Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Querying Inconsistent Description Logic Knowledge Bases under Preferred Repair Semantics

Meghyn Bienvenu and **Camille Bourgaux**
LRI, CNRS & Université Paris-Sud
Orsay, France

François Goasdoué
IRISA, Université de Rennes 1
Lannion, France

Abstract

Recently several inconsistency-tolerant semantics have been introduced for querying inconsistent description logic knowledge bases. Most of these semantics rely on the notion of a repair, defined as an inclusion-maximal subset of the facts (ABox) which is consistent with the ontology (TBox). In this paper, we study variants of two popular inconsistency-tolerant semantics obtained by replacing classical repairs by various types of preferred repair. We analyze the complexity of query answering under the resulting semantics, focusing on the lightweight logic DL-Lite$_{\mathcal{R}}$. Unsurprisingly, query answering is intractable in all cases, but we nonetheless identify one notion of preferred repair, based upon priority levels, whose data complexity is "only" coNP-complete. This leads us to propose an approach combining incomplete tractable methods with calls to a SAT solver. An experimental evaluation of the approach shows good scalability on realistic cases.

1 Introduction

Description logic (DL) knowledge bases consist of an ontology (called a TBox) expressing conceptual knowledge about a particular domain and a dataset (or ABox) containing facts about particular individuals (Baader et al. 2003). Recent years have seen an increasing interest in performing database-style query answering over DL knowledge bases. Since scalability is crucial in data-rich applications, there has been a trend to using so-called lightweight DLs for which query answering is tractable w.r.t. the size of the ABox. Particular attention has been paid to DLs of the DL-Lite family (Calvanese et al. 2007) which possess the notable property that query answering can be reduced to evaluation of standard database queries.

An important issue that arises in the context of DL query answering is how to handle the case in which the ABox is inconsistent with the TBox. Indeed, while it may be reasonable to assume that the TBox has been properly debugged, the ABox will typically be very large and subject to frequent modifications, both of which make errors likely. Since it may be too difficult or costly to identify and fix these errors, it is essential to be able to provide meaningful answers to queries in the presence of such data inconsistencies. Unfortunately, standard DL semantics is next to useless in such circumstances, as everything is entailed from a contradiction. To address this issue, several different inconsistency-tolerant semantics have been proposed for querying inconsistent DL knowledge bases. Among them, the AR and IAR semantics (Lembo et al. 2010) are the most well-known and well-studied. Both semantics are based upon the notion of a *repair*, defined as an inclusion-maximal subset of the ABox which is consistent with the TBox. The AR semantics amounts to computing those answers that hold no matter which repair is chosen, whereas the more cautious IAR semantics queries the intersection of the repairs.

When additional information on the reliability of ABox assertions is available, it is natural to use this information to identify *preferred repairs*, and to use the latter as the basis of inconsistency-tolerant query answering. In this paper, we investigate variants of the AR and IAR semantics obtained by replacing the classical notion of repair by one of four different types of preferred repairs. Cardinality-maximal repairs are intended for settings in which all ABox assertions are believed to have the same likelihood of being correct. The other three types of preferred repairs target the scenario in which some assertions are considered to be more reliable than others, which can be captured qualitatively by partitioning the ABox into *priority levels* (and then applying either the set inclusion or cardinality criterion to each level), or quantitatively by assigning *weights* to the ABox assertions (and selecting those repairs having the greatest weight).

The first contribution of the paper is a systematic study of the complexity of answering conjunctive and atomic queries under the eight resulting preferred repair semantics. We focus on the lightweight logic DL-Lite$_{\mathcal{R}}$ that underlies the OWL 2 QL profile (Motik et al. 2012), though many of our results can be generalized to other data-tractable ontology languages. For the IAR semantics, the use of preferred repairs significantly impacts complexity: we move from polynomial data complexity in the case of (plain) IAR semantics to coNP-hard data complexity (or worse) for IAR semantics based on preferred repairs. For the AR semantics, query answering is known to be coNP-complete in data complexity already for the standard notion of repairs, but adding preferences often results in even higher complexities. The sole exception is $\subseteq_P$-repairs (which combine priority levels and the set inclusion criterion), for which both AR and IAR query answering are "only" coNP-complete in data complexity.

Our second contribution is a practical approach to query answering in DL-Lite$_\mathcal{R}$ under the AR, $\subseteq_P$-AR, and $\subseteq_P$-IAR semantics. We first show how to encode query answering under these three semantics in terms of propositional unsatisfiability, using a reachability analysis to reduce the size of the encodings. In the CQAPri system we have implemented, a subset of the query results is computed using incomplete tractable methods, and a SAT solver is used to determine the status of the remaining possible answers. An experimental evaluation demonstrates the scalability of the approach in settings we presume realistic. This positive empirical result is due in large part to the efficacy of the incomplete methods, which leave only few cases to be handled by the SAT solver.

Proofs and further details on the experiments can be found in (Bienvenu, Bourgaux, and Goasdoué 2014).

2 Preliminaries

We briefly recall the syntax and semantics of description logics (DLs), and some relevant notions from complexity.

Syntax A DL *knowledge base (KB)* consists of an ABox and a TBox, both of which are constructed from a set $\mathsf{N_C}$ of *concept names* (unary predicates), a set of $\mathsf{N_R}$ of *role names* (binary predicates), and a set $\mathsf{N_I}$ of *individuals* (constants). The *ABox* (dataset) consists of a finite number of *concept assertions* of the form $A(a)$ with $A \in \mathsf{N_C}, a \in \mathsf{N_I}$ and *role assertions* of the form $R(a, b)$ with $R \in \mathsf{N_R}, a, b \in \mathsf{N_I}$. The *TBox* (ontology) consists of a set of axioms whose form depends on the DL in question.

In DL-Lite$_\mathcal{R}$, TBox axioms are either *concept inclusions* $B \sqsubseteq C$ or *role inclusions* $Q \sqsubseteq S$ formed using the following syntax (where $A \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$):

$$B := A \mid \exists Q,\ C := B \mid \neg B,\ Q := R \mid R^-,\ S := Q \mid \neg Q$$

Semantics An *interpretation* has the form $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, 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 $R \in \mathsf{N_R}$ to $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is straightforwardly extended to general concepts and roles, e.g. $(R^-)^\mathcal{I} = \{(c, d) \mid (d, c) \in R^\mathcal{I}\}$ and $(\exists Q)^\mathcal{I} = \{c \mid \exists d : (c, d) \in Q^\mathcal{I}\}$. An interpretation $\mathcal{I}$ satisfies an inclusion $G \sqsubseteq H$ if $G^\mathcal{I} \subseteq H^\mathcal{I}$; it satisfies $A(a)$ (resp. $R(a, b)$) if $a^\mathcal{I} \in A^\mathcal{I}$ (resp. $(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$). An interpretation $\mathcal{I}$ is a *model* of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if $\mathcal{I}$ satisfies all inclusions in $\mathcal{T}$ and assertions in $\mathcal{A}$. A KB $\mathcal{K}$ is *consistent* if it has a model, and we say that an ABox $\mathcal{A}$ is *$\mathcal{T}$-consistent* if the KB $\langle \mathcal{T}, \mathcal{A} \rangle$ is consistent.

Queries Our main focus will be on *conjunctive queries* (CQs) which take the form $\exists \vec{y}\, \psi$, where ψ is a conjunction of atoms of the forms $A(t)$ or $R(t, t')$, where t, t' are variables or individuals, and $\vec{y}$ is a tuple of variables from ψ. A CQ is called *Boolean* if all of its variables are existentially quantified; a CQ consisting of a single atom is an *atomic query* (AQ). When we use the generic term *query*, we mean a CQ. A Boolean CQ q is *entailed* from $\mathcal{K}$, written $\mathcal{K} \models q$, just in the case that q holds in all models of $\mathcal{K}$. For a non-Boolean CQ q with free variables $x_1, \ldots, x_k$, a tuple of individuals $\mathbf{a} = (a_1, \ldots, a_k)$ is a *(certain) answer* for q w.r.t. $\mathcal{K}$ just in the case that $\mathcal{K} \models q[\mathbf{a}]$, where $q[\mathbf{a}]$ is the Boolean query obtained by replacing each x_i by a_i. Thus, CQ answering is straightforwardly reduced to entailment of Boolean CQs. For this reason, we can focus w.l.o.g. on the latter problem.

Complexity There are two common ways of measuring the complexity of query entailment: *combined complexity* is with respect to the size of the whole input $(\mathcal{T}, \mathcal{A}, q)$, whereas *data complexity* is only with respect to the size of $\mathcal{A}$.

In addition to the well-known complexity classes P, NP, and coNP, our results will involve the following classes in the polynomial hierarchy: Δ_2^p (polynomial time using an NP oracle), $\Delta_2^p[O(\log \mathrm{n})]$ (polynomial time with at most logarithmically many NP oracle calls), Σ_2^p (non-deterministic polynomial time with an NP oracle) and its complement Π_2^p.

The following result resumes known results on the complexity of reasoning in DL-Lite$_\mathcal{R}$ under classical semantics.

Theorem 1 (Calvanese et al. 2007). *In DL-Lite$_\mathcal{R}$, consistency and AQ entailment are in* P *w.r.t. combined complexity, and CQ entailment is in* P *w.r.t. data complexity and co*NP*-complete w.r.t. combined complexity.*

3 Preferred Repair Semantics

In this section, we recall two important inconsistency-tolerant semantics and introduce variants of these semantics based upon different notions of preferred repairs. For simplicity, we state the definitions in terms of query entailment.

A central notion in inconsistency-tolerant query answering is that of a repair, which corresponds to a minimal way of modifying the ABox so as to restore consistency. Typically, minimality is defined in terms of set inclusion, yielding:

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.

Several inconsistency-tolerant semantics have been proposed based on this notion of repair. The most well-known, and arguably the most natural, is the AR semantics (Lembo et al. 2010), which was inspired by consistent query answering in relational databases (cf. (Bertossi 2011) for a survey).

Definition 2. A Boolean query q is entailed by $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *AR semantics* if $\langle \mathcal{T}, \mathcal{B} \rangle \models q$ for every repair $\mathcal{B}$ of $\mathcal{K}$.

The intuition underlying the AR semantics is as follows. In the absence of further information, we cannot identify the "correct" repair, and so we only consider a query to be entailed if it can be obtained from each of the repairs.

The IAR semantics (Lembo et al. 2010) adopts an even more cautious approach: only assertions which belong to every repair (or equivalently, are not involved in any contradiction) are considered when answering the query.

Definition 3. A Boolean query q is entailed by a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *IAR semantics* if $\langle \mathcal{T}, \mathcal{B}_\cap \rangle \models q$ where $\mathcal{B}_\cap$ is the intersection of all repairs of $\mathcal{K}$.

It is easy to see that every query that is entailed under IAR semantics is also entailed under AR semantics, but the converse does not hold in general.

The above notion of repair integrates a very simple preference relation, namely set inclusion. When additional information on the reliability of ABox assertions is available, it is natural to use this information to identify *preferred repairs*, and to use the latter as the basis of inconsistency-tolerant

	$\subseteq$	$\leq$	$\subseteq_P$	$\leq_P$	$\leq_w$
AR	coNP	$\mathbf{\Delta_2^p[O(\log n)]}$	**coNP**	$\mathbf{\Delta_2^p}^\dagger$	$\mathbf{\Delta_2^p}^\dagger$
IAR	in P	$\mathbf{\Delta_2^p[O(\log n)]}$	**coNP**	$\mathbf{\Delta_2^p}^\dagger$	$\mathbf{\Delta_2^p}^\dagger$

	$\subseteq$	$\leq$	$\subseteq_P$	$\leq_P$	$\leq_w$
AR	Π_2^p	$\mathbf{\Pi_2^p}$	$\mathbf{\Pi_2^p}$	$\mathbf{\Pi_2^p}$	$\mathbf{\Pi_2^p}$
IAR	NP	$\mathbf{\Delta_2^p[O(\log n)]}$	$\mathbf{\Delta_2^p[O(\log n)]}$	$\mathbf{\Delta_2^p}^\dagger$	$\mathbf{\Delta_2^p}^\dagger$

Figure 1: Data [left] and combined [right] complexity of CQ entailment over DL-Lite KBs under AR and IAR semantics for different types of preferred repairs. For AQs, the data and combined complexity coincide with the data complexity for CQs. All results are completeness results unless otherwise noted. New results in bold. † $\Delta_2^p[O(\log \text{n})]$-complete under the assumption that there is a bound on the number of priority classes (resp. maximal weight).

reasoning. This idea leads us generalize the earlier definitions[1], using preorders to model preference relations.

Definition 4. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a KB, and let $\preceq$ be a preorder over subsets of $\mathcal{A}$. A *$\preceq$-repair of* $\mathcal{K}$ is a $\mathcal{T}$-consistent subset of $\mathcal{A}$ which is maximal w.r.t. $\preceq$. The set of $\preceq$-repairs of $\mathcal{K}$ is denoted $Rep_{\preceq}(\mathcal{K})$.

Definition 5. A Boolean query q is entailed by $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the *$\preceq$-AR semantics*, written $\mathcal{K} \models_{\preceq\text{-AR}} q$, if $\langle \mathcal{T}, \mathcal{B} \rangle \models q$ for every $\mathcal{B} \in Rep_{\preceq}(\mathcal{K})$; it is entailed by $\mathcal{K}$ under the *$\preceq$-IAR semantics*, written $\mathcal{K} \models_{\preceq\text{-IAR}} q$, if $\langle \mathcal{T}, \mathcal{B}_\cap \rangle \models q$ where $\mathcal{B}_\cap = \bigcap_{\mathcal{B} \in Rep_{\preceq}(\mathcal{K})} \mathcal{B}$.

In this paper, we consider four standard ways of defining preferences over subsets, cf. (Eiter and Gottlob 1995).

Cardinality ($\leq$) A first possibility is to compare subsets using set cardinality: $\mathcal{A}_1 \leq \mathcal{A}_2$ iff $|\mathcal{A}_1| \leq |\mathcal{A}_2|$. The resulting notion of $\leq$-repair is appropriate when all assertions are believed to have the same (small) likelihood of being erroneous, in which case repairs with the largest number of assertions are most likely to be correct.

Priority levels ($\subseteq_P$, $\leq_P$) We next consider the case in which ABox assertions have been partitioned into priority levels $\mathcal{P}_1, \ldots, \mathcal{P}_n$ based on their perceived reliability, with assertions in $\mathcal{P}_1$ considered most reliable, and those in $\mathcal{P}_n$ least reliable. Such a prioritization can be used to separate a part of the dataset that has already been validated from more recent additions. Alternatively, one might stratify assertions based upon the concept or role names they use (when some predicates are known to be more reliable), or the data sources from which they originate (in information integration applications). Given a prioritization $P = \langle \mathcal{P}_1, \ldots, \mathcal{P}_n \rangle$ of $\mathcal{A}$, we can refine the $\subseteq$ and $\leq$ preorders as follows:

- Prioritized set inclusion: $\mathcal{A}_1 \subseteq_P \mathcal{A}_2$ iff $\mathcal{A}_1 \cap \mathcal{P}_i = \mathcal{A}_2 \cap \mathcal{P}_i$ for every $1 \leq i \leq n$, or there is some $1 \leq i \leq n$ such that $\mathcal{A}_1 \cap \mathcal{P}_i \subsetneq \mathcal{A}_2 \cap \mathcal{P}_i$ and for all $1 \leq j < i$, $\mathcal{A}_1 \cap \mathcal{P}_j = \mathcal{A}_2 \cap \mathcal{P}_j$.
- Prioritized cardinality: $\mathcal{A}_1 \leq_P \mathcal{A}_2$ iff $|\mathcal{A}_1 \cap \mathcal{P}_i| = |\mathcal{A}_2 \cap \mathcal{P}_i|$ for every $1 \leq i \leq n$, or there is some $1 \leq i \leq n$ such that $|\mathcal{A}_1 \cap \mathcal{P}_i| < |\mathcal{A}_2 \cap \mathcal{P}_i|$ and for all $1 \leq j < i$, $|\mathcal{A}_1 \cap \mathcal{P}_j| = |\mathcal{A}_2 \cap \mathcal{P}_j|$.

Notice that a single assertion on level $\mathcal{P}_i$ is preferred to any number of assertions from $\mathcal{P}_{i+1}$, so these preorders are best suited for cases in which there is a significant difference in the perceived reliability of adjacent priority levels.

[1]Ours is not the first work to consider preferred repairs – see Section 7 for references and discussion.

Weights ($\leq_w$) The reliability of different assertions can be modelled quantitatively using a function $w : \mathcal{A} \rightarrow \mathbb{N}$ assigning weights to the ABox assertions. The weight function w induces a preorder $\leq_w$ over subsets of $\mathcal{A}$ in the expected way: $\mathcal{A}_1 \leq_w \mathcal{A}_2$ iff $\sum_{\alpha \in \mathcal{A}_1} w(\alpha) \leq \sum_{\alpha \in \mathcal{A}_2} w(\alpha)$. If the ABox is populated using information extraction techniques, the weights may be derived from the confidence levels output by the extraction tool. Weight-based preorders can also be used in place of the $\leq_P$ preorder to allow for compensation between the priority levels.

4 Complexity Results

In this section, we study the complexity of query entailment under preferred repair semantics. We focus on knowledge bases formulated in DL-Lite$_\mathcal{R}$, since it is a popular DL for OBDA applications and the basis for OWL 2 QL (Motik et al. 2012). However, many of our results hold also for other DLs and ontology languages (see end of section).

Figure 1 recalls existing results for query entailment under the standard AR and IAR semantics and presents our new results for the different preferred repair semantics.

Theorem 2. *The results in Figure 1 hold.*

Proof idea. The upper bounds for AR-based semantics involve guessing a preferred repair that does not entail the query; for the IAR-based semantics, we guess preferred repairs that omit some ABox assertions and verify that the query is not entailed from the remaining assertions. For the lower bounds, we were able to adapt some proofs from (Bienvenu 2012; Bienvenu and Rosati 2013); the $\Delta_2^p[O(log\ n)]$ lower bounds proved the most challenging and required significant extensions of existing reductions. ☐

Let us briefly comment on the obtained results. Concerning data complexity, we observe that for preferred repairs, the IAR semantics is just as difficult as the AR semantics. This is due to the fact that there is no simple way of computing the intersection of preferred repairs, whereas this task is straightforward for $\subseteq$-repairs. However, if we consider combined complexity, we see that the IAR semantics still retains some computational advantage over AR semantics. This lower complexity translates into a concrete practical advantage: for the IAR semantics, one can precompute the intersection of repairs in an offline phase and then utilize standard querying algorithms at query time, whereas no such precomputation is possible for the AR semantics. Finally, if we compare the different types of preferred repairs, we find that the $\subseteq_P$ preorder leads to the lowest complexity, and $\leq_P$ and $\leq_w$ the greatest. However, under the reasonable assumption that there is a bound on the number of priority classes (resp. maximal weight), we obtain the same complexity for the semantics based on $\leq$-, $\leq_P$- and $\leq_w$-repairs.

We should point out that the only properties of DL-Lite$_\mathcal{R}$ that are used in the upper bound proofs are those stated in Theorem 1. Consequently, our combined complexity upper bounds apply to all ontology languages having polynomial combined complexity for consistency and instance checking and NP combined complexity for CQ entailment, and in particular to the OWL 2 EL profile (Motik et al. 2012). Our data complexity upper bounds apply to all data-tractable ontology languages, which includes Horn DLs (Hustadt, Motik, and Sattler 2007; Eiter et al. 2008) and several dialects of Datalog +/- (Calì, Gottlob, and Lukasiewicz 2012).

5 Query Answering via Reduction to SAT

In this section, we show how to answer queries over DL-Lite$_\mathcal{R}$ KBs under $\subseteq_P$-AR and $\subseteq_P$-IAR semantics by translation to propositional unsatisfiability. We chose to focus on $\subseteq_P$-repairs as they offer the lowest complexity among the different forms of preferred repairs, and seem natural from the point of view of applications.

To simplify the presentation of our encodings, we introduce the notions of conflicts of a KB and causes of a query.

Definition 6. A *conflict* for $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a minimal $\mathcal{T}$-inconsistent subset of $\mathcal{A}$. A *cause* for a Boolean CQ q is a minimal $\mathcal{T}$-consistent subset $\mathcal{C} \subseteq \mathcal{A}$ such that $\langle \mathcal{T}, \mathcal{C} \rangle \models q$.

Fact 1. In DL-Lite$_\mathcal{R}$, conflicts have cardinality at most two.

The encodings presented in this section use variables to represent ABox assertions, so that each valuation corresponds to a subset of the ABox. In the case of the $\subseteq_P$-AR semantics, the most obvious encoding would stipulate that the subset corresponding to a valuation (i) contains no cause for q, (ii) is maximal w.r.t. $\subseteq_P$, and (iii) contains no conflicts. However, such an encoding would contain as many variables as ABox facts, even though most of the ABox may be irrelevant for answering the query at hand.

In order to identify potentially relevant assertions, we introduce the notion of an oriented conflict graph (inspired by the conflict hypergraphs from (Chomicki, Marcinkowski, and Staworko 2004)). In what follows, we use $\alpha \preceq_P \beta$ to signify that there exist $i \leq j$ such that $\alpha \in \mathcal{P}_i$ and $\beta \in \mathcal{P}_j$.

Definition 7. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ KB and P be a prioritization of $\mathcal{A}$. The *oriented conflict graph for $\mathcal{K}$ and P*, denoted $G^P_\mathcal{K}$, is the directed graph whose set of vertices is $\mathcal{A}$ and which has an edge from β to α whenever $\alpha \preceq_P \beta$ and $\{\alpha, \beta\}$ is a conflict for $\mathcal{K}$.

We now give a more succinct encoding, which can be seen as selecting a set of assertions that contradict all of the query's causes (thereby ensuring that no cause is present), and verifying that this set can be extended to a $\subseteq_P$-repair. Importantly, to check the latter, it suffices to consider only those assertions that are reachable in $G^P_\mathcal{K}$ from an assertion that contradicts some cause.

Theorem 3. *Let q be a Boolean CQ , $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ KB, and $P = \langle \mathcal{P}_1, \ldots, \mathcal{P}_n \rangle$ be a prioritization of $\mathcal{A}$. Consider the following propositional formulas having variables of the form x_α for $\alpha \in \mathcal{A}$:*

$$\varphi_{\neg q} = \bigwedge_{\mathcal{C} \in \mathsf{causes}(q)} \Big(\bigvee_{\alpha \in \mathcal{C}} \bigvee_{\substack{\beta \in \mathsf{confl}(\alpha) \\ \beta \preceq_P \alpha}} x_\beta \Big)$$

$$\varphi_{\mathsf{max}} = \bigwedge_{\alpha \in R_q} \Big(x_\alpha \vee \bigvee_{\substack{\beta \in \mathsf{confl}(\alpha) \\ \beta \preceq_P \alpha}} x_\beta \Big)$$

$$\varphi_{\mathsf{cons}} = \bigwedge_{\substack{\alpha, \beta \in R_q \\ \beta \in \mathsf{confl}(\alpha)}} \neg x_\alpha \vee \neg x_\beta$$

where $\mathsf{causes}(q)$ contains all causes for q in $\mathcal{K}$, $\mathsf{confl}(\alpha)$ contains all assertions β such that $\{\alpha, \beta\}$ is a conflict for $\mathcal{K}$, and R_q is the set of assertions reachable in $G^P_\mathcal{K}$ from some assertion β such that x_β appears in $\varphi_{\neg q}$. Then $\mathcal{K} \models_{\subseteq_P\text{-}AR} q$ iff $\varphi_{\neg q} \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$ is unsatisfiable.

We observe that for the plain AR semantics, we can further simplify the encoding by dropping the formula φ_{max}.

For the $\subseteq_P$-IAR semantics, a query is not entailed just in the case that every cause is absent from some $\subseteq_P$-repair. This can be tested by using one SAT instance per cause.

Theorem 4. *Let q, $\mathcal{K}$, P, $\mathsf{causes}(q)$, and $\mathsf{confl}(\alpha)$ be as in Theorem 3. For each $\mathcal{C} \in \mathsf{causes}(q)$, consider the formulas:*

$$\varphi_{\neg \mathcal{C}} = \bigvee_{\alpha \in \mathcal{C}} \bigvee_{\substack{\beta \in \mathsf{confl}(\alpha) \\ \beta \preceq_P \alpha}} x_\beta$$

$$\varphi^{\mathcal{C}}_{\mathsf{max}} = \bigwedge_{\alpha \in R_\mathcal{C}} \Big(x_\alpha \vee \bigvee_{\substack{\beta \in \mathsf{confl}(\alpha) \\ \beta \preceq_P \alpha}} x_\beta \Big)$$

$$\varphi^{\mathcal{C}}_{\mathsf{cons}} = \bigwedge_{\substack{\alpha, \beta \in R_\mathcal{C} \\ \beta \in \mathsf{confl}(\alpha)}} \neg x_\alpha \vee \neg x_\beta$$

where $R_\mathcal{C}$ is the set of assertions reachable in $G^P_\mathcal{K}$ from some assertion β such that x_β appears in $\varphi_{\neg \mathcal{C}}$. Then $\mathcal{K} \models_{\subseteq_P\text{-}IAR} q$ iff there exists $\mathcal{C} \in \mathsf{causes}(q)$ such that the formula $\varphi_{\neg \mathcal{C}} \wedge \varphi^{\mathcal{C}}_{\mathsf{max}} \wedge \varphi^{\mathcal{C}}_{\mathsf{cons}}$ is unsatisfiable.

The above encodings can be used to answer *non-Boolean queries* using the standard reduction to the Boolean case: a tuple $\mathbf{a}$ is an answer to a non-Boolean CQ q iff the Boolean query $q[\mathbf{a}]$ is entailed under the considered semantics.

6 Experimental Evaluation

We implemented our query answering framework in Java within our CQAPri ("Consistent Query Answering with Priorities") tool. CQAPri is built on top of the relational database server PostgreSQL, the Rapid query rewriting engine for DL-Lite (Chortaras, Trivela, and Stamou 2011), and the SAT4J v2.3.4 SAT solver (Berre and Parrain 2010).

CQAPri stores the ABox in PostgreSQL, while it keeps the TBox in-memory together with the pre-computed set of conflicts for the KB. Conflicts are computed by evaluating over the ABox the SQLized rewritings of the queries looking for counter-examples to the negative TBox inclusions. They are stored as an oriented conflict graph (Definition 7), built from a single priority level for the IAR and AR semantics, and multiple levels for the $\subseteq_P$-IAR and $\subseteq_P$-AR semantics.

ABox id	#ABox	%conflicts	avg conflicts	o.c. graph (ms)
u1p15e-4	75708	2.05	0.04	3844
u1p5e-2	76959	30.97	1.03	4996
u1p2e-1	80454	57.99	3.96	6224
u5p15e-4	499674	1.70	0.03	19073
u5p5e-2	507713	33.12	1.21	24600
u5p2e-1	531607	58.29	4.29	32087
u10p15e-4	930729	2.37	0.05	33516
u10p5e-2	945450	33.92	1.31	43848
u10p2e-1	988882	58.89	4.86	62028
u20p15e-4	1982922	2.64	0.05	95659
u20p5e-2	2014129	33.91	1.6	122805
u20p2e-1	2103366	58.78	5.49	192450

Table 1: ABoxes in terms of size, percentage of assertions in conflicts, average number of conflicts per assertion, and time to compute the oriented conflict graph.

When a query arrives, CQAPri evaluates it over the ABox using its SQLized rewriting, to obtain its *possible answers* and their causes. Possible answers define a superset of the answers holding under the IAR, AR, $\subseteq_P$-IAR, and $\subseteq_P$-AR semantics. Among the possible answers, CQAPri identifies the IAR ones, or an approximation of the $\subseteq_P$-IAR ones, by checking whether there is some cause whose assertions have no outgoing edges in the oriented conflict graph. For those possible answers that are not found to be IAR answers, resp. in the approximation of the $\subseteq_P$-IAR answers, deciding whether they are entailed under the AR semantics, resp. the $\subseteq_P$-IAR and $\subseteq_P$-AR semantics, is done using the SAT encodings from the preceding section.

Experimental setting

TBox and Datasets We considered the modified LUBM benchmark from Lutz et al. (2013), which provides the DL-Lite$_\mathcal{R}$ version $\text{LUBM}^{\exists}_{20}$ of the original LUBM $\mathcal{ELI}$ TBox, and the Extended University Data Generator (EUDG) v0.1a (both available at www.informatik.uni-bremen.de/~clu/combined). We extended $\text{LUBM}^{\exists}_{20}$ with negative inclusions, to allow for contradictions. Inconsistencies in the ABox were introduced by contradicting the presence of an individual in a concept assertion with probability p, and the presence of each individual in a role assertion with probability $p/2$. Additionally, for every role assertion, its individuals are switched with probability $p/10$. Prioritizations of ABox were made to capture a variety of scenarios.

Table 1 describes the ABoxes we used for our experiments. Every ABox's id `uXpY` indicates the number `X` of universities generated by EUDG and the probability value `Y` of p for adding inconsistencies as explained above (`Me-P` reads $M.10^{-P}$). We chose the values for `X` and `Y` so as to have ABoxes of size varying from small to large, and a number of conflicts ranging from a value we found realistic up to values challenging our approach. We built 8 prioritizations for each of these ABoxes further denoted by the `id` of the ABox it derives from, and a suffix first indicating the number of priority levels and then how these levels were chosen. `lZdW` indicates the number `Z` of priority levels: 3 and 10 in our experiments, and the distribution `W`: $\texttt{cr}^=$, $\texttt{a}^=$, $\texttt{cr}^{\neq}$, or $\texttt{a}^{\neq}$ indicates whether priority levels were chosen per concept/role (`cr`) or assertion (`a`), and whether choosing between these levels was equiprobable ($^=$) or not ($^{\neq}$).

Queries We used the queries described in Table 2 for our experiments. Some queries were borrowed from LUBM-based experiments found in the literature: `Lutz1` and `Lutz5` come from (Lutz et al. 2013), and `req2` and `req3` are from (Pérez-Urbina, Horrocks, and Motik 2009). The other queries we designed ourselves. They show a variety of structural aspects and rewriting sizes; they yield enough possible answers to observe the behavior of the considered semantics.

Query id	shape	#atoms	#vars	#rews	rew time (ms)
req2	chain	3	2	1	0
req3	dag	6	3	23	4
g2	atomic	1	1	44	0
g3	atomic	1	1	44	0
q1	chain	2	2	80	0
q2	chain	2	2	44	15
q4	dag	7	6	25	16
Lutz1	dag	8	4	3887	328
Lutz5	tree	5	3	667	16

Table 2: Queries in terms of shape, numbers of atoms variables, number of rewritings, and rewriting time (Rapid).

Experimental results

We summarize the general tendencies we observed. The main conclusion is that our approach scales when the proportion of conflicting assertions is a few percent, as is likely to be the case in most real applications.

IAR and AR query answering We observed that the AR semantics only adds a limited number of new answers compared to the IAR semantics. For 60% of our ABox and query pairs, AR does not provide any additional answers, and it provides at most as many new answers as IAR ones.

Also, for a given number of universities (i.e., size), when the proportion of conflicting assertions increases, the number of IAR answers decreases, while the number of AR non-IAR and of possible non-AR ones increases. Such an increase significantly augments the time spent identifying AR non-IAR answers using the SAT solver, as exemplified in Figure 2 [left]. It explains that the lower the probability for generating conflicts, the more AR query answering times show a linear behavior w.r.t. ABox size (i.e., scales), up to the `uXp15e-4` and `uXp5e-2` ABoxes in our experiments.

$\subseteq_P$-IAR and $\subseteq_P$-AR query answering Similarly to above, the $\subseteq_P$-AR semantics adds a limited number of answers compared to the $\subseteq_P$-IAR semantics. Moreover, in most cases, the approximation of $\subseteq_P$-IAR using the ordered conflict graph identifies a large share of the $\subseteq_P$-IAR answers.

We also observed that adding prioritizations to the ABoxes complicates query answering, and using 3 priority levels typically led to harder instances than using 10 levels.

Our approach scales up to the `uXp15e-4lYdZ` set of ABoxes, as answering queries against them requires in most cases less than twice the time observed for the (plain) AR semantics. Figure 2 [right] exemplifies the observed trend for the `uXp5e-2lYdZ` ABoxes, where the use of priority levels significantly increases runtimes, as well as the number of queries running out of memory or producing a time-out.

7 Related Work

The closest related work is that of (Du, Qi, and Shen 2013) who study query answering under $\leq_w$-AR semantics for the

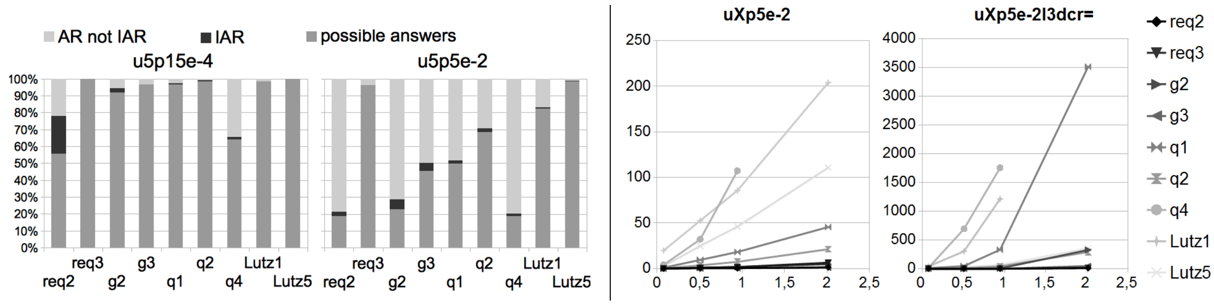

Figure 2: [left] Proportion of time spent by CQAPri to get the possible, IAR, and AR answers on two ABoxes [right] Time (in sec.) spent by CQAPri for AR and $\subseteq_P$-AR query answering on two sets of ABoxes (in millions of assertions)

expressive DL $\mathcal{SHIQ}$. They focus on ground CQs, as such queries are better-supported by $\mathcal{SHIQ}$ reasoners. By contrast, we work with DL-Lite and can thus use query rewriting techniques to handle CQs with existential variables. We also consider IAR-based semantics and other types of preferred repairs that are not considered by Du et al.

Also related is the work on preference-based semantics for querying databases that violate integrity constraints. In (Lopatenko and Bertossi 2007), the authors study the complexity of query answering in the presence of denial constraints under the $\leq$-AR and $\leq_w$-AR semantics. Because of the difference in setting, we could not transfer their complexity results to DL-Lite. Three further preference-based semantics are proposed in (Staworko, Chomicki, and Marcinkowski 2012), based upon partially ordering the assertions that appear together in a conflict. If such an ordering is induced from an ABox prioritization, then the three semantics all coincide with our $\subseteq_P$-AR semantics.

More generally, we note that the problem of reasoning on preferred subsets has been studied in a number of other areas of AI, such as abduction, belief change, argumentation, and non-monotonic reasoning, see (Eiter and Gottlob 1995; Nebel 1998; Amgoud and Vesic 2011; Brewka, Niemelä, and Truszczynski 2008) and references therein.

A recent line of work, including (Rosati 2011; Bienvenu 2012; Bienvenu and Rosati 2013), studies the complexity of query answering under IAR and AR semantics. We extend it by providing complexity results for variants of the IAR and AR semantics based on preferred repairs. In some cases, we were able to adapt proof ideas from the preference-free case.

In terms of implemented tools, we are aware of two systems for inconsistency-tolerant query answering over DL KBs: the system of Du et al. (2013) for querying $\mathcal{SHIQ}$ KBs under $\leq_w$-AR semantics, and the QuID system (Rosati et al. 2012) that handles IAR semantics (without preferences) and DL-Lite$_\mathcal{A}$ KBs. Neither system is directly comparable to our own, since they employ different semantics. We can observe some high-level similarities with Du et al.'s system which also employs SAT solvers and uses a reachability analysis to identify a query-relevant portion of the KB.

There are also a few systems for querying inconsistent relational databases. Most relevant to our work the present paper is EQUIP (Kolaitis, Pema, and Tan 2013), which reduces AR query answering with denial constraints to binary integer programming (BIP). We considered using BIP for our own system, but our early experiments comparing the two approaches revealed better performances of the SAT-based approach on difficult instances.

8 Concluding Remarks

Existing inconsistency-tolerant semantics for ontology-based data access are based upon a notion of repair that makes no assumptions about the relative reliability of ABox assertions. When information on the reliability of assertions is available, it can be exploited to identify preferred repairs and improve the quality of the query results. While this idea has been explored before in various settings, there had been no systematic study of the computational properties of preferred repairs for important lightweight DLs like DL-Lite$_\mathcal{R}$. We addressed this gap in the literature by providing a thorough analysis that established the data and combined complexity of answering conjunctive and atomic queries under AR- and IAR-based semantics combined with four types of preferred repairs. Unsurprisingly, the results are mainly negative, showing that adding preferences increases complexity. However, they also demonstrate that IAR-based semantics retain some advantage over AR-based semantics and identify $\subseteq_P$-repairs as the most computationally well-behaved.

Prior work on inconsistency-tolerant querying in DL-Lite left open whether the IAR constitutes a good approximation, or whether the AR semantics can be feasibly computed in practice. Encouraged by the performance of modern-day SAT solvers and recent positive results from the database arena, we proposed a practical SAT-based approach for query answering in DL-Lite$_\mathcal{R}$ under the AR, $\subseteq_P$-IAR, and $\subseteq_P$-AR semantics, which we implemented in our CQAPri system. Our experiments show that CQAPri scales up to large ABoxes for the IAR/AR and $\subseteq_P$-IAR/$\subseteq_P$-AR semantics, when the number of conflicting assertions varies from a few percent (for all of these semantics) to a few tens of percent (only for IAR/AR). We thus show that the AR semantics *can* be computed in practice and that this is due to the fact the IAR semantics often constitutes a *very good approximation* of the AR semantics. In a similar vein, we observed that our simple approximation of the $\subseteq_P$-IAR semantics often produced a large share of the $\subseteq_P$-IAR answers, which themselves constituted a large portion of the $\subseteq_P$-AR answers.

Our long-term goal is to equip CQAPri with a portfolio of query answering techniques and an optimizer that selects the most appropriate technique for the query at hand. To this end, we have started exploring other techniques for the $\subseteq_P$-based semantics to handle difficult problem instances.

Acknowledgements

The authors acknowledge Despoina Trivela for her work on an early version of the system. This work was supported by the ANR project PAGODA (ANR-12-JS02-007-01).

References

Amgoud, L., and Vesic, S. 2011. A new approach for preference-based argumentation frameworks. *Annals of Mathematics and Artificial Intelligence* 63(2):149–183.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Berre, D. L., and Parrain, A. 2010. The sat4j library, release 2.2. *JSAT* 7(2-3):59–64.

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

Bienvenu, M., and Rosati, R. 2013. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proc. of IJCAI*.

Bienvenu, M.; Bourgaux, C.; and Goasdoué, F. 2014. Querying inconsistent description logic knowledge bases under preferred repair semantics (Long version with appendix). Technical Report 1572, LRI, Orsay, France.

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

Brewka, G.; Niemelä, I.; and Truszczynski, M. 2008. Preferences and nonmonotonic reasoning. *AI Magazine* 29(4):69–78.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general datalog-based framework for tractable query answering over ontologies. *Journal of Web Semantics* 14:57–83.

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.; Marcinkowski, J.; and Staworko, S. 2004. Computing consistent query answers using conflict hypergraphs. In *Proc. of CIKM*, 417–426.

Chortaras, A.; Trivela, D.; and Stamou, G. 2011. Optimized query rewriting for OWL 2 QL. In *Proc. of CADE*.

Du, J.; Qi, G.; and Shen, Y.-D. 2013. Weight-based consistent query answering over inconsistent $\mathcal{SHIQ}$ knowledge bases. *Knowledge and Information Systems* 34(2):335–371.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42(1):3–42.

Eiter, T.; Gottlob, G.; Ortiz, M.; and Simkus, M. 2008. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proc. of JELIA*, 166–179.

Hustadt, U.; Motik, B.; and Sattler, U. 2007. Reasoning in description logics by a reduction to disjunctive datalog. *Journal of Automated Reasoning* 39(3):351–384.

Kolaitis, P. G.; Pema, E.; and Tan, W.-C. 2013. Efficient querying of inconsistent databases with binary integer programming. *PVLDB* 6(6):397–408.

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.

Lopatenko, A., and Bertossi, L. E. 2007. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics. In *Proc. of ICDT*, 179–193.

Lutz, C.; Seylan, I.; Toman, D.; and Wolter, F. 2013. The combined approach to OBDA: Taming role hierarchies using filters. In *Proc. of ISWC*, 314–330.

Motik, B.; Cuenca Grau, B.; Horrocks, I.; Wu, Z.; Fokoue, A.; and Lutz, C. 2012. OWL 2 Web Ontology Language profiles. W3C Recommendation. Available at http://www.w3.org/TR/owl2-profiles/.

Nebel, B. 1998. *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 3: Belief Change*. Kluwer. chapter How Hard is it to Revise a Belief Base?

Pérez-Urbina, H.; Horrocks, I.; and Motik, B. 2009. Efficient query answering for OWL 2. In *Proc. of ISWC*, 489–504.

Rosati, R.; Ruzzi, M.; Graziosi, M.; and Masotti, G. 2012. Evaluation of techniques for inconsistency handling in OWL 2 QL ontologies. In *Proc. of ISWC*, 337–349.

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

Staworko, S.; Chomicki, J.; and Marcinkowski, J. 2012. Prioritized repairing and consistent query answering in relational databases. *Annals of Mathematics and Artificial Intelligence* 64(2-3):209–246.