{A}$ uses only unary and binary relations.

First-order rewritability We say that an ontology $\mathcal{T}$ is *first-order (FO) rewritable (for CQ answering)* under semantics $\mathcal{S}$ if, for every CQ q, there exists an effectively computable FO query q' such that, for every set of ground facts $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{A} \rangle$ entails q under semantics $\mathcal{S}$ iff q' is satisfied in $\mathcal{I}_\mathcal{A}$ (in the classical sense). Such a query q' is called a *FO-rewriting* of q relative to $\mathcal{T}$ under semantics $\mathcal{S}$. Moreover, we say that an ontology language $\mathcal{L}$ is *FO-rewritable (for CQ answering)* under semantics $\mathcal{S}$ if every $\mathcal{L}$ ontology is FO-rewritable for CQ answering under $\mathcal{S}$.

Complexity There are two common ways of measuring the complexity of query entailment. The first, called *combined complexity*, is with respect to the size of the whole input

[2] This DL is referred to as *DL-Lite*$_{core}$ in [Calvanese *et al.*, 2007; Artale *et al.*, 2009].

$(\mathcal{T}, \mathcal{A}, q)$, whereas the second, called *data complexity*, is only with respect to the size of $\mathcal{A}$. Our complexity results utilize standard complexity classes, such as NLSPACE, P, NP, and coNP. We also require the following classes which may be less well-known: AC^0 (problems which can be solved by a family of circuits of constant depth and polynomial size, with unlimited fan-in AND gates and OR gates), Π_2^p (problems whose complement is solvable in non-deterministic polynomial time with access to an NP oracle), and $\Delta_2^p[O(\log n)]$ (problems which are solvable in polynomial time with at most logarithmically many calls to an NP oracle).

3 Inconsistency-tolerant Semantics

In this section, we formally introduce the consistent query answering (CQA) semantics and other relevant inconsistency-tolerant semantics.

All of the semantics considered in this paper rely on the notion of a repair, defined as follows:

Definition 1. A *repair* of a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an inclusion-maximal subset of $\mathcal{A}$ that is $\mathcal{T}$-consistent. We use $Rep(\mathcal{K})$ to denote the set of repairs of $\mathcal{K}$.

The repairs of a KB correspond to the different ways of achieving consistency while retaining as much of the original data as possible. Hence, if we consider that the data is mostly reliable, then it is reasonable to assume that one of the repairs accurately reflects the correct portion of the data.

The consistent query answering semantics (also known as the AR semantics [Lembo *et al.*, 2010]) is based upon the idea that, in the absence of further information, a query can be considered to hold if it can be inferred from each of the repairs. Formally:

Definition 2. A query q is entailed by a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the *consistent query answering semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathsf{CQA}} q$, if $\langle \mathcal{T}, \mathcal{B} \rangle \models q$ for every repair $\mathcal{B} \in Rep(\mathcal{K})$.

The following example illustrates the CQA semantics.

Example 1. Consider the *DL-Lite* ontology $\mathcal{T}_{\mathsf{univ}}$:

$$\mathsf{Prof} \sqsubseteq \mathsf{Faculty} \quad \mathsf{Lect} \sqsubseteq \mathsf{Faculty} \quad \mathsf{Fellow} \sqsubseteq \mathsf{Faculty}$$
$$\mathsf{Prof} \sqsubseteq \neg\mathsf{Lect} \quad \mathsf{Prof} \sqsubseteq \neg\mathsf{Fellow} \quad \mathsf{Lect} \sqsubseteq \neg\mathsf{Fellow}$$
$$\mathsf{Prof} \sqsubseteq \exists\mathsf{teaches} \quad \mathsf{Lect} \sqsubseteq \exists\mathsf{teaches} \quad \exists\mathsf{teaches}^- \sqsubseteq \neg\mathsf{Faculty}$$

which states that professors, lecturers, and research fellows are disjoint classes of faculty, that professors and lecturers must teach something, and that whatever is taught is not faculty. Now let $\mathcal{A}_{\mathsf{sam}}$ be as follows:

$$\{\mathsf{Prof}(\mathsf{sam}), \mathsf{Lect}(\mathsf{sam}), \mathsf{Fellow}(\mathsf{sam})\}$$

It is easy to see that KB $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle$ is inconsistent and has three repairs: $\mathcal{R}_1 = \{\mathsf{Prof}(\mathsf{sam})\}$, $\mathcal{R}_2 = \{\mathsf{Lect}(\mathsf{sam})\}$ and $\mathcal{R}_3 = \{\mathsf{Fellow}(\mathsf{sam})\}$. Observe that from each of the repairs, we can infer $q_1 = \mathsf{Faculty}(\mathsf{sam})$, so $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \models_{\mathsf{CQA}} q_1$. However, $q_2 = \exists x.\mathsf{Faculty}(\mathsf{sam}) \wedge \mathsf{teaches}(\mathsf{sam}, x)$ is not entailed from $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{R}_3 \rangle$, so $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \not\models_{\mathsf{CQA}} q_2$.

Unfortunately, while the CQA semantics is intuitively appealing, it is well-known that answering queries under this semantics is usually intractable w.r.t. data complexity [Lembo *et al.*, 2010; Bienvenu, 2012]. This stems from the fact that the number of repairs of $\langle \mathcal{T}, \mathcal{A} \rangle$ may be exponential in the size of $\mathcal{A}$, even when $\mathcal{T}$ is formulated in extremely simple ontology languages.

To overcome the computational problems of the CQA semantics, a sound approximation of it, called the IAR semantics, was proposed in [Lembo *et al.*, 2010].

Definition 3. A query q is entailed by a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the *IAR semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathsf{IAR}} q$, if $\langle \mathcal{T}, \mathcal{D} \rangle \models q$ where $\mathcal{D} = \bigcap_{\mathcal{B} \in Rep(\mathcal{K})} \mathcal{B}$.

The IAR semantics is more conservative than the CQA semantics, as it only uses those facts which are not involved in any contradiction. This has the advantage of yielding query results which are almost surely correct, but also the drawback that some plausible inferences may be missed, as demonstrated by the following example.

Example 2. Reconsider the KB $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle$ and CQ q_1 from Example 1. The intersection of the repairs $\mathcal{R}_1 \cap \mathcal{R}_2 \cap \mathcal{R}_3$ is the empty set, so $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \not\models_{\mathsf{IAR}} q_1$, despite the fact that all the information in $\mathcal{A}_{\mathsf{sam}}$ supports q_1 being true.

From the computational perspective, the IAR semantics can be much better-behaved than the CQA semantics. Indeed, it was shown in [Lembo *et al.*, 2011] that *DL-Lite*$_A$ is FO-rewritable for CQ answering under the IAR semantics, and this result was recently extended to linear Datalog +/- ontologies [Lukasiewicz *et al.*, 2012b].

Finally, to obtain a natural overapproximation of the CQA semantics, we introduce its brave version.

Definition 4. A query q is entailed by a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the *brave semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathsf{brave}} q$, if $\langle \mathcal{T}, \mathcal{B} \rangle \models q$ for some repair $\mathcal{B} \in Rep(\mathcal{K})$.

We illustrate the brave semantics on our running example.

Example 3. As q_2 is entailed by $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{R}_1 \rangle$, we have $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \models_{\mathsf{brave}} q_2$. Also note that every fact in $\mathcal{A}_{\mathsf{sam}}$ appears in some repair, hence, all facts in $\mathcal{A}_{\mathsf{sam}}$ are entailed under the brave semantics.

As Example 3 demonstrates, the brave semantics has the undesirable feature of allowing contradictory statements to be entailed. Nonetheless, this semantics can still serve a useful role by providing a means of showing that a query is *not entailed* under the CQA semantics.

4 Approximations of the CQA Semantics

In this section, we propose two new families of inconsistency-tolerant semantics, which provide increasingly fine-grained under- and over-approximations of the CQA semantics. As these semantics will be shown in Section 5 to enjoy the same nice computational properties as the IAR semantics, our new approach allows us to marry the advantages of the IAR and CQA semantics.

We begin by presenting our new family of sound approximations of the CQA semantics. The intuition is as follows: if a query q is entailed under the CQA semantics, then this is because there is a set $\{S_1, \ldots, S_n\}$ of $\mathcal{T}$-supports for q such that every repair contains some S_i. The k-support semantics we propose is obtained by allowing a maximum of k different supports to be used.

Definition 5. A query q is entailed by $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the *k-support semantics*, written $\mathcal{K} \models_{k\text{-supp}} q$, if there exist (not necessarily distinct) subsets $S_1, \ldots, S_k$ of $\mathcal{A}$ satisfying the following conditions:

- each S_i is a $\mathcal{T}$-support for q in $\mathcal{A}$
- for every $R \in Rep(\mathcal{K})$, there is some S_i with $S_i \subseteq R$

Example 4. The three repairs of $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle$ all use different supports for q_1. We thus have $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \models_{3\text{-supp}} q_1$, but $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \not\models_{2\text{-supp}} q_1$.

The following theorem resumes the important properties of the family of k-support semantics, showing that they interpolate between the IAR and CQA semantics.

Theorem 1. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a KB and q a query. Then:*

1. *$\mathcal{K} \models_{\mathsf{IAR}} q$ if and only if $\mathcal{K} \models_{1\text{-supp}} q$*
2. *$\mathcal{K} \models_{\mathsf{CQA}} q$ if and only if $\mathcal{K} \models_{k\text{-supp}} q$ for some k*
3. *for every $k \geq 0$, if $\mathcal{K} \models_{k\text{-supp}} q$, then $\mathcal{K} \models_{k+1\text{-supp}} q$*

The k-support semantics allows us to approximate more and more closely the set of queries entailed under the CQA semantics, but provides no way of showing that a particular query is *not entailed* under this semantics. This motivates the study of complete approximations of the CQA semantics.

The observation underlying our new family of complete approximations is the following: if a query q is not entailed under the CQA semantics, this is because there is a $\mathcal{T}$-consistent set of facts which contradicts all of the $\mathcal{T}$-supports of q. The k-defeater semantics corresponds to there being no way to construct such a "defeating" set using at most k facts.

Definition 6. A query q is entailed by $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the *k-defeater semantics*, written $\mathcal{K} \models_{k\text{-def}} q$, if there does not exist a $\mathcal{T}$-consistent subset S of $\mathcal{A}$ with $|S| \leq k$ such that $\langle \mathcal{T}, S \cup C \rangle \models \bot$ for every minimal $\mathcal{T}$-support $C \subseteq \mathcal{A}$ of q.

Note that if q has no $\mathcal{T}$-support, then it is not entailed under 0-defeater semantics since one can simply take $S = \emptyset$.

Example 5. We have $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \not\models_{1\text{-def}} q_2$, since by choosing $S = \{\mathsf{Fellow}(\mathsf{sam})\}$, we can invalidate the two minimal $\mathcal{T}$-supports of q_2, which are $\{\mathsf{Prof}(\mathsf{sam})\}$ and $\{\mathsf{Lect}(\mathsf{sam})\}$.

The next theorem shows that the family of k-defeater semantics provides increasingly closer over-approximations of the CQA semantics, starting from the brave semantics presented in Section 3.

Theorem 2. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a KB and q a query. Then:*

1. *$\mathcal{K} \models_{\mathsf{brave}} q$ if and only if $\mathcal{K} \models_{0\text{-def}} q$*
2. *$\mathcal{K} \models_{\mathsf{CQA}} q$ if and only if $\mathcal{K} \models_{k\text{-def}} q$ for every k*
3. *for every $k \geq 1$, if $\mathcal{K} \models_{k+1\text{-def}} q$, then $\mathcal{K} \models_{k\text{-def}} q$*

5 Data Complexity

In this section, we study the data complexity of conjunctive query answering under the k-support and k-defeater semantics. Our main result is the following theorem which shows that, for a broad class of ontology languages, conjunctive query answering under these semantics can be done using FO-rewriting, and hence is in AC^0 w.r.t. data complexity.

Theorem 3. *Let $\mathcal{T}$ be an ontology that is FO-rewritable for CQ answering under classical semantics and such that for every CQ q, there exist ℓ, m such that for every $\mathcal{A}$, every minimal $\mathcal{T}$-support for q relative to $\mathcal{A}$ has cardinality at most ℓ, and every minimal $\mathcal{T}$-inconsistent subset of $\mathcal{A}$ has cardinality at most m. Then:*

(i) for every $k \geq 1$, $\mathcal{T}$ is FO-rewritable for conjunctive query answering under the k-support semantics;

(ii) for every $k \geq 0$, $\mathcal{T}$ is FO-rewritable for conjunctive query answering under the k-defeater semantics.

Proof sketch. Let $\mathcal{T}$ be as stated, and let q be a CQ. By assumption, we can find ℓ and m such that for every $\mathcal{A}$, the minimal $\mathcal{T}$-supports for q relative to $\mathcal{A}$ have cardinality at most ℓ, and the minimal $\mathcal{T}$-inconsistent subsets of $\mathcal{A}$ have cardinality bounded by m. For point (i), a FO-rewriting of q relative to $\mathcal{T}$ for the k-support semantics can be obtained by considering the first-order query $\varphi_q = q_1 \vee \ldots \vee q_n$, where the disjuncts q_i correspond to the different possible choices of k $\mathcal{T}$-supports for q of cardinality at most ℓ, and each q_i asserts that the chosen supports are present in $\mathcal{A}$ and that there is no $\mathcal{T}$-consistent subset of $\mathcal{A}$ of cardinality at most km which conflicts with each of the supports. For point (ii), the desired FO-rewriting of q takes the form $\neg(q_1 \vee \ldots \vee q_n)$, where every q_i asserts the existence of a $\mathcal{T}$-consistent set of facts of cardinality at most k which conflicts with every minimal $\mathcal{T}$-support for q. Here we again utilize the fact that the size of minimal $\mathcal{T}$-supports is bounded by ℓ, and hence there are only finitely many supports to consider. □

Theorem 3 significantly strengthens earlier positive results for the IAR semantics [Lembo *et al.*, 2011; Lukasiewicz *et al.*, 2012a] by covering a full range of semantics and an entire class of practically relevant ontology languages. Indeed, it is easy to verify that *all* ontology languages that are currently known to be first-order rewritable under classical semantics satisfy the hypotheses of Theorem 3: that is, all logics of the original *DL-Lite* family [Calvanese *et al.*, 2007] and almost all members of the extended *DL-Lite* family [Artale *et al.*, 2009], as well as all dialects of Datalog+/- that are known to be FO-rewritable under classical semantics [Calì *et al.*, 2012].

The following examples illustrate the construction of FO-rewritings for the k-support and k-defeater semantics.

Example 6. We consider how to rewrite the CQ q_1 under the k-support semantics. When $k = 1$, we can take as our FO-rewriting the disjunction of the following formulas:

$\mathsf{Faculty}(\mathsf{sam}) \wedge \neg\exists x\, \mathsf{teaches}(x, \mathsf{sam})$

$\mathsf{Prof}(\mathsf{sam}) \wedge \neg\exists x\, \mathsf{teaches}(x, \mathsf{sam}) \wedge \neg\mathsf{Lect}(\mathsf{sam}) \wedge \neg\mathsf{Fellow}(\mathsf{sam})$

$\mathsf{Lect}(\mathsf{sam}) \wedge \neg\exists x\, \mathsf{teaches}(x, \mathsf{sam}) \wedge \neg\mathsf{Prof}(\mathsf{sam}) \wedge \neg\mathsf{Fellow}(\mathsf{sam})$

$\mathsf{Fellow}(\mathsf{sam}) \wedge \neg\exists x\, \mathsf{teaches}(x, \mathsf{sam}) \wedge \neg\mathsf{Lect}(\mathsf{sam}) \wedge \neg\mathsf{Prof}(\mathsf{sam})$

Note that each disjunct expresses that one of the four possible $\mathcal{T}$-supports is present and is not contradicted by other facts. To obtain the rewriting for $k = 2$, we must introduce additional disjuncts which assert that a pair of $\mathcal{T}$-supports is present and cannot be simultaneously contradicted. We obtain three new disjuncts (the other combinations being subsumed by one of the other disjuncts):

$\mathsf{Prof}(\mathsf{sam}) \wedge \mathsf{Lect}(\mathsf{sam}) \wedge \neg\exists x\, \mathsf{teaches}(x, \mathsf{sam}) \wedge \neg\mathsf{Fellow}(\mathsf{sam})$

$\mathsf{Lect}(\mathsf{sam}) \wedge \mathsf{Fellow}(\mathsf{sam}) \wedge \neg\exists x\, \mathsf{teaches}(x, \mathsf{sam}) \wedge \neg\mathsf{Prof}(\mathsf{sam})$
$\mathsf{Fellow}(\mathsf{sam}) \wedge \mathsf{Prof}(\mathsf{sam}) \wedge \neg\exists x\, \mathsf{teaches}(x, \mathsf{sam}) \wedge \neg\mathsf{Lect}(\mathsf{sam})$

Finally, for $k = 3$, we must add further disjuncts to check for the existence of a triple of $\mathcal{T}$-supports which are present and cannot be defeated. In our case, this leads to one new (non-subsumed) disjunct:

$$\mathsf{Prof}(\mathsf{sam}) \wedge \mathsf{Lect}(\mathsf{sam}) \wedge \mathsf{Fellow}(\mathsf{sam}) \wedge \neg\exists x\, \mathsf{teaches}(x, \mathsf{sam})$$

Note that this last disjunct is satisfied in $\mathcal{I}_{\mathcal{A}_{\mathsf{sam}}}$, witnessing the entailment $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \models_{\text{3-supp}} q_1$. Notice also that in this particular example, the CQA and 3-support semantics coincide, and so the FO-rewriting we have constructed is also a FO-rewriting under the CQA semantics.

Example 7. We now consider how to rewrite the query q_2 under the k-defeater semantics. When $k = 0$, the construction yields the following FO-rewriting:

$$\neg\big(\neg(\exists x\, \mathsf{Faculty}(\mathsf{sam}) \wedge \mathsf{teaches}(\mathsf{sam}, x)) \wedge \neg\mathsf{Prof}(\mathsf{sam}) \\ \wedge \neg\mathsf{Lect}(\mathsf{sam}) \wedge \neg(\exists x\, \mathsf{Fellow}(\mathsf{sam}) \wedge \mathsf{teaches}(\mathsf{sam}, x))\big)$$

Inside the negation, there is a single disjunct which asserts that the empty set conflicts with every $\mathcal{T}$-support, or equivalently, that there are no $\mathcal{T}$-supports. When $k = 1$, we must add further disjuncts inside the negation to capture single facts which conflict with all $\mathcal{T}$-supports. In our case, we must add two new disjuncts:

$$\exists x\, \mathsf{teaches}(x, \mathsf{sam}) \qquad \mathsf{Fellow}(\mathsf{sam}) \wedge \neg\mathsf{teaches}(\mathsf{sam}, x)$$

The first disjunct is required since any fact of the form $\mathsf{teaches}(x, \mathsf{sam})$ contradicts $\mathsf{Faculty}(\mathsf{sam})$, and hence every $\mathcal{T}$-support for q_2. The second disjunct treats the case where there is no atom of the form $\mathsf{teaches}(\mathsf{sam}, x)$, in which case the only possible $\mathcal{T}$-supports for q_2 are $\mathsf{Prof}(\mathsf{sam})$ and $\mathsf{Lect}(\mathsf{sam})$, both of which are contradicted by $\mathsf{Fellow}(\mathsf{sam})$. Notice that this last disjunct holds in $\mathcal{I}_{\mathcal{A}_{\mathsf{sam}}}$, which proves that $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{sam}} \rangle \not\models q_2$.

We briefly remark that polynomial data complexity is not preserved under the new semantics. Indeed, in the lightweight DL $\mathcal{EL}_\bot$, CQ answering and unsatisfiability are P-complete w.r.t. data complexity, but it was shown in [Rosati, 2011] that instance checking under the IAR (equiv. 1-support) semantics is coNP-hard w.r.t. data complexity, and it is not hard to show intractability also for the brave (equiv. 0-defeater) semantics.

6 Combined Complexity

To gain further insight into the computational properties of the different inconsistency-tolerant semantics considered in this paper, we study the combined complexity of instance checking and CQ entailment under these semantics for *DL-Lite* and *DL-Lite*$_{Horn}$ KBs.

The results of our analysis are reported in Figure 1. Before presenting the results in more detail, let us begin with some general observations. First, it is interesting to note that for *DL-Lite* KBs, the complexities obtained for the IAR, k-support, brave, k-defeater, and classical semantics all coincide, and are strictly lower than the complexity w.r.t. the CQA semantics. By contrast, for *DL-Lite*$_{Horn}$ KBs, instance checking under any of the considered inconsistency-tolerant semantics is of higher complexity than under classical semantics. Moreover, we lose the symmetry between the sound and complete approximations. Indeed, if we consider CQ entailment, then we find that the complexities of the sound approximations (IAR and k-support) is higher than for the complete approximations (brave and k-defeater semantics).

Finally, we remark that in several cases, and in particular, for the k-support semantics, the complexity for *DL-Lite*$_{Horn}$ is higher than for *DL-Lite*. This can be explained by the fact that for *DL-Lite* KBs, the size of a minimal $\mathcal{T}$-support of a query is linear in the size of the query and independent of $\mathcal{T}$, whereas for *DL-Lite*$_{Horn}$ KBs, the bound on minimal $\mathcal{T}$-supports depends also on the size of $\mathcal{T}$. Overall, these results suggest that while the k-support and k-defeater semantics are tractable w.r.t. data complexity for both *DL-Lite* and *DL-Lite*$_{Horn}$, it will likely be much easier to obtain practical algorithms for *DL-Lite* KBs.

We now present our different complexity results and some brief ideas concerning the proofs. We start by showing that for *DL-Lite*, instance checking under the proposed semantics has the same low complexity as under classical semantics.

Theorem 4. *In DL-Lite, instance checking under the k-support semantics is* NLSPACE*-complete w.r.t. combined complexity, for every $k \geq 1$. The same holds for the k-defeater semantics, for every $k \geq 0$.*

Proof idea. The proof exploits the fact that when $\mathcal{T}$ is a *DL-Lite* ontology, minimal $\mathcal{T}$-supports for IQs consist of single facts, and minimal $\mathcal{T}$-inconsistent subsets contain at most two facts. This means in particular that every k-tuple of minimal $\mathcal{T}$-supports contains at most k facts, and at most k facts are needed to contradict all k supports. This enables a NLSPACE procedure which guesses k facts and verifies that each fact is a $\mathcal{T}$-support, and that there is no set with at most k facts which contradicts all of the guessed facts. The upper bound for the k-defeater semantics uses similar ideas. □

In *DL-Lite*$_{Horn}$, instance checking is intractable already for the IAR and brave semantics, and the lower bounds can be used to show intractability also for the k-support and k-defeater semantics. For the k-defeater semantics, a matching upper bound follows from Theorem 6, while the precise complexity for the k-support semantics remains open.

Theorem 5. *Instance checking in DL-Lite*$_{Horn}$ *is co*NP*-complete w.r.t. combined complexity under the IAR semantics, co*NP*-hard w.r.t. combined complexity under the k-support semantics, and* NP*-complete w.r.t. combined complexity under both the brave semantics and the k-defeater semantics.*

Proof idea. We sketch the coNP lower bound for the IAR semantics, which is by reduction from UNSAT. Let $\varphi = c_1 \wedge \ldots \wedge c_n$ be a propositional CNF over variables $x_1, \ldots, x_m$. Consider the *DL-Lite*$_{Horn}$ KB with

$$\mathcal{T} = \{T_i \sqsubseteq C_j \mid x_i \in c_j\} \cup \{F_i \sqsubseteq C_j \mid \neg x_i \in c_j\} \cup \\ \{T_i \sqcap F_i \sqsubseteq \bot \mid 1 \leq i \leq m\} \cup \{A \sqcap C_1 \sqcap \ldots \sqcap C_n \sqsubseteq \bot\}$$

and $\mathcal{A} = \{A(a)\} \cup \{T_i(a), F_i(a) \mid 1 \leq i \leq m\}$. It is easily verified that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{IAR} A(a)$ iff φ is unsatisfiable. □

We next consider the complexity of CQ entailment under our proposed semantics. For *DL-Lite*, we obtain precisely the same complexity as under the classical semantics.

		classical	IAR	k-supp ($k > 1$)	CQA	k-def ($k > 0$)	brave
IC	*DL-Lite*	NLSPACE	NLSPACE	NLSPACE	coNP	NLSPACE	NLSPACE
	DL-Lite$_{Horn}$	P	coNP	$\geq$ *co*NP, $\leq \Delta_2^p[O(\log n)]$	coNP	NP	NP
CQ	*DL-Lite*	NP	NP	NP	Π_2^p	NP	NP
	DL-Lite$_{Horn}$	NP	$\Delta_2^p[O(\log n)]$	$\Delta_2^p[O(\log n)]$	Π_2^p	NP	NP

Figure 1: Combined complexity of instance checking (IC) and conjunctive query entailment (CQ) under classical semantics and various inconsistency-tolerant semantics. All results are completeness results, unless otherwise noted.

Theorem 6. *In DL-Lite, CQ entailment under the k-support semantics is NP-complete w.r.t. combined complexity, for every $k \geq 1$. For both DL-Lite and DL-Lite$_{Horn}$, CQ entailment under the k-defeater semantics is NP-complete w.r.t. combined complexity, for every $k \geq 1$.*

Proof idea. We sketch the upper bound for the k-defeater semantics. Fix a *DL-Lite$_{Horn}$* KB $\langle \mathcal{T}, \mathcal{A} \rangle$ and a CQ q. Let $S_1, \ldots, S_m$ be the $\mathcal{T}$-consistent subsets of $\mathcal{A}$ with cardinality at most k. Guess a sequence $C_1, \ldots, C_m$ of subsets of $\mathcal{A}$ of cardinality at most $c = 2 \cdot |q| \cdot |\mathcal{T}|$, together with polynomial certificates that $\langle \mathcal{T}, C_i \rangle \models q$, for each C_i. Output yes if for every $1 \leq i \leq m$, the certificate is valid and $S_i \cup C_i$ is $\mathcal{T}$-consistent. As m is polynomial in $|\mathcal{A}|$ (since k is fixed), and both conditions can be verified in polynomial time for *DL-Lite$_{Horn}$* KBs, we obtain an NP procedure. Correctness relies on the fact that because $\mathcal{T}$ is a *DL-Lite$_{Horn}$* ontology, every minimal $\mathcal{T}$-support for q has cardinality at most c. □

For *DL-Lite$_{Horn}$*, CQ entailment under the IAR and k-support semantics rises to $\Delta_2^p[O(\log n)]$-complete.

Theorem 7. *In DL-Lite$_{Horn}$, CQ entailment under the k-support semantics is $\Delta_2^p[O(\log n)]$-complete w.r.t. combined complexity, for every $k \geq 1$.*

Proof idea. The lower bound is by a non-trivial reduction from the Parity(SAT) problem [Wagner, 1987]. For the upper bound, consider the following algorithm which takes as input a *DL-Lite$_{Horn}$* KB $\langle \mathcal{T}, \mathcal{A} \rangle$ and CQ q:

1. For every k-tuple $(\alpha_1, \ldots, \alpha_k) \subseteq \mathcal{A}^k$ of facts, use an NP oracle to decide whether every repair contains some α_i. Let S contain all k-tuples for which the test succeeds.
2. A final oracle call checks if there is a k-tuple $(C_1, \ldots, C_k)$ of subsets of $\mathcal{A}$ of cardinality at most $c = 2 \cdot |\mathcal{T}| \cdot |q|$ such that (i) every C_i is $\mathcal{T}$-consistent and $\langle \mathcal{T}, C_i \rangle \models q$, and (ii) every k-tuple $(\beta_1, \ldots, \beta_k)$ with $\beta_i \in C_i$ belongs to S. Return yes if the call succeeds, else no.

Since every minimal $\mathcal{T}$-support for q contains at most c facts, the algorithm returns yes if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k\text{-supp}} q$. Conversely, if the output is yes, with $(C_1, \ldots, C_k)$ the k-tuple from Step 2, then by (i), every C_i is a $\mathcal{T}$-support for q. Moreover, (ii) ensures that every repair contains some C_i, for it not, we could find some k-tuple $(\beta_1, \ldots, \beta_k) \in C_1 \times \ldots \times C_k$ which does not belong to S, contradicting (ii). Note that the algorithm runs in polynomial time with an NP oracle, since there are only polynomially many k-tuples to consider, for fixed k. As the oracle calls can be organized into a tree, membership in $\Delta_2^p[O(\log n)]$ follows by a result in [Gottlob, 1995]. □

Finally, we determine the combined complexity of instance checking and CQ entailment for the CQA semantics (prior results for this semantics only considered data complexity).

Theorem 8. *For DL-Lite and DL-Lite$_{Horn}$, instance checking (resp. CQ entailment) under the CQA semantics is coNP-complete (resp. Π_2^p-complete) w.r.t. combined complexity.*

Proof idea. The upper bounds are easy: guess a repair and show that it does not entail the query. The coNP-lower bound for instance checking follows from the coNP-hardness of this problem w.r.t. data complexity. The Π_2^p-hardness result involves a non-trivial reduction from 2-QBF validity. □

We should point out that although in this section we focused on two particular members of the *DL-Lite* family, our proofs are quite generic and can be directly used (or trivially extended) to obtain results for a whole rangle of *DL-Lite* dialects (as well as other ontology languages).

7 Conclusion and Future Work

In this paper, we have presented a powerful, flexible, and semantically grounded approach to consistent query answering in ontology-based data access, based upon two novel classes of inconsistency-tolerant semantics. We have shown that our approach is computationally feasible for a large class of practically relevant ontology languages, and in particular, for DLs of the *DL-Lite* family, which underly the OWL 2 QL profile.

The present approach can be extended in several directions. First, we believe that our approach can have a practical impact on OBDA systems, so we aim to implement and experiment with the approach by extending current systems. It would also be very interesting to investigate the connections between our approach and approximate knowledge compilation [Selman and Kautz, 1996]. In particular, it would be important (also for practical purposes) to study the possibility of effectively "compiling" our semantics. It is also relevant to extend our analysis to more complex OBDA systems, where the ontology elements are related to the data sources through complex mappings [Poggi *et al.*, 2008]. Finally, while the present approach is computationally attractive for all known FO-rewritable ontology languages, tractable approximations of the CQA semantics for other tractable yet non-FO-rewritable ontology languages (like $\mathcal{EL}_\bot$ [Baader *et al.*, 2005]) are still missing.

Acknowledgments. The first author has been supported by a Université Paris-Sud Attractivité grant and ANR project PAGODA (ANR-12-JS02-007-01). The second author has been partially supported by EU FP7 project Optique – Scalable End-user Access to Big Data (grant n. FP7-318338).

References

[Arenas *et al.*, 1999] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In *Proc. of PODS*, pages 68–79. ACM Press, 1999.

[Artale *et al.*, 2009] Allessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI*, pages 364–369, 2005.

[Bertossi, 2011] Leopoldo E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[Bienvenu, 2012] Meghyn Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc. of AAAI*, 2012.

[Calì *et al.*, 2011] Andrea Calì, Georg Gottlob, and Andreas Pieris. New expressive languages for ontological query answering. In *Proc. of AAAI*, 2011.

[Calì *et al.*, 2012] Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *Artificial Intelligence*, 193:87–128, 2012.

[Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

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

[Lembo *et al.*, 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proc. of RR*, pages 103–117, 2010.

[Lembo *et al.*, 2011] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Query rewriting for inconsistent DL-Lite ontologies. In *Proc. of RR*, pages 155–169, 2011.

[Lukasiewicz *et al.*, 2012a] Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I. Simari. Inconsistency handling in datalog+/- ontologies. In *Proc. of ECAI*, pages 558–563, 2012.

[Lukasiewicz *et al.*, 2012b] Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I. Simari. Inconsistency-tolerant query rewriting for linear datalog+/-. In *Proc. of Datalog 2.0*, pages 123–134, 2012.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal of Data Semantics*, 10:133–173, 2008.

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

[Selman and Kautz, 1996] Bart Selman and Henry A. Kautz. Knowledge compilation and theory approximation. *Journal of the ACM*, 43(2):193–224, 1996.

[Wagner, 1987] Klaus W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53–80, 1987.