

Inconsistency-tolerant Instance Checking in Tractable Description Logics

Rafael Peñaloza

KRDB Research Centre, Free University of Bozen-Bolzano, Italy
(penaloza@inf.unibz.it)

Abstract. Research on inconsistency-tolerant query answering usually assumes that the terminological knowledge is correct, and only the facts (ABox) need to be repaired. In this paper we study the problem of answering instance queries over inconsistent ontologies, by repairing the whole knowledge base (KB). Contrary to ABox repairs, when KB repairs are considered, instance checking in $DL\text{-}Lite_{Horn}$ w.r.t. the brave semantics remains tractable, and the intersection semantics allow for an any-time algorithm. We also show that inconsistency-tolerant instance checking w.r.t. ABox repairs is intractable even if only polynomially many ABox repairs exist.

1 Introduction

Inconsistent-tolerant reasoning arose as a means of obtaining meaningful information from an inconsistent knowledge base (KB). The main idea behind this reasoning task is to consider the different ways in which the KB can be repaired to avoid the inconsistency—or, more generally, any other observed error. The most natural approach is to consider only consequences that follow from *all* such repairs. This approach, which is at the root of the original consistent query answering in databases [1], provides many logical guarantees. Most importantly, it only yields answers that are certain in the sense that they will remain the same, regardless of how the inconsistency is removed. However, providing these answers tends to be computationally hard. In order to regain tractability, several different semantics have been proposed [8, 20]. Two notable examples are the *brave* semantics that require the consequence to follow from at least one repair, and the *intersection* semantics that limit reasoning to the intersection of all repairs.

The computational complexity of dealing with all these semantics has been thoroughly studied for a wide variety of description logics, which have the capability of expressing terminological knowledge along with facts [6, 24, 35]. A running assumption in all this previous work is that the terminological knowledge is always correct, and hence repairs are only made on the data (ABox repairs). However, this assumption is not always valid, since ontology engineering is a very error-prone task.

In this work, we consider the problem of inconsistent-tolerant query answering where repairs allow also parts of the terminological knowledge to be removed,

which we call KB repairs. To the best of our knowledge, the only work where a similar problem has been considered before is [12]. In that work, they consider the problem of inconsistency-tolerant query answering over KB repairs, in a different logic. To guide our complexity analysis, we focus on instance checking in the light-weight description logic $DL-Lite_{Horn}$. This choice is motivated by $DL-Lite_{Horn}$ being the largest member of the $DL-Lite$ family where instance checking remains tractable (w.r.t. so-called KB complexity). For the scope of this paper, we focus only on instance checking as a first step towards dealing with conjunctive queries. Moreover, we do not analyse the data complexity since the problem depends strongly on the input terminological knowledge.

We show that under KB repairs, brave instance checking remains tractable. Although cautious and IR semantics become $coNP$ -complete, we provide an anytime algorithm for the latter, based on methods for enumerating minimal inconsistent subKBs. Afterwards, we revisit the case of ABox repairs. We show that hardness of this problem goes deeper than previously implied, as brave and cautious instance checking cannot be solved in polynomial time even if only polynomially many ABox repairs exist.

2 Preliminaries

We start by introducing the description logic $DL-Lite_{Horn}$ [4], along with the main reasoning problems that we consider in this paper. The importance of looking at $DL-Lite_{Horn}$ is that it is one of the largest members of the $DL-Lite$ family that remains tractable w.r.t. combined complexity.

Let N_C , N_R , and N_I be mutually disjoint sets of *concept*, *role*, and *individual* names, respectively. The classes of $DL-Lite$ concepts B , and roles R are defined, respectively, by the syntax rules:

$$\begin{aligned} B &::= A \mid \exists R \mid \perp \\ R &::= P \mid P^- \end{aligned}$$

where $A \in N_C$ and $P \in N_R$. A $DL-Lite_{Horn}$ TBox is a finite set of *concept inclusions* (CIs) of the form $B_1 \sqcap \dots \sqcap B_n \sqsubseteq B$, $n \geq 1$, where each B, B_i is a concept, and *role inclusions* (RIs) $R_1 \sqsubseteq R_2$, where R_1, R_2 are roles. An ABox is a finite set of *assertions* of the form $A(a)$ or $P(a, b)$ with $A \in N_C, P \in N_R$, and $a, b \in N_I$. A *knowledge base* (KB) is a pair $(\mathcal{T}, \mathcal{A})$, where \mathcal{T} is a TBox and \mathcal{A} an ABox.

The semantics of this logic is based on interpretations. An *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain* and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps every $a \in N_I$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every $A \in N_C$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and every $P \in N_R$ to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This interpretation function is extended to concepts and roles by setting $(P^-)^{\mathcal{I}} := \{(y, x) \mid (x, y) \in P^{\mathcal{I}}\}$, $\exists R^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} \mid \exists y. (x, y) \in R^{\mathcal{I}}\}$, and $\perp^{\mathcal{I}} := \emptyset$. The interpretation \mathcal{I} *satisfies* the CI $B_1 \sqcap \dots \sqcap B_n \sqsubseteq B$ if $\bigcap_{i=1}^n B_i^{\mathcal{I}} \subseteq B^{\mathcal{I}}$. It *satisfies* the assertions $A(a)$ and $P(a, b)$ if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$, respectively. This interpretation is a *model* of the TBox \mathcal{T} (ABox \mathcal{A} , respectively), if

it satisfies all the CIs in \mathcal{T} (all the assertions in \mathcal{A} , respectively). It is a *model* of the KB $(\mathcal{T}, \mathcal{A})$ if it is a model of both, \mathcal{T} and \mathcal{A} .

To simplify the notation, throughout this paper we will often speak of KBs as finite sets of CIs and assertions, unless the distinction between ABox and TBox is relevant. Set operations between KBs are defined in the obvious manner, by operating over each component. For example, if $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$, then $\mathcal{K} \cap \mathcal{K}' := (\mathcal{T} \cap \mathcal{T}', \mathcal{A} \cap \mathcal{A}')$ and $\mathcal{K} \cup \mathcal{K}' := (\mathcal{T} \cup \mathcal{T}', \mathcal{A} \cup \mathcal{A}')$.

Two of the most basic reasoning problems in description logics are deciding consistency of a KB and instance checking. A KB is *consistent* if it has a model. The individual name a is an *instance* of the concept B w.r.t. the KB \mathcal{K} (denoted by $\mathcal{K} \models B(a)$) if $a^{\mathcal{I}} \in B^{\mathcal{I}}$ holds for all models \mathcal{I} of \mathcal{K} . It is known that deciding consistency and instance checking in *DL-Lite_{Horn}* is polynomial on the size of the KB, which is usually known as the *KB complexity* in the query answering literature.

When a KB is inconsistent, its lack of models means that all logical consequences follow trivially. In particular, every individual is an instance of every concept w.r.t. an inconsistent KB. Thus, instance checking is uninformative in this case. Inconsistency-tolerant semantics have been introduced as a way to extract meaningful information from inconsistent KBs by considering some or all the ways in which the inconsistency can be avoided through a repair.

Definition 1. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an inconsistent KB. A KB repair of \mathcal{K} is a consistent KB $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ with $\mathcal{T}' \subseteq \mathcal{T}$, $\mathcal{A}' \subseteq \mathcal{A}$ such that for all $\mathcal{T}' \subseteq \mathcal{T}'' \subseteq \mathcal{T}$ and all $\mathcal{A}' \subseteq \mathcal{A}'' \subseteq \mathcal{A}$ it holds that $(\mathcal{T}', \mathcal{A}'')$ and $(\mathcal{T}'', \mathcal{A}')$ are inconsistent.

An ABox repair of \mathcal{K} is a maximal (w.r.t. set inclusion) subset $\mathcal{A}' \subseteq \mathcal{A}$ such that $(\mathcal{T}, \mathcal{A}')$ is consistent.

Intuitively, a KB repair is obtained by removing the least amount of axioms from a KB to regain consistency. An ABox repair is similar, but assuming that the TBox is correct, thus removing only assertions from the ABox. This latter notion is the one most typically used in the literature of inconsistency-tolerant reasoning. For the rest of this paper, we will use the generic term *repair* to encompass both, KB- and ABox repairs.

It is easy to see that repairs are not unique in general. In fact, a single KB may contain exponentially many repairs (see e.g. [22]). We denote by $\text{Rep}_{\mathcal{K}}$ the set of all repairs of \mathcal{K} . Depending on how these repairs are used, different inconsistency-tolerant semantics can be defined. We present the three most commonly studied, focusing on the case of instance checking.

Definition 2. Let \mathcal{K} be an inconsistent KB, $a \in N_I$ and B a concept. We say that a is a:

- cautious instance of B ($\mathcal{K} \models_c B(a)$) if $\mathcal{R} \models B(a)$ holds for every $\mathcal{R} \in \text{Rep}_{\mathcal{K}}$;
- brave instance of B ($\mathcal{K} \models_b B(a)$) if $\mathcal{R} \models B(a)$ holds for some $\mathcal{R} \in \text{Rep}_{\mathcal{K}}$;
- intersection repair (IR) instance of B ($\mathcal{K} \models_{IR} B(a)$) if $\bigcap_{\mathcal{R} \in \text{Rep}_{\mathcal{K}}} \mathcal{R} \models B(a)$ holds.

When it is relevant to distinguish the class of repairs under consideration, we will denote it through a superscript on the entailment relation. Thus, $\mathcal{K} \models_c^{KB} B(a)$ refers to cautious entailment w.r.t. the set of all KB repairs, while for $\mathcal{K} \models_b^A B(a)$ the inference is made w.r.t. ABox repairs only.

Notice that every IR instance is a cautious instance, and cautious instances themselves are always brave, as long as the input KB is inconsistent. Interest on IR semantics has arisen since they have been shown to provide some tractability guarantees for different scenarios.

The dual notion of a repair is that of a *MinA*—also known as *justification* [16, 18, 28] or MIPS [33, 34]—which is a minimal inconsistent sub-KB of \mathcal{K} . We will denote the set of all MinAs as $\text{Min}_{\mathcal{K}}$. It is well known that repairs and MinAs are dual in the sense that the set of all MinAs can be obtained from all the justifications, and *vice versa*, through the complementation of all Hitting Sets [5, 31].¹ A simple consequence of this duality is that the union of MinAs complements the intersection of repairs; that is, $\bigcup_{\mathcal{M} \in \text{Min}_{\mathcal{K}}} \mathcal{M} = \mathcal{K} \setminus \bigcap_{\mathcal{R} \in \text{Rep}_{\mathcal{K}}} \mathcal{R}$. Notice that although the computation of Hitting Sets is an important problem with applications in many fields of computer science [15], its precise complexity remains unknown. It has been shown that this problem can be solved in a time bound below $n^{o(\log n)}$, which implies that it is most likely not coNP -hard [14]. It is conjectured that, together with several computationally equivalent problems, it forms a class properly contained between P and coNP [13].

An important property that will be useful in the following sections is that if \mathcal{K} is an inconsistent KB, then every consistent sub-KB of \mathcal{K} can be extended to a repair, by adding axioms that do not affect its consistency. Analogously, every inconsistent sub-KB of \mathcal{K} can be reduced to a MinA, by removing all superfluous axioms. Thus, for instance, brave instance checking is equivalent to deciding the existence of a consistent sub-KB \mathcal{R} (not necessarily maximal w.r.t. set inclusion) that entails the desired instance relation.

Finally, to simplify the presentation of the paper, we assume w.l.o.g. that the concept \perp appears only in CIs of the form $B_1 \sqcap \dots \sqcap B_n \sqsubseteq \perp$, where for every $i, 1 \leq i \leq n, B_i \neq \perp$. Notice that a CI that contains \perp on its left-hand side is trivially satisfied by every interpretation, and would hence appear in every repair, and in no MinA.

3 Instance Checking Under KB Repairs

In this section, we study the case where KB repairs are taken into account. The case of ABox repairs will be the focus of the following section. As the result depends on the whole KB, including the TBox, it does not make much sense to consider the data complexity, in which only the size of the ABox is taken into account. Thus, in the following we focus on the KB complexity only, without mentioning it explicitly in every instance. In previous work [8] it has been shown that inconsistency-tolerant instance checking in *DL-Lite_{Horn}* w.r.t.

¹ A *hitting set* for e.g. $\text{Rep}_{\mathcal{K}}$ is a set \mathcal{S} that satisfies $\mathcal{S} \cap \mathcal{R} \neq \emptyset$ for every $\mathcal{R} \in \text{Rep}_{\mathcal{K}}$.

ABox repairs is hard in general, even if one focuses on the simpler brave and intersection semantics. We show that the same does not hold if KB repairs are taken into account. Indeed, as we show next, brave instance checking remains tractable in this case.

Theorem 3. *Brave instance checking w.r.t. KB repairs can be decided in polynomial time.*

Proof. Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and an individual $a \in N_I$, let \mathcal{A}_a be the sub-ABox of \mathcal{A} that contains only the assertions that refer to a ; that is, \mathcal{A}_a contains all axioms $A(a), P(a, b)$, and $P(b, a)$ appearing in \mathcal{A} with $A \in N_C, P \in N_R$, and $b \in N_I$. Define $\mathcal{T}_\top := \mathcal{T} \setminus \{B_1 \sqcap \dots \sqcap B_n \sqsubseteq \perp \in \mathcal{T}\}$, to be the sub-TBox of \mathcal{T} that does not use \perp . Clearly, a is a brave instance of the concept B iff $(\mathcal{T}_\top, \mathcal{A}_a) \models B(a)$. Since (standard) instance checking in *DL-Lite_{Horn}* is polynomial, we obtain the desired upper bound. \square

Unfortunately, tractability does not extend to the other two semantics. As we show next, hardness for cautious and IR instance checking is a consequence of the intractability of finding simple paths in a graph that cross through a given edge.

Theorem 4. *Cautious and IR instance checking are coNP-complete.*

Proof. The upper bounds are obvious, so we focus only on showing hardness through a reduction from the following NP-hard problem: given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, two nodes $v, v' \in \mathcal{V}$ and an edge $e \in \mathcal{E}$, decide whether there is a simple path from v to v' in \mathcal{G} that passes through e .

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $v, v' \in \mathcal{V}$, and $e = (u, u') \in \mathcal{E}$, be an instance of this decision problem. We assume w.l.o.g. that the edge (v, v') does not appear in \mathcal{E} , and that u, u' are both different from v and v' . These cases can be dealt with easily. For every $w \in \mathcal{V} \setminus \{v, v'\}$ create a concept name B_w , and additionally create two individual names a, b . Then, we construct the *DL-Lite_{Horn}* KB $\mathcal{K}_{\mathcal{G}} = (\mathcal{T}, \mathcal{A})$, where

$$\begin{aligned} \mathcal{T} &:= \{B_w \sqsubseteq B_{w'} \mid (w, w') \in \mathcal{E}, v, v' \notin \{w, w'\}\} \cup \{B_w \sqsubseteq \perp \mid (w, v') \in \mathcal{E}\}, \\ \mathcal{A} &:= \{B_w(a) \mid (v, w) \in \mathcal{E}\} \cup \{B_u(b)\}. \end{aligned}$$

It is easy to see that there is a simple path from v to v' passing through (u, u') iff there is a repair for $\mathcal{K}_{\mathcal{G}}$ that does not contain the axiom $B_u \sqsubseteq B_{u'}$. Thus, such a path exists iff b is *not* a cautious instance *nor* an IR instance of $B_{u'}$. \square

Interestingly, despite this hardness result, the duality between MinAs and repairs can be exploited to produce an any-time algorithm for deciding intersection repair instances. The main idea of this algorithm consists on enumerating all the possible MinAs for inconsistency of the KB. Since the union of all MinAs and the intersection of all repairs are complements of each other, any set of MinAs provides an approximation of the intersection of all repairs. If it is possible to enumerate all MinAs with only a polynomial (on the size of the KB) delay

between each new output, then we can efficiently improve this approximation in polynomial time steps.

A first step towards developing this algorithm is to show that MinAs can be enumerated in polynomial delay. It was previously shown that this holds for *DL-Lite_{Horn}* TBoxes, but the case in which an ABox considered was left open [29,30]. We build on those previous ideas and provide an algorithm capable of handling ABoxes too.

First, we make a small simplifying assumption. For the following approach, we assume that the TBox contains no role inclusions $R_1 \sqsubseteq R_2$. Notice that, in the case of *DL-Lite_{Horn}*, this assumption is without loss of generality since the role inclusion $R_1 \sqsubseteq R_2$ can be equivalently expressed through the CI $\exists R_1 \sqsubseteq \exists R_2$.

Our algorithm is based on the notion of a hypergraph. Formally, a (directed) *hypergraph* is a pair $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is a finite set of *vertices* and \mathcal{E} is a set of *edges* of the form $V \rightarrow v$, where $V \subseteq \mathcal{V}$ and $v \in \mathcal{V}$. Given two vertices $v, w \in \mathcal{V}$, a *path* from v to w in \mathcal{H} is a sequence of edges $\mathcal{P} = V_1 \rightarrow v_1, \dots, V_n \rightarrow v_n$ such that for every $i, 1 \leq i \leq n$, $V_i \subseteq \{v\} \cup \bigcup_{j=1}^{i-1} \{v_j\}$ and $v_n = w$. Such a path is called *simple* if no subsequence of \mathcal{P} is also a path from v to w .

Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we construct a directed hypergraph $\mathcal{H}_{\mathcal{K}}$ as follows. The set of vertices $\mathcal{V}_{\mathcal{K}}$ of $\mathcal{H}_{\mathcal{K}}$ contains one element v_a for each individual name a appearing in the ABox \mathcal{A} , and an element w_B for every concept B appearing in the KB \mathcal{K} ; that is, either in the TBox or in the ABox. We call the vertices of the form v_a *individual nodes*. The set of hyperedges of this hypergraph is defined by $\mathcal{E}_{\mathcal{K}} := \mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{A}}$, where

$$\begin{aligned} \mathcal{E}_{\mathcal{A}} &:= \{v_a \rightarrow w_B \mid B(a) \in \mathcal{A}\} \cup \{v_a \rightarrow w_{\exists R}, v_b \rightarrow w_{\exists R^-} \mid R(a, b) \in \mathcal{A}\}, \\ \mathcal{E}_{\mathcal{T}} &:= \{\{w_{B_1}, \dots, w_{B_n}\} \rightarrow w_B \mid B_1 \sqcap \dots \sqcap B_n \sqsubseteq B \in \mathcal{T}\}. \end{aligned}$$

It is easy to see that \mathcal{K} is inconsistent iff there is a path from some v_a to w_{\perp} in $\mathcal{H}_{\mathcal{K}}$. More interestingly, every simple path of this kind corresponds to a MinA for the inconsistency of \mathcal{K} . Unfortunately, this relationship between simple paths and MinAs is not bijective. As the following example shows, two simple paths may correspond to the same MinA.

Example 5. Consider the *DL-Lite_{Horn}* KB $\mathcal{K}_{\text{exa}} = (\mathcal{T}_{\text{exa}}, \mathcal{A}_{\text{exa}})$ defined by

$$\begin{aligned} \mathcal{A}_{\text{exa}} &:= \{P(a, b), P(b, a), C(a)\} \\ \mathcal{T}_{\text{exa}} &:= \{\exists P \sqcap \exists P^- \sqsubseteq B, B \sqsubseteq \perp, C \sqcap \exists P \sqsubseteq \perp\}. \end{aligned}$$

The hypergraph $\mathcal{H}_{\mathcal{K}_{\text{exa}}}$ is depicted in Figure 1(a). Two simple paths from an individual node to w_{\perp} are shown below (Figures 1(b) and 1(c)). Clearly, both paths correspond to the same MinA $\mathcal{M} = (\mathcal{T}_1, \mathcal{A}_1)$ with $\mathcal{A}_1 := \{P(a, b), P(b, a)\}$ and $\mathcal{T}_1 := \{\exists P \sqcap \exists P^- \sqsubseteq B, B \sqsubseteq \perp\}$.

In order to enumerate all MinAs with polynomial delay, we thus try to enumerate all the simple paths leading from an individual node to w_{\perp} , but taking care of removing all those paths that would yield a repeated MinA. The idea behind the enumeration of all simple paths in a hypergraph is, given one such path,

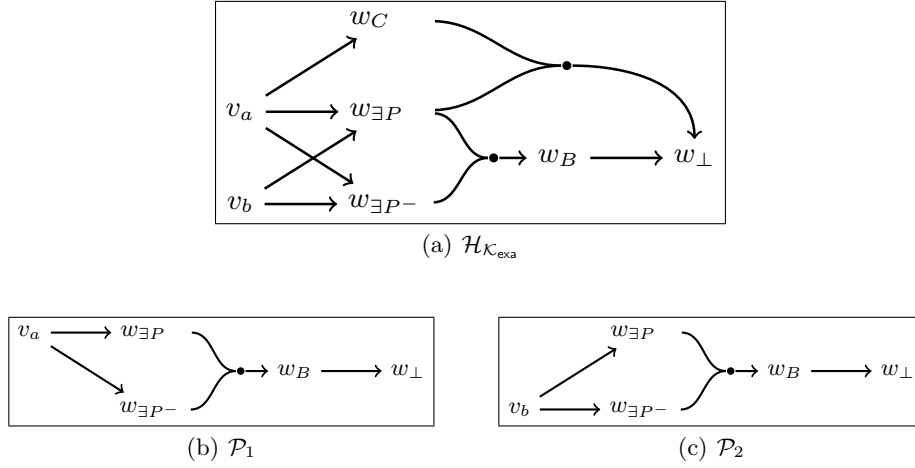


Fig. 1. The hypergraph $\mathcal{H}_{\mathcal{K}_{\text{exa}}}$ from Example 5, and two simple paths.

construct a set of sub-hypergraphs that partition the set of all remaining simple paths from an individual node to w_{\perp} , in the sense that each such path exists in exactly one of the generated sub-hypergraphs. The main insight needed for dealing with the repetition of MinAs is that role assertions $R(a, b)$ generate two hyperedges in $\mathcal{H}_{\mathcal{K}}$, even though they correspond to the same ABox axiom. Thus, these two hyperedges need to be always considered simultaneously: either they are both included, or both excluded in the search of a new simple path.

One (arbitrary) simple path from an individual node to w_{\perp} can be found in polynomial time using standard techniques [25, 27]. Given such a path \mathcal{P} , let v_a be the (unique) individual node appearing in it. Then, by definition \mathcal{P} is of the form $L_1 \rightarrow r_1, \dots, L_n \rightarrow r_n$, where n is the size of \mathcal{P} . Intuitively, the sequence defined by \mathcal{P} provides an ordering of the edges in such a way that all the required nodes to traverse an edge are visited before the head is observed.

Given \mathcal{P} and $n := |\mathcal{P}|$, we define n subgraphs of $\mathcal{H}_{\mathcal{K}}$ as follows. For each $i, 1 \leq i \leq n$, let

$$\mathcal{H}'_i := \mathcal{H}_{\mathcal{K}} \setminus \bigcup_{i < j \leq n} \{L \rightarrow r \in \mathcal{H}_{\mathcal{K}} \mid r = r_j, L \neq L_j\}.$$

Then we define

$$\mathcal{H}_i := \begin{cases} \mathcal{H}'_i \setminus \{v_a \rightarrow w_{\exists R}, v_b \rightarrow w_{\exists R-}\} & \text{if } L_i \rightarrow r_i \text{ was created by } R(a, b) \in \mathcal{A}, \\ \mathcal{H}'_i \setminus \{L_i \rightarrow r_i\} & \text{otherwise.} \end{cases}$$

Example 6. Consider again the KB \mathcal{K}_{exa} from Example 5, and the path \mathcal{P}_1 depicted in Figure 1(b), which is defined by the sequence

$$v_a \rightarrow w_{\exists P}, \quad v_a \rightarrow w_{\exists P-}, \quad \{w_{\exists P}, w_{\exists P-}\} \rightarrow w_B, \quad w_B \rightarrow w_{\perp}$$

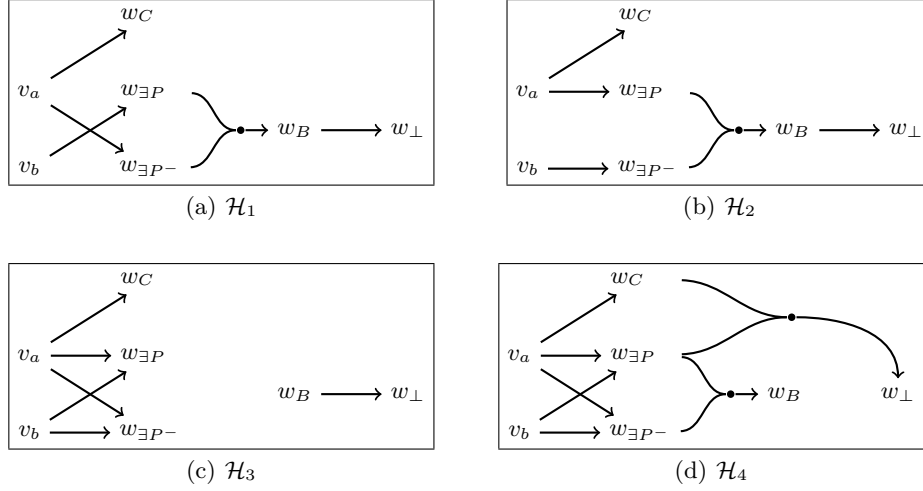


Fig. 2. The subgraphs $\mathcal{H}_1, \dots, \mathcal{H}_4$ of $\mathcal{H}_{\mathcal{K}_{\text{exa}}}$ obtained from \mathcal{P}_1 .

The four subgraphs $\mathcal{H}_1, \dots, \mathcal{H}_4$ of $\mathcal{H}_{\mathcal{K}_{\text{exa}}}$ obtained through this path are depicted in Figure 2. As it can be seen from the figure, only \mathcal{H}_4 contains a path from an individual node to w_{\perp} . This path represents the only other MinA for the inconsistency of \mathcal{K}_{exa} .

Intuitively, each hypergraph \mathcal{H}_i can only contain paths reaching w_{\perp} in which the last edges coincide with the last $n - i$ edges of the path \mathcal{P} . For that reason, in the previous example only \mathcal{H}_4 contains the edge $\{w_C, w_{\exists P}\} \rightarrow w_{\perp}$; in all other subgraphs, this edge is removed to guarantee that only paths ending with the edge $w_B \rightarrow w_{\perp}$ are considered. Taking this intuition into account, it is easy to see that the following result holds.

Lemma 7. *Let \mathcal{P} be a simple path from some individual node v_a to w_{\perp} in $\mathcal{H}_{\mathcal{K}}$, and \mathcal{H}_i , $1 \leq i \leq n := |\mathcal{P}|$ constructed as above. Then, for every MinA \mathcal{M} , if $\mathcal{P} \not\subseteq \mathcal{H}_{\mathcal{M}}$, then there exists exactly one i , $1 \leq i \leq n$ such that $\mathcal{H}_{\mathcal{M}}$ contains a MinA for inconsistency in \mathcal{H}_i .*

Notice that the sets \mathcal{H}_i can be all computed in polynomial time. Hence, Lemma 7 suggests an approach for enumerating all MinAs with only polynomial delay between answers. Recall here that our interest is not in enumerating these MinAs *per se*, but rather on finding their union to answer intersection repair instance queries in any-time. Algorithm 1 shows an approach for doing this. The algorithm iteratively updates the intersection of all repairs (stored in the set \mathcal{U}) and checks whether this KB still entails the instance query q . As soon as the answer is no, the algorithm stops and returns this answer. Otherwise, it answers yes. This algorithm can be stopped at any time, in which case, only an (upper) approximation of $\bigcap_{\mathcal{R} \in \text{Rep}_{\mathcal{K}}} \mathcal{R}$ is computed. Although the answer (yes) in this case

Algorithm 1 Any-time intersection repair instance query answering.

Procedure $\text{IC}(\mathcal{K}, q)$ (\mathcal{K} inconsistent *DL-Lite_{Horn}* KB, q instance query)

- 1: $\mathcal{U} \leftarrow \mathcal{K}$
- 2: $\text{ALL-MINAS}(\mathcal{H}_{\mathcal{K}}, q)$
- 3: **return** “yes”

Procedure $\text{ALL-MINAS}(\mathcal{H}_{\mathcal{K}}, q)$

- 1: $\mathcal{M} \leftarrow$ a MinA in $\mathcal{H}_{\mathcal{K}}$
 - 2: $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{M}$
 - 3: **if** $\mathcal{U} \not\models q$ **then return** “no”
 - 4: **for** $1 \leq i \leq |\mathcal{M}|$ **do**
 - 5: compute \mathcal{H}_i from \mathcal{M}
 - 6: **if** there is a path from an individual node to w_{\perp} in \mathcal{H}_i **then**
 - 7: $\text{ALL-MINAS}(\mathcal{H}_i, q)$
-

is not guaranteed to be correct, the approximation gets tighter in polynomial intervals.

We now turn our attention to inconsistency-tolerant instance query answering when repairs are limited to removing assertions from the ABox only.

4 Instance Checking Under ABox Repairs

Answering inconsistency-tolerant instance queries under ABox repairs has been shown to be harder, in terms of computational complexity, to standard instance query answering over KBs [6–8]. The blame for this increase in complexity is often laid on the exponential number of repairs available. We argue that the issue goes deeper than this argument suggests showing that brave and cautious instance checking are hard already for KBs having polynomially many ABox repairs. One reason behind this result is the fact that even knowing whether a set of repairs is equal to $\text{Rep}_{\mathcal{K}}$ is computationally expensive.²

Definition 8. Let \mathcal{K} be a KB and \mathfrak{R} a set of ABox repairs of \mathcal{K} . The problem *ALL-REPS* consists on deciding whether $\mathfrak{R} = \text{Rep}_{\mathcal{K}}$.

Theorem 9. *ALL-REPS* is *coNP-complete*.

Proof. We prove *coNP-hardness* through a reduction from the negation of the following NP-complete problem [31]: given a monotone Boolean formula φ (that is, a propositional formula without negations) and a set \mathfrak{V} of valuations of the variables in φ , decide whether there exists a maximal valuation \mathcal{V} of the variables in φ that *falsifies* φ and does not contain any valuation in \mathfrak{V} .³

Given an instance φ, \mathfrak{V} of this problem, let **sub** be the set of all subformulas of φ , and **csub** the subset of **sub** not containing any propositional variable. For

² Throughout this section, $\text{Rep}_{\mathcal{K}}$ denotes the set of all ABox repairs.

³ For simplicity, we identify a propositional valuation \mathcal{V} with the set of variables that it makes true.

each $\psi \in \text{sub}$, we create a concept name A_ψ , and for every $\psi \in \text{csub}$ define the TBox

$$\mathcal{T}_\psi := \begin{cases} \{A_{\xi_1} \sqcap A_{\xi_2} \sqsubseteq A_\psi\} & \text{if } \psi = \xi_1 \wedge \xi_2 \\ \{A_{\xi_1} \sqsubseteq A_\psi, A_{\xi_2} \sqsubseteq A_\psi\} & \text{if } \psi = \xi_1 \vee \xi_2. \end{cases}$$

Let now

$$\begin{aligned} \mathcal{T}_{\varphi, \mathfrak{W}} &:= \bigcup_{\psi \in \text{csub}} \mathcal{T}_\psi \cup \{A_\varphi \sqsubseteq \perp\} \\ \mathcal{A}_{\varphi, \mathfrak{W}} &:= \{A_x(a) \mid x \text{ is a variable from } \varphi\}. \end{aligned}$$

It is easy to see that $\mathfrak{R} := \{(\mathcal{T}_{\varphi, \mathfrak{W}}, \{A_x(a) \mid x \in \mathcal{W}\}) \mid \mathcal{W} \in \mathfrak{W}\}$ is the set of all ABox repairs of $(\mathcal{T}_{\varphi, \mathfrak{W}}, \mathcal{A}_{\varphi, \mathfrak{W}})$ iff every maximal valuation falsifying φ contains some $\mathcal{W} \in \mathfrak{W}$. \square

Since all repairs can be computed in exponential time, e.g. by testing all possible sub-KBs, this theorem intuitively means that there exist KBs that have polynomially many repairs, but finding them all requires super-polynomial time. In terms of enumeration complexity theory, the set of all repairs cannot be computed in *output polynomial time* [17].

Obviously, Theorem 9 does not imply that inconsistency-tolerant semantics are necessarily hard on the number of repairs. However, it does give an indication that efficient algorithms cannot rely on finding all repairs. We strengthen previous results by showing that these problems remain hard even in the number of ABox repairs. First, we recall the notion of *repair-polynomial* from [22].

Definition 10. *An inconsistency-tolerant decision problem w.r.t. a KB \mathcal{K} is called repair-polynomial if it can be solved by an algorithm that runs in polynomial time on the size of both \mathcal{K} and $\text{Rep}_{\mathcal{K}}$.*

Theorem 11. *Cautious instance checking w.r.t. ABox repairs is not repair-polynomial, unless $P = NP$.*

Proof. We show this result through a reduction similar to the one presented in the proof of Theorem 9. Given φ and \mathfrak{W} , we construct the TBoxes \mathcal{T}_ψ and the ABox $\mathcal{A}_{\varphi, \mathfrak{W}}$ as in the forementioned proof. But now we set

$$\mathcal{T}'_{\varphi, \mathfrak{W}} := \bigcup_{\psi \in \text{csub}} \mathcal{T}_\psi \cup \{A_\varphi \sqsubseteq \perp\} \cup \left\{ \prod_{x \in \mathcal{W}} A_x \sqsubseteq B \mid \mathcal{W} \in \mathfrak{W} \right\},$$

where B is a fresh concept name. Then a is a cautious instance of B w.r.t. $\mathcal{K}_\varphi := (\mathcal{T}'_{\varphi, \mathfrak{W}}, \mathcal{A}_{\varphi, \mathfrak{W}})$ under ABox repairs iff every maximal valuation falsifying φ contains some $\mathcal{W} \in \mathfrak{W}$. Notice that $(\mathcal{T}'_{\varphi, \mathfrak{W}}, \mathcal{A}_{\varphi, \mathfrak{W}})$ has as many ABox repairs as φ has maximal valuations falsifying it.

Suppose now that brave instance checking was repair-polynomial. Then, there would be an algorithm for solving it that would run in time bounded by $p(\mathcal{K}, |\text{Rep}_{\mathcal{K}}|)$, where p is a polynomial. Then, we can decide whether there

is a new maximal valuation falsifying φ by running this algorithm for time $p(\mathcal{K}_\varphi, |\text{Rep}_{\mathcal{K}_\varphi}|)$. If it answers *no* or does not finish in that time bound, then a new valuation exists. Otherwise, the no new valuation exists. \square

A similar method can be used to prove that brave instance checking is not repair-polynomial either.

Theorem 12. *Brave instance checking w.r.t. ABox repairs is not repair-polynomial, unless $P = NP$.*

Proof. We use an idea similar to the proof of Theorem 11, but using a reduction from the following NP-complete problem [5, 10, 11]: given a monotone Boolean formula φ and a set \mathfrak{V} of valuations, decide whether there exists a minimal valuation \mathcal{V} that *satisfies* φ and does not contain any valuation in \mathfrak{V} .⁴

Given an instance φ, \mathfrak{V} of this problem, we construct the TBoxes $\mathcal{T}_\psi, \psi \in \text{csub}$ and the ABox $\mathcal{A}_{\varphi, \mathfrak{V}}$ as in the proof of Theorem 9. We define

$$\mathcal{T}_{\varphi, \mathfrak{V}}'' := \bigcup_{\psi \in \text{csub}} \mathcal{T}_\psi \cup \left\{ \prod_{x \in \mathcal{W}} A_x \sqsubseteq \perp \mid \mathcal{W} \in \mathfrak{V} \right\}.$$

Then, a is a brave instance of A_φ iff there exists a valuation satisfying φ that does not contain any $\mathcal{W} \in \mathfrak{V}$. Using the same argument from the proof of Theorem 11, this shows that brave instance checking is not repair-polynomial. \square

To the best of our efforts, we have not been able to find a similar hardness result for IR instance checking. However, it is possible to show that the any-time method described in the previous section cannot work when dealing with ABox repairs. The reason behind this negative result is that ABox MinAs cannot be enumerated with polynomial delay. In fact, they cannot be enumerated in output polynomial time.

Theorem 13. *Given an inconsistent KB \mathcal{K} and a set \mathfrak{M} of ABox MinAs of \mathcal{K} , deciding whether $\mathfrak{M} = \text{Min}_{\mathcal{K}}$ is coNP-complete.*

Proof. We show hardness through a reduction from the same NP-complete problem used in the proof of Theorem 12.

Given an instance φ, \mathfrak{V} of this problem, we construct $\mathcal{T}_{\varphi, \mathfrak{V}}$ and $\mathcal{A}_{\varphi, \mathfrak{V}}$ as in the proof of Theorem 9. Then, $\mathfrak{M} := \{(\mathcal{T}_{\varphi, \mathfrak{V}}, \{A_x(a) \mid x \in \mathcal{W}\}) \mid \mathcal{W} \in \mathfrak{V}\}$ is the set of all ABox MinA of $(\mathcal{T}_{\varphi, \mathfrak{V}}, \mathcal{A}_{\varphi, \mathfrak{V}})$ iff every minimal valuation satisfying φ contains some $\mathcal{W} \in \mathfrak{V}$. \square

Using standard techniques from enumeration complexity (see e.g. [19]), it is easy to show that Theorem 13 implies the impossibility of enumerating in output polynomial time.

Corollary 14. *All ABox MinAs of an inconsistent DL-Lite_{Horn} KB cannot be enumerated in output polynomial time (unless $P = NP$).*

⁴ One can think of this problem as the dual of the one considered in the proof of Theorem 9.

5 Conclusions

We have studied the complexity of answering inconsistency-tolerant instance queries under different semantics considered in the literature for the description logic $DL-Lite_{Horn}$. To the best of our knowledge, we are the first to consider any case of inconsistent-tolerant query answering problems under KB repairs. In the literature, the TBox is typically considered to be correct and fixed [20, 21, 23, 24, 32, 35].

We have shown that considering KB repairs reduces the complexity of inconsistency-tolerant instance checking. Indeed, although cautious and IR instance checking is shown to be intractable, the brave semantics can be verified in polynomial time in this case. Moreover, we provided an any-time algorithm for verifying IR instances. The algorithm is based on enumerating all MinAs with polynomial delay, and verifying that the complement of their union (which increasingly approximates the intersection of all repairs) still entails the desired instance query. A simple consequence of this algorithm is that IR instance checking is *MinA-polynomial*; that is, it can be solved in time polynomial on the size of the KB and the number of MinAs.

For the cautious semantics, it remains open whether instance checking w.r.t. KB repairs is repair-polynomial or MinA-polynomial. Given recent work on the enumeration of maximal consistent subformulae of a Horn formula [26], it is likely that repairs in $DL-Lite_{Horn}$ can be enumerated with polynomial delay, which would suggest a repair-polynomial algorithm for cautious instance checking. In practical terms, however, a MinA-polynomial algorithm would be more interesting. Indeed, empirical analyses have shown that realistic ontologies typically contain very few MinAs, but a very large number of repairs [2, 3, 22].

When restricting to ABox repairs, we have shown that the causes for hardness for the inconsistent-tolerant semantics go beyond the sheer number of repairs available, as suggested in previous work. We have shown that brave and cautious instance checking are not repair-polynomial. This means that these problems may take super-polynomial time, even for KBs that have a polynomial number of repairs. In addition, the any-time algorithm proposed for IR instance checking under KB repairs cannot work for ABox repairs, since ABox MinAs cannot be enumerated in output polynomial time, much less with a polynomial delay.

The study of repair-polynomial and MinA-polynomial algorithms is much in the spirit of parameterized complexity theory [9]. In this context, *repair-polynomial* is analogous to *fixed-parameter tractability* where the number of repairs is the fixed parameter. It thus makes sense to try to understand the precise parameterized complexity class to which these problems belong. Dually, it would be interesting to find other meaningful parameters under which tractability can be regained.

Another important open question is the effect on the complexity when more complex queries, such as conjunctive queries, are considered instead of the simple instance queries that were the scope of this paper. We conjecture that brave and cautious conjunctive query answering is not repair-polynomial even under KB repairs. A thorough analysis of this case will be the focus of future work.

We conclude by highlighting that there exist other inconsistency-tolerant semantics beyond the three studied in this paper; see [8] for some examples. These differ mainly on the properties of the repairs that are considered for making an inference. If our conjecture is correct and all KB repairs from a $DL\text{-}Lite_{Horn}$ KB can be enumerated with polynomial delay, then most of these semantics will be repair-polynomial under KB repairs. However, this result is unlikely to be helpful in practice. Thus, in future work we will focus on developing specialized techniques and finding special query cases for which inconsistency-tolerant instance checking—and conjunctive query answering in general—remains feasible.

References

1. Arenas, M., Bertossi, L.E., Chomicki, J.: Consistent query answers in inconsistent databases. In: Vianu, V., Papadimitriou, C.H. (eds.) Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 99). pp. 68–79. ACM Press (1999), <http://doi.acm.org/10.1145/303976.303983>
2. Arif, M.F., Mencía, C., Ignatiev, A., Manthey, N., Peñaloza, R., Marques-Silva, J.: BEACON: an efficient sat-based tool for debugging $EL+$ ontologies. In: Creignou, N., Berre, D.L. (eds.) Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT 2016). Lecture Notes in Computer Science, vol. 9710, pp. 521–530. Springer-Verlag (2016), http://dx.doi.org/10.1007/978-3-319-40970-2_32
3. Arif, M.F., Mencía, C., Marques-Silva, J.: Efficient axiom pinpointing with $EL2MCS$. In: Hölldobler, S., Krötzsch, M., Peñaloza, R., Rudolph, S. (eds.) Proceedings of the 38th German Conference on Artificial Intelligence (KI 2007). Lecture Notes in Computer Science, vol. 9324, pp. 225–233. Springer-Verlag (2015), http://dx.doi.org/10.1007/978-3-319-24489-1_17
4. Artale, A., Calvanese, D., Kontchakov, R., Zakharyashev, M.: The $DL\text{-}Lite$ family and relations. *Journal of Artificial Intelligence Research* 36, 1–69 (2009)
5. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic \mathcal{EL}^+ . In: Proceedings of the 30th German Conference on Artificial Intelligence (KI 2007). Lecture Notes in Artificial Intelligence, vol. 4667, pp. 52–67. Springer-Verlag, Osnabrück, Germany (2007)
6. Bienvenu, M.: On the complexity of consistent query answering in the presence of simple ontologies. In: Hoffmann, J., Selman, B. (eds.) Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12). AAAI Press (2012), <http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4928>
7. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Querying inconsistent description logic knowledge bases under preferred repair semantics. In: Brodley, C.E., Stone, P. (eds.) Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14). pp. 996–1002. AAAI Press (2014), <http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8231>
8. Bienvenu, M., Rosati, R.: Tractable approximations of consistent query answering for robust ontology-based data access. In: Rossi, F. (ed.) Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI-13). pp. 775–781. AAAI Press (2013), <http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6904>

9. Downey, R.G., Fellows, M.: Parameterized Complexity. Monographs in Computer Science, Springer-Verlag (1999)
10. Eiter, T., Gottlob, G.: Identifying the minimal transversals of a hypergraph and related problems. Tech. Rep. CD-TR 91/16, Christian Doppler Laboratory for Expert Systems, TU Vienna (1991)
11. Eiter, T., Gottlob, G.: Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing* 24(6), 1278–1304 (1995)
12. Eiter, T., Lukasiewicz, T., Predoiu, L.: Generalized consistent query answering under existential rules. In: Baral, C., Delgrande, J.P., Wolter, F. (eds.) *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*. pp. 359–368. AAAI Press (2016)
13. Fredman, M.L., Khachiyan, L.: On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms* 21(3), 618–628 (1996)
14. Gottlob, G., Malizia, E.: Achieving new upper bounds for the hypergraph duality problem through logic. *The Computing Research Repository (CoRR) abs/1407.2912* (2014), <http://arxiv.org/abs/1407.2912>
15. Hagen, M.: Algorithmic and Computational Complexity Issues of MONET. Ph.D. dissertation, Institut für Informatik, Friedrich-Schiller-Universität Jena (2008)
16. Horridge, M., Parsia, B., Sattler, U.: Explaining inconsistencies in OWL ontologies. In: Godo, L., Pugliese, A. (eds.) *Proceedings of the 3rd International Conference on Scalable Uncertainty Management (SUM 2009)*. *Lecture Notes in Computer Science*, vol. 5785, pp. 124–137. Springer-Verlag (2009)
17. Johnson, D.S., Yannakakis, M., Papadimitriou, C.H.: On generating all maximal independent sets. *Information Processing Letters* 27(3), 119–123 (1988)
18. Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C.: Repairing unsatisfiable concepts in OWL ontologies. In: Sure, Y., Domingue, J. (eds.) *Proceedings of the 3rd European Semantic Web Conference (ESWC 2006)*. *Lecture Notes in Computer Science*, vol. 4011, pp. 170–184. Springer-Verlag (2006)
19. Kavvadias, D.J., Sideri, M., Stavropoulos, E.C.: Generating all maximal models of a boolean expression. *Information Processing Letters* 74(3-4), 157–162 (2000), [http://dx.doi.org/10.1016/S0020-0190\(00\)00023-5](http://dx.doi.org/10.1016/S0020-0190(00)00023-5)
20. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Hitzler, P., Lukasiewicz, T. (eds.) *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems (RR 2010)*. *Lecture Notes in Computer Science*, vol. 6333, pp. 103–117. Springer-Verlag (2010), http://dx.doi.org/10.1007/978-3-642-15918-3_9
21. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant query answering in ontology-based data access. *Journal of Web Semantics* 33, 3–29 (2015), <http://dx.doi.org/10.1016/j.websem.2015.04.002>
22. Ludwig, M., Peñaloza, R.: Error-tolerant reasoning in the description logic *el*. In: Fermé, E., Leite, J. (eds.) *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA 2014)*. *Lecture Notes in Artificial Intelligence*, vol. 8761, pp. 107–121. Springer-Verlag, Madeira, Portugal (2014)
23. Lukasiewicz, T., Martínez, M.V., Pieris, A., Simari, G.I.: From classical to consistent query answering under existential rules. In: Bonet, B., Koenig, S. (eds.) *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*. pp. 1546–1552. AAAI Press (2015), <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9817>
24. Lukasiewicz, T., Martínez, M.V., Simari, G.I.: Inconsistency handling in datalog+/- ontologies. In: Raedt, L.D., Bessière, C., Dubois, D., Doherty, P., Fras-

- coni, P., Heintz, F., Lucas, P.J.F. (eds.) Proceedings of the 20th European Conference on Artificial Intelligence (ECAI-2012). *Frontiers in Artificial Intelligence and Applications*, vol. 242, pp. 558–563. IOS Press (2012), <http://dx.doi.org/10.3233/978-1-61499-098-7-558>
25. Marino, A.: *Analysis and Enumeration: Algorithms for Biological Graphs*. Springer-Verlag (2015)
 26. Marques-Silva, J., Ignatiev, A., Mencia, C., Peñaloza, R.: Efficient reasoning for inconsistent horn formulae. In: Michael, L., Kakas, A.C. (eds.) Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA 2016). *Lecture Notes in Computer Science*, vol. 10021, pp. 336–352. Springer-Verlag (2016), http://dx.doi.org/10.1007/978-3-319-48758-8_22
 27. Nielsen, L.R., Andersen, K.A., Pretolani, D.: Finding the K shortest hyperpaths. *Computers and Operations Research* 32(6), 1477–1497 (2005)
 28. Parsia, B., Sirin, E., Kalyanpur, A.: Debugging OWL ontologies. In: Ellis, A., Hagino, T. (eds.) Proceedings of the 14th international conference on World Wide Web (WWW 2005). pp. 633–640. ACM (2005)
 29. Peñaloza, R., Sertkaya, B.: Complexity of axiom pinpointing in the dl-lite family of description logics. In: Coelho, H., Studer, R., Wooldridge, M. (eds.) Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-2010). *Frontiers in Artificial Intelligence and Applications*, vol. 215, pp. 29–34. IOS Press (2010)
 30. Peñaloza, R., Sertkaya, B.: Understanding the complexity of axiom pinpointing in lightweight description logics. *Artificial Intelligence* (2017), to appear
 31. Peñaloza, R.: *Axiom pinpointing in description logics and beyond*. Ph.D. thesis, Technische Universität Dresden (2009), <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-24743>
 32. Rosati, R.: On the complexity of dealing with inconsistency in description logic ontologies. pp. 1057–1062. AAAI Press (2011), <http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-181>
 33. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Gottlob, G., Walsh, T. (eds.) Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03). pp. 355–362. Morgan Kaufmann (2003)
 34. Schlobach, S., Huang, Z., Cornet, R., Harmelen, F.: Debugging incoherent terminologies. *Journal of Automated Reasoning* 39(3), 317–349 (2007)
 35. Tsalapati, E., Stoilos, G., Stamou, G.B., Koletsos, G.: Efficient query answering over expressive inconsistent description logics. pp. 1279–1285. AAAI Press (2016), <http://www.ijcai.org/Abstract/16/185>