Explaining Inconsistency-Tolerant Query Answering over Description Logic Knowledge Bases

Meghyn Bienvenu, Camille Bourgaux, François Goasdoué

Explaining Inconsistency-Tolerant Query Answering over Description Logic Knowledge Bases

Meghyn Bienvenu
CNRS, Univ. Montpellier, Inria
Montpellier, France

Camille Bourgaux
Univ. Paris-Sud, CNRS
Orsay, France

François Goasdoué
Univ. Rennes 1, CNRS
Lannion, France

Abstract

Several inconsistency-tolerant semantics have been introduced for querying inconsistent description logic knowledge bases. This paper addresses the problem of explaining why a tuple is a (non-)answer to a query under such semantics. We define explanations for positive and negative answers under the brave, AR and IAR semantics. We then study the computational properties of explanations in the lightweight description logic DL-Lite$_\mathcal{R}$. For each type of explanation, we analyze the data complexity of recognizing (preferred) explanations and deciding if a given assertion is relevant or necessary. We establish tight connections between intractable explanation problems and variants of propositional satisfiability (SAT), enabling us to generate explanations by exploiting solvers for Boolean satisfaction and optimization problems. Finally, we empirically study the efficiency of our explanation framework using the well-established LUBM benchmark.

1 Introduction

Description logic (DL) knowledge bases (KBs) consist of a TBox (ontology) that provides conceptual knowledge about the application domain and an ABox (dataset) that contains facts about particular entities (Baader et al. 2003). The problem of querying such KBs using database-style queries (in particular, conjunctive queries) has been a major focus of recent DL research. Since scalability is a key concern, much of the work has focused on lightweight DLs for which query answering can be performed in polynomial time w.r.t. the size of the ABox. The DL-Lite family of lightweight DLs (Calvanese et al. 2007) is especially popular due to the fact that query answering can be reduced, via query rewriting, to the problem of standard database query evaluation.

Since the TBox is usually developed by experts and subject to extensive debugging, it is often reasonable to assume that its contents are correct. By contrast, the ABox is typically substantially larger and subject to frequent modifications, making errors almost inevitable. As such errors may render the KB inconsistent, several inconsistency-tolerant semantics have been introduced in order to provide meaningful answers to queries posed over inconsistent KBs. Arguably the most well-known is the *AR semantics* (Lembo et al. 2010), inspired by work on consistent query answering in databases (cf. (Bertossi 2011) for a survey). Query answering under AR semantics amounts to considering those answers (w.r.t. standard semantics) that can be obtained from every *repair*, i.e. inclusion-maximal subset of the ABox that is consistent with the TBox. The more cautious *IAR semantics* (Lembo et al. 2010) queries the intersection of the repairs and provides a lower bound on AR semantics. The *brave* semantics (Bienvenu and Rosati 2013), which considers those answers holding in at least one repair, provides a natural upper bound.

The complexity of inconsistency-tolerant query answering in the presence of ontologies is now well understood (see e.g. (Rosati 2011; Bienvenu 2012; Lukasiewicz, Martinez, and Simari 2013)), so attention has turned to the problem of implementing these alternative semantics. There are currently two systems for querying inconsistent DL-Lite KBs: the QuID system of (Rosati et al. 2012) implements the IAR semantics, using either query rewriting or ABox cleaning, and our CQAPri system (2014) implements the AR, IAR and brave semantics, using tractable methods to obtain the answers under IAR and brave semantics and calls to a SAT solver to identify the answers holding under AR semantics.

The need to equip reasoning systems with explanation services is widely acknowledged by the DL community (see Section 6 for discussion and references), and such facilities are all the more essential when using inconsistency-tolerant semantics, as recently argued in (Arioua et al. 2014). Indeed, the brave, AR, and IAR semantics allow one to classify query answers into three categories of increasing reliability, and a user may naturally wonder why a given tuple was assigned to, or excluded from, one of these categories. In this paper, we address this issue by proposing and exploring a framework for explaining query answers under these three semantics. Our contributions are as follows:

• We define explanations of positive and negative query answers under brave, AR and IAR semantics. Intuitively, such explanations pinpoint the portions of the ABox that, in combination with the TBox, suffice to obtain the considered query answer. We focus on ABox assertions since inconsistencies are assumed to stem from errors in the ABox, and because this already yields a non-trivial framework to study.

• We investigate the main search and decision problems related to explanations: generating an (arbitrary) explanation, generating a most preferred explanation according to some natural ranking criteria, recognizing (most preferred) explanations, and checking whether an assertion is relevant / necessary (i.e. appears in some / all explanations). We study the data complexity of these problems for DL-Lite$_\mathcal{R}$, showing (in)tractability of each of the tasks and pinpointing the exact complexity of the intractable decision problems.

• We establish tight connections between the intractable decision problems, as well as the problem of generating (preferred) explanations, and SAT-based reasoning tasks. This enables effective solutions to these problems using solvers for Boolean satisfaction and optimization problems.

• Finally, we present our implementation of our explanation services on top of the CQAPri system and SAT4J solver. Using the LUBM benchmark, we show its practical interest on a large ABox with increasing number of conflicting assertions: explanations of answers are computed rapidly overall, typically in a few milliseconds, rarely above a second.

2 Preliminaries

We briefly recall the syntax and semantics of DLs, and the inconsistency-tolerant semantics we use.

Syntax A DL *knowledge base (KB)* consists of an ABox and a TBox, both 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 $A(a)$ and *role assertions* of the form $R(a, b)$, where $A \in \mathsf{N_C}$, $R \in \mathsf{N_R}$, $a, b \in \mathsf{N_I}$. 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 according to 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 $(\exists Q)^\mathcal{I} = \{c \mid \exists d : (c, d) \in Q^\mathcal{I}\}$. We say that $\mathcal{I}$ 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. $(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$). We call $\mathcal{I}$ a *model* of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ satisfies all axioms in $\mathcal{T}$ and assertions in $\mathcal{A}$. A KB $\mathcal{K}$ is *consistent* if it has a model; otherwise it is inconsistent, denoted $\mathcal{K} \models \bot$. An ABox $\mathcal{A}$ is *$\mathcal{T}$-consistent* if the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent.

Example 1. As a running example, we consider a simple KB $\mathcal{K}_\mathsf{ex} = (\mathcal{T}_\mathsf{ex}, \mathcal{A}_\mathsf{ex})$ about the university domain that contains concepts for postdoctoral researchers (Postdoc), professors (Pr) of two levels of seniority (APr, FPr), and PhD holders (PhD), as well as roles to link advisors to their students (Adv) and instructors to their courses (Teach). The ABox $\mathcal{A}_\mathsf{ex}$ provides information about an individual a:

$$\begin{aligned}\mathcal{T}_\mathsf{ex} =\{&\mathsf{Postdoc} \sqsubseteq \mathsf{PhD}, \mathsf{Pr} \sqsubseteq \mathsf{PhD}, \mathsf{Postdoc} \sqsubseteq \neg\mathsf{Pr},\\ &\mathsf{FPr} \sqsubseteq \mathsf{Pr}, \mathsf{APr} \sqsubseteq \mathsf{Pr}, \mathsf{APr} \sqsubseteq \neg\mathsf{FPr}, \exists\mathsf{Adv} \sqsubseteq \mathsf{Pr}\}\\ \mathcal{A}_\mathsf{ex} =\{&\mathsf{Postdoc}(a), \mathsf{FPr}(a), \mathsf{APr}(a), \mathsf{Adv}(a, b),\\ &\mathsf{Teach}(a, c_1), \mathsf{Teach}(a, c_2), \mathsf{Teach}(a, c_3)\}\end{aligned}$$

Observe that $\mathcal{A}_\mathsf{ex}$ is $\mathcal{T}_\mathsf{ex}$-inconsistent.

Queries We focus 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 $R(t, t')$, t, t' are variables or individuals, and $\vec{y}$ is a tuple of variables from ψ. When we use the generic term *query*, we mean a CQ. Given a CQ q with free variables $x_1, \dots, x_k$ and a tuple of individuals $\vec{a} = (a_1, \dots, a_k)$, we use $q(\vec{a})$ to denote the first-order sentence resulting from replacing each x_i by a_i. A tuple $\vec{a}$ is a *certain answer* to q over $\mathcal{K}$, written $\mathcal{K} \models q(\vec{a})$, iff $q(\vec{a})$ holds in every model of $\mathcal{K}$.

Causes and conflicts A *cause* for $q(\vec{a})$ w.r.t. KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is a minimal $\mathcal{T}$-consistent subset $\mathcal{C} \subseteq \mathcal{A}$ such that $\mathcal{T}, \mathcal{C} \models q(\vec{a})$. We use $\mathsf{causes}(q(\vec{a}), \mathcal{K})$ to refer to the set of causes for $q(\vec{a})$. A *conflict* for $\mathcal{K}$ is a minimal $\mathcal{T}$-inconsistent subset of $\mathcal{A}$, and $\mathsf{confl}(\mathcal{K})$ denotes the set of conflicts for $\mathcal{K}$.

When $\mathcal{K}$ is a DL-Lite$_\mathcal{R}$ KB, every conflict for $\mathcal{K}$ has at most two assertions. We can thus define the set of *conflicts of a set of assertions* $\mathcal{C} \subseteq \mathcal{A}$ as follows:

$$\mathsf{confl}(\mathcal{C}, \mathcal{K}) = \{\beta \mid \exists\alpha \in \mathcal{C}, \{\alpha, \beta\} \in \mathsf{confl}(\mathcal{K})\}.$$

Inconsistency-tolerant semantics A *repair* of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is an inclusion-maximal subset of $\mathcal{A}$ that is $\mathcal{T}$-consistent. We consider three previously studied inconsistency-tolerant semantics based upon repairs. Under *AR semantics*, a tuple $\vec{a}$ is an answer to q over $\mathcal{K}$, written $\mathcal{K} \models_\text{AR} q(\vec{a})$, just in the case that $\mathcal{T}, \mathcal{R} \models q(\vec{a})$ for *every repair* $\mathcal{R}$ of $\mathcal{K}$ (equivalently: every repair contains some cause of $q(\vec{a})$). If there exists *some repair* $\mathcal{R}$ such that $\mathcal{T}, \mathcal{R} \models q(\vec{a})$ (equivalently: $\mathsf{causes}(q(\vec{a}), \mathcal{K}) \neq \emptyset$), then $\vec{a}$ is an answer to q under *brave semantics*, written $\mathcal{K} \models_\text{brave} q(\vec{a})$. For *IAR semantics*, we have $\mathcal{K} \models_\text{IAR} q(\vec{a})$ iff $\mathcal{T}, \mathcal{R}_\cap \models q(\vec{a})$ (equivalently, $\mathcal{R}_\cap$ contains some cause for $q(\vec{a})$), where $\mathcal{R}_\cap$ is the *intersection of all repairs* of $\mathcal{K}$. The three semantics are related as follows:

$$\mathcal{K} \models_\text{IAR} q(\vec{a}) \quad\Rightarrow\quad \mathcal{K} \models_\text{AR} q(\vec{a}) \quad\Rightarrow\quad \mathcal{K} \models_\text{brave} q(\vec{a})$$

For $S \in \{\text{AR}, \text{brave}, \text{IAR}\}$, we call $\vec{a}$ a *(positive) S-answer* (resp. *negative S-answer*) if $\mathcal{K} \models_\text{S} q(\vec{a})$ (resp. $\mathcal{K} \not\models_\text{S} q(\vec{a})$).

Example 2. The example KB $\mathcal{K}_\mathsf{ex}$ has three repairs:

$$\begin{aligned}\mathcal{R}_1 &= \mathcal{A}_\mathsf{ex} \setminus \{\mathsf{FPr}(a), \mathsf{APr}(a), \mathsf{Adv}(a, b)\}\\ \mathcal{R}_2 &= \mathcal{A}_\mathsf{ex} \setminus \{\mathsf{Postdoc}(a), \mathsf{FPr}(a)\}\\ \mathcal{R}_3 &= \mathcal{A}_\mathsf{ex} \setminus \{\mathsf{Postdoc}(a), \mathsf{APr}(a)\}\end{aligned}$$

We consider the following example queries: $q_1 = \mathsf{Prof}(x)$, $q_2 = \exists y\, \mathsf{PhD}(x) \wedge \mathsf{Teach}(x, y)$, and $q_3 = \exists y \mathsf{Teach}(x, y)$. Evaluating these queries on $\mathcal{K}_\mathsf{ex}$ yields the following results:

$$\begin{array}{lll}\mathcal{K}_\mathsf{ex} \models_\text{brave} q_1(a) & \mathcal{K}_\mathsf{ex} \models_\text{AR} q_2(a) & \mathcal{K}_\mathsf{ex} \models_\text{IAR} q_3(a)\\ \mathcal{K}_\mathsf{ex} \not\models_\text{AR} q_1(a) & \mathcal{K}_\mathsf{ex} \not\models_\text{IAR} q_2(a) & \end{array}$$

3 Explaining Query Results

The aforementioned inconsistency-tolerant semantics allows us to identify three types of positive query answer:

$$\text{IAR-answers} \quad\subseteq\quad \text{AR-answers} \quad\subseteq\quad \text{brave-answers}$$

The goal of the present work is to help the user understand the classification of a particular tuple, e.g. why is $\vec{a}$

an AR-answer, and why is it not an IAR-answer? To this end, we introduce the notion of *explanation* for positive and negative query answers under brave, AR, and IAR semantics. For consistent KBs, these three semantics collapse into the classical one, so existing explanation frameworks can be used (Borgida, Calvanese, and Rodriguez-Muro 2008; Calvanese et al. 2013; Du, Wang, and Shen 2014).

Formally, the explanations we consider will take either the form of a set of ABox assertions (viewed as a conjunction) or a set of sets of assertions (interpreted as a disjunction of conjunctions). We chose to focus on ABox assertions, rather than TBox axioms, since we target scenarios in which inconsistencies are due to errors in the ABox, so understanding the link between (possibly faulty) ABox assertions and query results is especially important. Moreover, as we shall see in Sections 4 and 5, our 'ABox-centric' explanation framework already poses non-trivial computational challenges.

The simplest answers to explain are positive brave- and IAR-answers. We can use the query's causes as explanations for the former, and the causes that do not participate in any contradiction for the latter. Note that in what follows we suppose that $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is a KB and q is a query.

Definition 1. An explanation for $\mathcal{K} \models_{\mathsf{brave}} q(\vec{a})$ is a cause for $q(\vec{a})$ w.r.t. $\mathcal{K}$. An explanation for $\mathcal{K} \models_{\mathrm{IAR}} q(\vec{a})$ is a cause $\mathcal{C}$ for $q(\vec{a})$ w.r.t. $\mathcal{K}$ such that $\mathcal{C} \subseteq \mathcal{R}$ for every repair $\mathcal{R}$ of $\mathcal{K}$.

Example 3. There are three explanations for $\mathcal{K}_{\mathsf{ex}} \models_{\mathsf{brave}} q_1(a)$: $\mathsf{FPr}(a)$, $\mathsf{APr}(a)$, and $\mathsf{Adv}(a, b)$. There are twelve explanations for $\mathcal{K}_{\mathsf{ex}} \models_{\mathsf{brave}} q_2(a)$: $\mathsf{Postdoc}(a) \wedge \mathsf{Teach}(a, c_j)$, $\mathsf{FPr}(a) \wedge \mathsf{Teach}(a, c_j)$, $\mathsf{APr}(a) \wedge \mathsf{Teach}(a, c_j)$, and $\mathsf{Adv}(a, b) \wedge \mathsf{Teach}(a, c_j)$, for each $j \in \{1, 2, 3\}$. There are three explanations for $\mathcal{K}_{\mathsf{ex}} \models_{\mathrm{IAR}} q_3(a)$: $\mathsf{Teach}(a, c_1)$, $\mathsf{Teach}(a, c_2)$, and $\mathsf{Teach}(a, c_3)$.

To explain why a tuple is an AR-answer, it is no longer sufficient to give a single cause, since different repairs may use different causes. We will therefore define explanations as (minimal) disjunctions of causes that 'cover' all repairs.

Definition 2. An explanation for $\mathcal{K} \models_{\mathrm{AR}} q(\vec{a})$ is a set $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\} \subseteq \mathsf{causes}(q(\vec{a}), \mathcal{K})$ such that (i) every repair $\mathcal{R}$ of $\mathcal{K}$ contains some $\mathcal{C}_i$, and (ii) no proper subset of $\mathcal{E}$ satisfies this property.

Example 4. There are 36 explanations for $\mathcal{K}_{\mathsf{ex}} \models_{\mathrm{AR}} q_2(a)$, each taking one of the following two forms:

$$
\begin{aligned}
\mathcal{E}_{ij} =& (\mathsf{Postdoc}(a) \wedge \mathsf{Teach}(a, c_i)) \vee (\mathsf{Adv}(a, b) \wedge \mathsf{Teach}(a, c_j)) \\
\mathcal{E}'_{ijk} =& (\mathsf{Postdoc}(a) \wedge \mathsf{Teach}(a, c_i)) \vee (\mathsf{FPr}(a) \wedge \mathsf{Teach}(a, c_j)) \\
& \vee (\mathsf{APr}(a) \wedge \mathsf{Teach}(a, c_k))
\end{aligned}
$$

for some $i, j, k \in \{1, 2, 3\}$.

We next consider how to explain negative AR- and IAR-answers, i.e., brave-answers not entailed under AR or IAR semantics. For AR semantics, the idea is to give a (minimal) subset of the ABox that is consistent with the TBox and contradicts every cause of the query, since any such subset can be extended to a repair that omits all causes. For IAR semantics, the formulation is slightly different as we need only ensure that every cause is contradicted by some consistent subset, as this shows that no cause belongs to all repairs.

Definition 3. An explanation for $\mathcal{K} \not\models_{\mathrm{AR}} q(\vec{a})$ is a $\mathcal{T}$-consistent subset $\mathcal{E} \subseteq \mathcal{A}$ such that: (i) $\mathcal{T}, \mathcal{E} \cup \mathcal{C} \models \bot$ for every $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$, (ii) no proper subset of $\mathcal{E}$ has this property. An explanation for $\mathcal{K} \not\models_{\mathrm{IAR}} q(\vec{a})$ is a (possibly $\mathcal{T}$-inconsistent) subset $\mathcal{E} \subseteq \mathcal{A}$ such that: (i) for every $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$, there exists a $\mathcal{T}$-consistent subset $\mathcal{E}' \subseteq \mathcal{E}$ with $\mathcal{T}, \mathcal{E}' \cup \mathcal{C} \models \bot$, (ii) no $\mathcal{E}' \subsetneq \mathcal{E}$ has this property.

Example 5. The unique explanation for $\mathcal{K}_{\mathsf{ex}} \not\models_{\mathrm{AR}} q_1(a)$ is $\mathsf{Postdoc}(a)$, which contradicts the three causes of $q_1(a)$. The explanations for $\mathcal{K}_{\mathsf{ex}} \not\models_{\mathrm{IAR}} q_2(a)$ are: $\mathsf{FPr}(a) \wedge \mathsf{Postdoc}(a)$, $\mathsf{APr}(a) \wedge \mathsf{Postdoc}(a)$, and $\mathsf{Adv}(a, b) \wedge \mathsf{Postdoc}(a)$, where the first assertion of each explanation contradicts the causes of $q_2(a)$ that contain $\mathsf{Postdoc}(a)$, and the second one contradicts those that contain $\mathsf{FPr}(a)$, $\mathsf{APr}(a)$ or $\mathsf{Adv}(a, b)$.

When there is a large number of explanations for a given answer, it may be impractical to present them all to the user. In such cases, one may choose to rank the explanations according to some preference criteria, and to present one or a small number of most *preferred explanations*. In this work, we will use *cardinality* to rank explanations for brave- and IAR-answers and negative AR- and IAR-answers. For positive AR-answers, we consider two ranking criteria: the *number of disjuncts*, and the total *number of assertions*.

Example 6. Reconsider explanations $\mathcal{E}_{1\,1}$ and $\mathcal{E}'_{1\,2\,3}$ for $\mathcal{K}_{\mathsf{ex}} \models_{\mathrm{AR}} q_2(a)$. There are at least two reasons why $\mathcal{E}_{1\,1}$ may be considered easier to understand than $\mathcal{E}'_{1\,2\,3}$. First, $\mathcal{E}_{1\,1}$ contains fewer disjuncts, hence requires less disjunctive reasoning. Second, both disjuncts of $\mathcal{E}_{1\,1}$ use the same Teach assertion, whereas $\mathcal{E}'_{1\,2\,3}$ uses three different Teach assertions, which may lead the user to (wrongly) believe all are needed to obtain the query result. Preferring explanations having the fewest number of disjuncts, and among them, those involving a minimal set of assertions, leads to focusing on the explanations of the form $\mathcal{E}_{i\,i}$, where $i \in \{1, 2, 3\}$.

A second complementary approach is to concisely summarize the set of explanations in terms of the *necessary assertions* (i.e. appearing in every explanation) and the *relevant assertions* (i.e. appearing in at least one explanation).

Example 7. If we tweak the example KB to include n courses taught by a, then there would be $n^2 + n^3$ explanations for $\mathcal{K}_{\mathsf{ex}} \models_{\mathrm{AR}} q_2(a)$, built using only $n + 4$ assertions. Presenting the necessary assertions (here: $\mathsf{Postdoc}(a)$) and relevant ones ($\mathsf{FPr}(a)$, $\mathsf{APr}(a)$, $\mathsf{Adv}(a, b)$, $\mathsf{Teach}(a, c_i)$) gives a succinct overview of the set of explanations.

4 Complexity Analysis

We next study the computational properties of the different notions of explanation defined in Section 3. In addition to the problem of generating a single explanation (GENONE), or a single best explanation (GENBEST) according a given criteria, we consider four related decision problems: decide whether a given assertion appears in some explanation (REL) or in every explanation (NEC), decide whether a candidate is an explanation (REC), resp. a best explanation (BEST REC).

In the remainder of the paper, we focus on KBs expressed in the lightweight logic DL-Lite$_{\mathcal{R}}$ since it is a popular choice for ontology-based data access and the only DL for which

	brave, IAR	AR	neg. AR	neg. IAR
GENONE	in P	NP-h	NP-h	in P
GENBEST[†]	in P	Σ_2^p-h[‡]	NP-h	NP-h[*]
REL	in P	Σ_2^p-co	NP-co	in P
NEC	in P	NP-co	coNP-co	in P
REC	in P	BH_2-co	in P	in P
BEST REC[†]	in P	Π_2^p-co[‡]	coNP-co[*]	coNP-co[*]

[†] upper bounds hold for ranking criteria that can be decided in P
[‡] lower bounds hold for smallest disjunction or fewest assertions
[*] lower bounds hold for cardinality-minimal explanations

Figure 1: Data complexity results for CQs.

the three considered semantics have been implemented. As we target applications in which the ABox is significantly larger than the TBox and query, we use the *data complexity* measure, which is only with respect to the size of the ABox.

Our complexity results are displayed in Figure 1.

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

In what follows, we present some key ideas underlying Theorem 1 (detailed proofs are provided in (Bienvenu, Bourgaux, and Goasdoué 2016)).

Positive brave- and IAR-answers We recall that in DL-Lite$_\mathcal{R}$, KB satisfiability and query answering are in P w.r.t. data complexity (Calvanese et al. 2007), and conflicts are of size at most two. It follows that the causes and their conflicts can be computed in P w.r.t. data complexity (using e.g. standard query rewriting algorithms). From the causes and conflicts, we can immediately read off the explanations for brave- and IAR-answers; a simple examination of the set of explanations enables us to solve the decision problems.

Positive AR-answers We relate explanations of AR-answers to minimal unsatisfiable subsets of a set of propositional clauses. Let us recall that, given sets F and H of soft and hard clauses respectively, a subset $M \subseteq F$ is a *minimal unsatisfiable subset* (MUS) of F w.r.t. H if (i) $M \cup H$ is unsatisfiable, and (ii) $M' \cup H$ is satisfiable for every $M' \subsetneq M$.

To explain $\mathcal{K} \models_{\text{AR}} q(\vec{a})$, we consider the soft clauses

$$\varphi_{\neg q} = \{\lambda_\mathcal{C} \mid \mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})\} \text{ with } \lambda_\mathcal{C} = \bigvee_{\beta \in \mathsf{confl}(\mathcal{C},\mathcal{K})} x_\beta$$

and the hard clauses

$$\varphi_{cons} = \{\neg x_\alpha \vee \neg x_\beta \mid x_\alpha, x_\beta \in \mathsf{vars}(\varphi_{\neg q}), \{\alpha, \beta\} \in \mathsf{confl}(\mathcal{K})\}$$

It was proven by Bienvenu et al. (2014) that $\mathcal{K} \models_{\text{AR}} q(\vec{a})$ iff $\varphi_{\neg q} \cup \varphi_{cons}$ is unsatisfiable, and we can further show:

Proposition 1. *A set $\mathcal{E} \subseteq \mathsf{causes}(q(\vec{a}), \mathcal{K})$ is an explanation for $\mathcal{K} \models_{AR} q(\vec{a})$ iff $\{\lambda_\mathcal{C} \mid \mathcal{C} \in \mathcal{E}\}$ is a MUS of $\varphi_{\neg q}$ w.r.t. φ_{cons}.*

In addition to enabling us to exploit MUS algorithms for explanation generation, Proposition 1 can be combined with known complexity results for MUSes (Liberatore 2005) to infer the upper bounds for REL, REC, and BEST REC. Moreover, we can find a TBox $\mathcal{T}^*$ and assertion α^* such that for every unsatisfiable clause set φ, we can construct in polytime an ABox A_φ such that the explanations for $(\mathcal{T}^*, \mathcal{A}_\varphi) \models_{\text{AR}} \alpha^*$ correspond (in a precise sense) to the MUSes of φ (w.r.t. $\emptyset$). This reduction enables us to transfer complexity lower bounds for MUSes to our setting.

Negative AR-answers We relate explanations of negative AR-answers to minimal models of $\varphi_{\neg q} \cup \varphi_{cons}$. Given a clause set ψ over variables X, a set $M \subseteq X$ is a *minimal model* of ψ iff (i) every valuation that assigns true to all variables in M satisfies ψ, (ii) no $M' \subsetneq M$ satisfies this condition. Cardinality-minimal models are defined analogously.

Proposition 2. *A set $\mathcal{E}$ is an explanation (resp. cardinality-minimal explanation) for $\mathcal{K} \not\models_{AR} q(\vec{a})$ iff $\{x_\alpha \mid \alpha \in \mathcal{E}\}$ is a minimal (resp. cardinality-minimal) model of $\varphi_{\neg q} \cup \varphi_{cons}$.*

The preceding result enables us to generate explanations using tools for computing (cardinality-)minimal models; it also yields a coNP upper bound for NEC since α belongs to all explanations just in the case that $\varphi_{\neg q} \cup \varphi_{cons} \cup \{\neg x_\alpha\}$ is unsatisfiable. Recognizing an explanation $\mathcal{E}$ can be done in P by checking consistency of $\mathcal{E}$, inconsistency of $\mathcal{E} \cup \mathcal{C}$ for every cause $\mathcal{C}$, and minimality of $\mathcal{E}$. It follows that REL is in NP and BEST REC in coNP for ranking criteria that can be decided in P (guess an explanation that contains the assertion, resp. is a better explanation). The NP and coNP lower bounds are proved by reductions from (UN)SAT.

Negative IAR-answers We relate explanations of negative IAR-answers to minimal models of the clause set $\varphi_{\neg q}$.

Proposition 3. *A set $\mathcal{E}$ is an explanation (resp. cardinality-minimal explanation) for $\mathcal{K} \not\models_{IAR} q(\vec{a})$ iff $\{x_\alpha \mid \alpha \in \mathcal{E}\}$ is a minimal (resp. cardinality-minimal) model of $\varphi_{\neg q}$.*

Importantly, $\varphi_{\neg q}$ does not contain any negative literals, and it is known that for *positive* clause sets, a single minimal model can be computed in P, and the associated relevance problem is also in P. The intractability results for GENBEST and BEST REC are obtained by relating cardinality-minimal models of monotone 2CNF formulas to cardinality-minimal explanations of negative IAR-answers.

5 Prototype and experiments

We implemented our explanation framework in Java within our CQAPri system (`www.lri.fr/~bourgaux/CQAPri`) that supports querying of DL-Lite$_\mathcal{R}$ KBs using several inconsistency-tolerant semantics, including brave, AR and IAR. We used the SAT4J v2.3.4 SAT solver (`www.sat4j.org`) to compute MUSes and cardinality-minimal models (Berre and Parrain 2010).

CQAPri classifies a query answer $\vec{a}$ into one of 3 classes:

- **Possible**: $\mathcal{K} \models_{\text{brave}} q(\vec{a})$ and $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$
- **Likely**: $\mathcal{K} \models_{\text{AR}} q(\vec{a})$ and $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$
- **(Almost) sure**: $\mathcal{K} \models_{\text{IAR}} q(\vec{a})$

Explaining the answer $\vec{a}$ consists in providing *all* the explanations for $\vec{a}$ being a positive answer under the first semantics and a *single* explanation for it being a negative answer under the other one (i.e., a counter-example), together with the necessary and relevant assertions. Positive explanations

Query id	shape	#atoms	#variables	#rewritings
g2	atomic	1	1	44
g3	atomic	1	1	44
q1	dag	5	2	6401
q2	tree	3	2	450
q3	tree	2	3	155
q4	dag	6	4	202579

Table 1: Queries in term of shape, number of atoms, number of variables, and number of CQs in the UCQ-rewriting.

	nb ans	< 10	$[10, 100[$	$[100, 1000[$	> 1000
g2 Sure	29074	100	0	0	0
Likely	1166	86.2	0	6.4	7.4
Poss.	18162	52.2	2.7	39	6.1
g3 Sure	129083	99.999	0.001	0	0
Likely	8902	99.98	0.02	0	0
Poss.	19737	98.58	0.05	1.36	0.01
q1 Likely	10	50	50	0	0
q2 Poss.	133	100	0	0	0
q3 Poss.	208752	99.578	0.421	0	0.001
q4 Sure	128616	99.99	0.01	0	0
Likely	192	99.5	0.5	0	0
Poss.	64820	87.055	12.942	0	0.003

Table 2: Number of answers of each class and distribution of explanation times (in ms) per query on c29.

are ranked as explained in Section 3; for ranking positive-AR answers, the user can choose the priority between the numbers of disjuncts and the total number of assertions.

Explanations are computed using the results on positive and negative answers in Section 4. We thus need the *causes* of the query answers as well as their *conflicts*. The conflicts are directly available from CQAPri; for the causes, CQAPri uses query rewriting to identify consistent (but not necessarily minimal) subsets of the ABox entailing the answers, from which we must prune non-minimal ones.

To assess the practical interest of our framework, we empirically study the properties of our implementation, in particular: the impact of varying the percentage of assertions in conflict, the typical number and size of explanations, and the extra effort required to generate cardinality-minimal explanations for negative IAR-answers rather than arbitrary ones.

Experimental setting We used the CQAPri benchmark available at www.lri.fr/~bourgaux/CQAPri, which builds on the DL-Lite$_\mathcal{R}$ version (Lutz et al. 2013) of the Lehigh University Benchmark (swat.cse.lehigh.edu/projects/lubm). It extends the DL-Lite$_\mathcal{R}$ TBox with negative inclusions and describes how to obtain an ABox with a natural repartition of conflicts by adding assertions to an initial ABox consistent with the enriched TBox.

We used a consistent database with 100 universities (more than 10 million assertions) from which we generated seven inconsistent ABoxes with different ratios of assertions in conflicts by adding from 8005 to 351724 assertions. These ABoxes are denoted cX, with X the ratio of conflicts varying from 5% to a value of 44% challenging our approach. Also, the way we generate conflicts ensures $\mathtt{cX} \subseteq \mathtt{cY}$ if $\mathtt{X} \leq \mathtt{Y}$.

The queries can be found on CQAPri website. Table 1 displays the characteristics of these queries, which have (i) a variety of structural aspects and number of rewritings, and (ii) answers in the three considered classes (see Table 2).

Our hardware is an Intel Xeon X5647 at 2.93 GHz with 16 GB of RAM, running CentOS 6.7. Reported times are averaged over 5 runs.

Experimental results We summarize below the general tendencies we observed. Table 2 shows the number of answers from each class for each query, as well as the distribution of the explanation times for these answers. Figure 2 shows the proportion of time spent in the different phases and the total time to explain *all* query answers over ABoxes of increasing difficulty. The explanation cost, given by the upper bar, consists in pruning non-minimal consistent subsets of the ABox entailing the answers to get the causes, and computing the explanations from the causes and conflicts. The two lower bars relate to the query answering phase, which consists in rewriting and executing the query (execute), and identifying Sure, Likely, and Possible answers (classify).

The main conclusion is that explaining a *single* query answer, as described above, is always feasible and fast ($\leq$1s) when there are a *few* percent of conflicts in the ABox (c5 case), as is likely to be the case in most real applications. Even with a *significant* percentage of conflicts (c29 case, Table 2), the longest time observed is below 8s. In *all* the experiments we made, explaining a *single* answer typically takes less than 10ms, rarely more than 1s. (Computing explanations of *all* answers can be prohibitively expensive when there are very many answers (Figure 2, left), which is why we do not produce them all by default.)

In more detail, adding conflicts to the ABox complicates the explanations of answers, due to their shift from the Sure to the Likely and Possible classes, as Table 3 shows. Explaining such answers indeed comes at higher computational cost. Figure 2 illustrates this phenomenon. The general trend is exemplified with q3: adding more conflicts causes the difficulty of explaining to grow more rapidly.

For negative IAR-answers, we compared the generation of explanations using a polynomial procedure and of *smallest* explanations using the SAT solver. The sizes of arbitrary explanations are generally very close to those of smallest explanations (at most two extra assertions on c29) and the time saved may be important. Arbitrary explanations are generated in less than 10ms, whereas for g2 on c29, $\sim$50min is spent in computing a smallest explanation due to its unusual size (18 assertions, whereas other negative explanations consist in a few assertions).

Finally, we observed that the average number of explanations per answer is often reasonably low, although some answers have a large number of explanations (e.g., on c29, often less than 10 on average, but for g2, 686 for an IAR-answer, 4210 for an AR-answer, and 740 for a brave-answer). Regarding the size of explanations of AR-answers, the number of causes in the disjunction was up to 21, showing the practical interest of ranking the explanations.

	g2			g3			q1			q2			q3			q4		
	c5	c29	c44	c5	c29	c44	c5	c29	c44	c5	c29	c44	c5	c29	c44	c5	c29	c44
Sure	95	60	38	99	82	61	100	0	0	29	0	0	27	0	0	84.85	66.4	50.1
Likely	2	2.4	4	0.2	5.5	12	0	100	0	0	0	0	0	0	0	0.06	0.1	0.6
Possible	3	37.6	58	0.8	12.5	27	0	0	100	71	100	100	73	100	100	15.09	33.5	49.3

Table 3: Distribution of answers in the different classes on ABoxes with 5%, 29%, and 44% of assertions involved in a conflict.

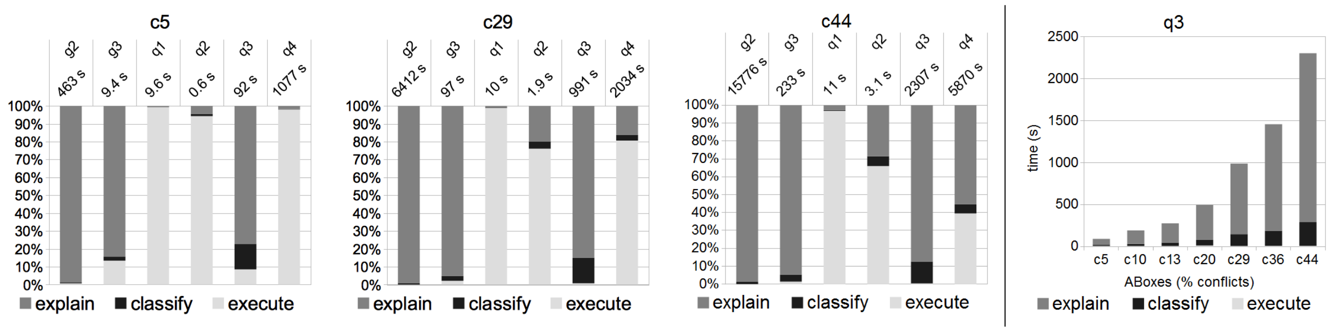

Figure 2: Impact of conflicts [left] Proportion of time spent in the different phases and total time (in sec.) to explain all query answers on c5, c29, and c44. [right] Time spent for explaining all answers of q3 on ABoxes with growing ratio of conflicts.

6 Related Work on Explanations

As mentioned in Section 1, there has been significant interest in equipping DL reasoning systems with explanation facilities. The earliest work proposed formal proof systems as a basis for explaining concept subsumptions (McGuinness and Borgida 1995; Borgida, Franconi, and Horrocks 2000), while The post-2000 literature mainly focuses on *axiom pinpointing* (Schlobach and Cornet 2003; Kalyanpur et al. 2005; Horridge, Parsia, and Sattler 2012), in which the problem is to generate minimal subsets of the KB that yield a given (surprising or undesirable) consequence; such subsets are often called *justifications*. For the lightweight DL $\mathcal{EL}+$, justifications correspond to minimal models of propositional Horn formulas and can be computed using SAT solvers (Sebastiani and Vescovi 2009); a polynomial algorithm has been proposed to compute one justification in (Baader, Peñaloza, and Suntisrivaraporn 2007). In DL-Lite, the problem is simpler: all justifications can be enumerated in polynomial delay (Peñaloza and Sertkaya 2010).

It should be noted that work on axiom pinpointing has thus far focused on explaining entailed TBox axioms (or possibly ABox assertions), but not answers to conjunctive queries. The latter problem is considered in (Borgida, Calvanese, and Rodriguez-Muro 2008), which introduces a proof-theoretic approach to explaining positive answers to CQs over DL-Lite$_\mathcal{A}$ KBs. The approach outputs a single proof, involving both TBox axioms and ABox assertions, using minimality criteria to select a 'simplest' proof.

More recently, the problem of explaining negative query answers over DL-Lite$_\mathcal{A}$ KBs has been studied (Calvanese et al. 2013). Formally, the explanations for $\mathcal{T}, \mathcal{A} \not\models q(\vec{a})$ correspond to sets $\mathcal{A}'$ of ABox assertions such that $\mathcal{T}, \mathcal{A} \cup \mathcal{A}' \models q(\vec{a})$. Practical algorithms and an implementation for computing such explanations were described in (Du, Wang, and Shen 2014). The latter work was recently extended to the case of inconsistent KBs (Du, Wang, and Shen 2015) . Essentially the idea is to add a set of ABox assertions that will lead to the answer holding under IAR semantics (in particular, the new assertions must not introduce any inconsistencies). By contrast, in our setting, negative query answers result not from the absence of supporting facts, but rather the presence of conflicting assertions. This is why our explanations are composed of assertions from the original ABox.

Probably the closest related work is (Arioua, Tamani, and Croitoru 2015) which introduces an argumentation framework for explaining positive and negative answers under the inconsistency-tolerant semantics ICR (Bienvenu 2012). Their motivations are quite similar to our own, and there are some high-level similarities in the definition of explanations (e.g., to explain positive ICR-answers, they consider sets of arguments that minimally cover the preferred extensions, whereas for positive AR-answers, we use sets of causes that minimally cover the repairs). Our work differs from theirs by considering different semantics and by providing a detailed complexity analysis and implemented prototype.

Finally, we note that the problem of explaining query results has been studied in the database community (Cheney, Chiticariu, and Tan 2009; Herschel and Hernández 2010).

7 Conclusion and Future Work

We devised a framework for explaining query (non-)answers over DL KBs under three well-established inconsistency-tolerant semantics (brave, AR, IAR). We then studied the computational properties of our framework, focusing on DL-Lite$_\mathcal{R}$ that underpins W3C's OWL2 QL (Motik et al. 2012). For intractable explanation tasks, we exhibited tight connections with variants of propositional satisfiability, enabling us to implement a prototype using modern SAT solvers. Our experiments showed its practical interest: explanations of query (non-)answers are generated fast overall, i.e., for realistic to challenging ratios of conflicting assertions.

There are several natural directions for future work. First, we plan to accompany our explanations with details on the TBox reasoning involved, using the work of (Borgida, Calvanese, and Rodriguez-Muro 2008) on proofs of positive answers as a starting point. The difficulty of such proofs could provide an additional criteria for ranking explanations (cf. the work on the cognitive complexity of justifications (Horridge et al. 2011)). Second, our experiments showed that an answer can have a huge number of explanations, many of which are quite similar in structure. We thus plan to investigate ways of improving the presentation of explanations, e.g. by identifying and grouping similar explanations (cf. (Bail, Parsia, and Sattler 2013) on comparing justifications), or by defining a notion of representative explanation as in (Du, Wang, and Shen 2014). Third, we plan to experiment with other methods of generating explanations of negative answers, by comparing alternative encodings and using tools for computing hitting sets or diagnoses. Finally, it would be interesting to explore how explanations can be used to partially repair the data based upon the user's feedback.

Acknowledgements

This work was supported by ANR-12-JS02-007-01.

References

Arioua, A.; Tamani, N.; Croitoru, M.; and Buche, P. 2014. Query failure explanation in inconsistent knowledge bases using argumentation. In *Proc. of COMMA*.

Arioua, A.; Tamani, N.; and Croitoru, M. 2015. Query answering explanation in inconsistent datalog +/- knowledge bases. In *Proc. of DEXA*.

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.

Baader, F.; Peñaloza, R.; and Suntisrivaraporn, B. 2007. Pinpointing in the description logic EL^+. In *Proc. of KI*.

Bail, S.; Parsia, B.; and Sattler, U. 2013. The logical diversity of explanations in OWL ontologies. In *Proc. of CIKM*.

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. In *Proc. of AAAI*.

Bienvenu, M.; Bourgaux, C.; and Goasdoué, F. 2016. Explaining inconsistency-tolerant query answering over description logic knowledge bases (long version with appendix). Available at `https://www...`

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

Borgida, A.; Calvanese, D.; and Rodriguez-Muro, M. 2008. Explanation in the DL-Lite family of description logics. In *Proc. of OTM*.

Borgida, A.; Franconi, E.; and Horrocks, I. 2000. Explaining ALC subsumption. In *Proc. of ECAI*.

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. *J. Autom. Reasoning (JAR)* 39(3):385–429.

Calvanese, D.; Ortiz, M.; Simkus, M.; and Stefanoni, G. 2013. Reasoning about explanations for negative query answers in DL-Lite. *J. Artif. Intell. Res. (JAIR)* 48:635–669.

Cheney, J.; Chiticariu, L.; and Tan, W. C. 2009. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases* 1(4):379–474.

Du, J.; Wang, K.; and Shen, Y. 2014. A tractable approach to abox abduction over description logic ontologies. In *Proc. of AAAI*.

Du, J.; Wang, K.; and Shen, Y. 2015. Towards tractable and practical abox abduction over inconsistent description logic ontologies. In *Proc. of AAAI*.

Herschel, M., and Hernández, M. A. 2010. Explaining missing answers to SPJUA queries. *PVLDB* 3(1):185–196.

Horridge, M.; Bail, S.; Parsia, B.; and Sattler, U. 2011. The cognitive complexity of OWL justifications. In *Proc. of ISWC*.

Horridge, M.; Parsia, B.; and Sattler, U. 2012. Extracting justifications from bioportal ontologies. In *Proc. of ISWC*.

Kalyanpur, A.; Parsia, B.; Sirin, E.; and Hendler, J. A. 2005. Debugging unsatisfiable classes in OWL ontologies. *J. Web Sem.* 3(4):268–293.

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

Liberatore, P. 2005. Redundancy in logic I: Cnf propositional formulae. *Artif. Intell. (AIJ)*.

Lukasiewicz, T.; Martinez, M. V.; and Simari, G. I. 2013. Complexity of inconsistency-tolerant query answering in datalog+/-. In *Proc. of OTM*, 488–500.

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

McGuinness, D. L., and Borgida, A. 1995. Explaining subsumption in description logics. In *Proc. of IJCAI*.

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/`.

Peñaloza, R., and Sertkaya, B. 2010. Complexity of axiom pinpointing in the dl-lite family of description logics. In *Proc. of ECAI*.

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*.

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

Schlobach, S., and Cornet, R. 2003. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. of IJCAI*.

Sebastiani, R., and Vescovi, M. 2009. Axiom pinpointing in lightweight description logics via horn-sat encoding and conflict analysis. In *Proc. of CADE*.