Understanding the Complexity of Axiom Pinpointing in Lightweight Description Logics

Rafael Peñaloza[a,*], Barış Sertkaya[b]

[a] *KRDB Research Centre, Free University of Bozen-Bolzano, Italy*
[b] *Frankfurt University of Applied Sciences, Frankfurt, Germany*

Abstract

Lightweight description logics are knowledge representation formalisms characterised by the low complexity of their standard reasoning tasks. They have been successfully employed for constructing large ontologies that model domain knowledge in several different practical applications. In order to maintain these ontologies, it is often necessary to detect the axioms that cause a given consequence. This task is commonly known as axiom pinpointing.

In this paper, we provide a thorough analysis of the complexity of several decision, counting, and enumeration problems associated to axiom pinpointing in lightweight description logics.

Keywords: Description Logics, Axiom Pinpointing, Computational Complexity, Counting Complexity, Enumeration Complexity

1. Introduction

The success of description logics (DLs) [1] as knowledge representation languages has been witnessed by the development of more, and usually larger, ontologies based on these formalisms. In these languages, the knowledge of a domain is represented via a set of axioms that express the relationships between the different notions being modelled. With the help of advanced ontology editing and versioning tools [2–4], it is becoming increasingly easier for domain experts who may not be versed in the underlying logics to model part of their knowledge, and collaborate with other experts to represent large domains.

It is well known that ontology development is an error-prone task, where minor variations may lead to unwanted or erroneous consequences. Moreover, even when the ontology is correct and no errors are detected, some of its (implicit) consequences can be surprising to the knowledge engineers and domain experts.

[*] Corresponding author
Email addresses: rafael.penaloza@unibz.it (Rafael Peñaloza), sertkaya@fb2.fra-uas.de (Barış Sertkaya)

In order to understand the reason for a consequence, it is helpful to extract only those axioms that are responsible for it [5, 6]. This task is commonly known as axiom pinpointing in the DL literature. Briefly, axiom pinpointing refers to the task of finding minimal sets of axioms that are responsible for a given entailment. These sets are sometimes called *justifications* in the literature [7–9], as they clarify the causes for the consequence to appear. To avoid confusion with other uses of the term, we use the more technical term *MinA*, which stands for "minimal axiom set."

Two subfamilies of DLs, namely the DL-Lite [10] and $\mathcal{EL}$ [11] families, have attracted special interest due to their polynomial-time standard reasoning problems. Thus, they are typically referred to as "lightweight" DLs. Due to their low computational complexity, these logics have been successfully used to model large ontologies, in particular in the bio-medical domain [12]. In fact, we see more and often larger ontologies being used in practical applications within this domain. The most well-known of these ontologies is the SNOMED CT medical ontology [13] that contains over half a million axioms from a slight extension of $\mathcal{EL}$.

Due to their size and practical use, it is important to provide automated axiom pinpointing tools that can handle these ontologies efficiently. As a first step towards this aim, it is important to understand the theoretical limits of axiom pinpointing for lightweight DLs. In particular, it is interesting to detect the cases in which finding some or all MinAs with certain properties remains tractable. For that reason, in this paper we extensively study the complexity of many decision, counting, and enumeration problems related to axiom pinpointing and its applications for ontology debugging and consequence understanding. More precisely, we study the complexity of enumerating all MinAs with and without a specific order, of counting the total number of MinAs, and of deciding whether some given axioms belong or not to some or all MinAs, among other problems (see Table 5 in page 36 for a full summary of the results).

One important thing to notice is that standard reasoning is a sub-task of axiom pinpointing. Thus, the latter necessarily has a higher (or equal) complexity than the former. By focusing on lightweight DLs, whose standard reasoning problems are polynomial, we also gain insights in the intrinsic complexity that is added by axiom pinpointing, as opposed to mere reasoning. For instance, we observe that deciding whether an axiom belongs to at least one MinA is intrinsically a hard problem, becoming NP-hard even for the most basic kind of axioms studied.

We emphasise that axiom pinpointing is relevant beyond the context of description logics and consequence understanding and repair. MinAs have also been studied, under the name of *minimal unsatisfiable subsets* (MUS) [14] in the context of propositional logic and maximal satisfiability of formulas (or MaxSAT) [15, 16]. As for description logics, computing, counting, and enumerating MinAs has been shown to be fundamental for other non-standard extensions and reasoning tasks. These include, among many others, dealing with trust, provenance, and preferences [17, 18], error-tolerant [19–21] and context-based reasoning [22], as well as some variants of probabilistic [23–25] and fuzzy logic [26]. Studying the complexity of axiom pinpointing and methods for proving their lower bounds has a direct impact on our understanding of those non-standard reasoning problems.

Some of the results presented in this paper appeared previously in preliminary versions. Specifically, in [27, 28] we investigated the complexity of axiom pinpointing in the propositional Horn fragment, and in $\mathcal{EL}$, and in [29, 30] we studied this complexity for the DL-Lite family, which has been very popular due to its success in efficiently accessing large data and answering complex queries on this data [10, 31]. For this family various aspects of finding explanations have already been considered in [32, 33]. The main focus of those papers is on the problem of explaining query answering and ABox reasoning, which are the most standard types of reasoning problems considered in the DL-Lite family. In particular the authors investigate in detail the problem of determining why a value is returned as an answer to a conjunctive query posed to a DL-Lite ABox, why a conjunctive query is unsatifiable, and why a particular value is not returned as answer to a conjunctive query. Complementary to the work in [32, 33] we consider only TBox reasoning. Here, we extend the previous results by considering also inverse roles in DL-Lite. In addition, we filled several gaps regarding the enumeration of MinAs in $DL\text{-}Lite_{core}$ and $DL\text{-}Lite_{krom}$ TBoxes, studied in more detail the $DL\text{-}Lite_{bool}$ case, and added a new case of ordered enumeration that takes into account the size of the MinAs.

The paper is structured as follows. After briefly recalling the basic notions from lightweight DLs, axiom pinpointing, and enumerating and counting complexity classes, we study the complexity of decision problems associated with axiom pinpointing (Section 3). In Sections 4 and 5 we analyse the complexity of enumerating and counting MinAs, respectively. All these results are summarised in Section 6, before giving conclusions and directions for future research.

2. Preliminaries

We briefly recall basic notions from propositional logic, the $\mathcal{EL}$ and DL-Lite families of description logics (DLs), axiom pinpointing, and non-standard complexity measures that deal with enumeration and counting problems.

In propositional logic we build formulae using a set of *propositional variables* and the *Boolean connectives* $\neg$ (negation), $\vee$ (disjunction) and $\wedge$ (conjunction). A variable or its negation is called a *literal*, and a disjunction of literals, e.g. $\neg p_1 \vee \neg p_2 \vee p_3$, is called a *clause*. Clauses like the previous one are sometimes written as *implications* of the form $p_1 \wedge p_2 \rightarrow p_3$. A clause is called a *Horn* (respectively *dual-Horn*) clause if it contains at most one positive (negative) literal, and a *definite Horn (definite dual-Horn)* clause if it contains exactly one positive (respectively negative) literal. Throughout the text we will call definite Horn (definite dual-Horn) clauses just Horn (dual-Horn) clauses for short. In other words, Horn clauses are of the form $p_1 \wedge \ldots \wedge p_n \rightarrow p$ and dual-Horn clauses are of the form $p \rightarrow p_1 \wedge \ldots \wedge p_n$. We will call clauses with exactly one positive and one negative literal, like $p_1 \rightarrow p_2$, *core* clauses.

Table 1: Syntax and semantics of $\mathcal{EL}$ and DL-Lite constructors.

Syntax	Semantics
$\top$	$\Delta^\mathcal{I}$
$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
$\exists r.C$	$\{x \in \Delta^\mathcal{I} \mid \exists y \in \Delta^\mathcal{I} : (x, y) \in r^\mathcal{I} \wedge y \in C^\mathcal{I}\}$
r^-	$\{(x, y) \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid (y, x) \in r^\mathcal{I}\}$
$\geq q\ s$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \mid (x, y) \in s^\mathcal{I}\} \geq q\}$
$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
$\bot$	$\emptyset$
$C \sqsubseteq D$	$C^\mathcal{I} \subseteq D^\mathcal{I}$

In DLs one formalizes the relevant notions of an application domain through *concepts*. Concepts are inductively built from a set of *concept names*, *role names*, and *concept constructors* that are allowed by the particular DL language in use. The knowledge of the domain is then described through a set of *axioms* that restrict the way in which these concepts can be interpreted.

2.1. The $\mathcal{EL}$ Family

Concepts of the DL $\mathcal{EL}$ are formed from a set of concept names $\mathsf{N_C}$ and a set of role names $\mathsf{N_R}$ using the three constructors $\sqcap$ (conjunction), $\exists$ (existential restrictions) and $\top$ (top). More formally, if A is a concept name and r is a role name, then concepts are built using the syntactic rule:

$$C \quad ::= \quad A \quad | \quad \top \quad | \quad \exists r.C \quad | \quad C_1 \sqcap C_2.$$

The semantics of $\mathcal{EL}$ is defined in terms of *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where the *domain* $\Delta^\mathcal{I}$ is a non-empty set of individuals, and the interpretation function $\cdot^\mathcal{I}$ maps each concept name $A \in \mathsf{N_C}$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$ and each role name $r \in \mathsf{N_R}$ to a binary relation $r^\mathcal{I}$ on $\Delta^\mathcal{I}$. The mapping $\cdot^\mathcal{I}$ can be extended to arbitrary $\mathcal{EL}$ concepts as shown in the second colum of Table 1.

An $\mathcal{EL}$ *TBox* is a finite set of *general concept inclusion axioms* (GCIs) of the form $C \sqsubseteq D$, where C, D are two $\mathcal{EL}$ concepts. The interpretation $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$ if, for every GCI $C \sqsubseteq D$ in $\mathcal{T}$ it holds that $C^\mathcal{I} \subseteq D^\mathcal{I}$ (see the last row of Table 1). The main inference problem for $\mathcal{EL}$ is the subsumption problem [34, 35], which is defined as follows: given two $\mathcal{EL}$ concepts C, D and an $\mathcal{EL}$ TBox $\mathcal{T}$, check if C is *subsumed* by D w.r.t. $\mathcal{T}$ (written $\mathcal{T} \models C \sqsubseteq D$); that is, decide whether $C^\mathcal{I} \subseteq D^\mathcal{I}$ holds in every model $\mathcal{I}$ of $\mathcal{T}$.

We will call a concept description *simple* if it is of the form A or $\exists r.A$ for $A \in \mathsf{N_C}, r \in \mathsf{N_R}$, and a GCI a *Horn-$\mathcal{EL}$ GCI* if it is of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq D$, where C_i, D are simple concept descriptions, $1 \leq i \leq n$.

2.2. The DL-Lite Family

DL-Lite concepts are constructed and interpreted in a similar way to $\mathcal{EL}$ concepts. We briefly introduce the syntax of the DL-Lite family following the notation in [10], restricting our attention to the members of DL-Lite that we use in this paper only. For a full overview on DL-Lite, we refer the reader to [10].

Let A be a concept name, r a role name, and q a natural number. Then DL-Lite concepts and roles are constructed as follows:

$$\begin{array}{lcl} s & := & r \mid r^-, \\ B & ::= & \bot \mid A \mid \geq q\ s, \\ C & ::= & B \mid \neg C \mid C_1 \sqcap C_2, \end{array}$$

Concepts of the form B are called *basic*, and those of form C are called *general* concepts. The semantics of this constructors are presented in the second column of Table 1.

A *DL-Lite*$^{\mathcal{N}}_{bool}$ TBox is a set of GCIs of the form $C_1 \sqsubseteq C_2$, where C_1, C_2 are general concepts. A TBox is a *DL-Lite*$^{\mathcal{N}}_{core}$ TBox if its axioms are of the form $B_1 \sqsubseteq B_2$, or $B_1 \sqsubseteq \neg B_2$, where B_1, B_2 are basic concepts. *DL-Lite*$^{\mathcal{N}}_{krom}$ TBoxes generalize core ones by allowing also axioms of the form $\neg B_1 \sqsubseteq B_2$. Finally, a *DL-Lite*$^{\mathcal{N}}_{horn}$ TBox is composed only of axioms of the form $\bigsqcap_k B_k \sqsubseteq B$ with B, B_i basic concepts.

We can drop the superscript $\mathcal{N}$ from the name of the languages by allowing only number restrictions of the form $\geq 1\ s$ for constructing basic concepts. In this case, we will sometimes use the expression $\exists s$ to represent $\geq 1\ s$ with the intuition that $\geq 1\ s$ requires the existence of at least one s-successor. To any of the previously defined TBoxes, we can also add *role inclusion axioms* of the form $s_1 \sqsubseteq s_2$, with s_1, s_2 roles. This will be denoted using the superscript $\mathcal{H}$ in the name; e.g. *DL-Lite*$^{\mathcal{HN}}_{bool}$. Since we are not dealing with so-called individuals in the present work, role inclusion axioms do not add any expressivity to *DL-Lite*$^{\mathcal{H}}_{\alpha}$ TBoxes for any $\alpha \in \{core, horn, krom\}$. Indeed, a basic concept B will only make use of a role s if B is an existential restriction $\exists s$. As we are only interested in concept subsumption, we can then represent the role inclusion axiom $s_1 \sqsubseteq s_2$ through the concept inclusion $\exists s_1 \sqsubseteq \exists s_2$. Thus, the complexity results we present here for *DL-Lite*$_{\alpha}$ TBoxes immediately hold also for *DL-Lite*$^{\mathcal{H}}_{\alpha}$ TBoxes. Note that this may not be true if number restrictions are allowed; that is, the complexity results for *DL-Lite*$^{\mathcal{N}}_{\alpha}$ may not transfer directly to *DL-Lite*$^{\mathcal{HN}}_{\alpha}$. One interesting observation is that, despite increasing the expressivity of DL-Lite, inverse roles do not seem to affect the complexity of axiom-pinpointing as will become apparent in the following sections.

2.3. Knowledge Bases

We refer to both propositional clauses and GCIs, either from $\mathcal{EL}$ or DL-Lite, as *axioms*, and a set of axioms as a *knowledge base (KB)*. We say that a KB is of type Horn (core, dual-Horn, Horn-$\mathcal{EL}$, $\mathcal{EL}$, *DL-Lite*$_{core}$, *DL-Lite*$_{horn}$, *DL-Lite*$_{krom}$, *DL-Lite*$_{bool}$, respectively) if it contains only Horn (core, dual-Horn, Horn-$\mathcal{EL}$, $\mathcal{EL}$, *DL-Lite*$_{core}$, *DL-Lite*$_{horn}$, *DL-Lite*$_{krom}$, or *DL-Lite*$_{bool}$,

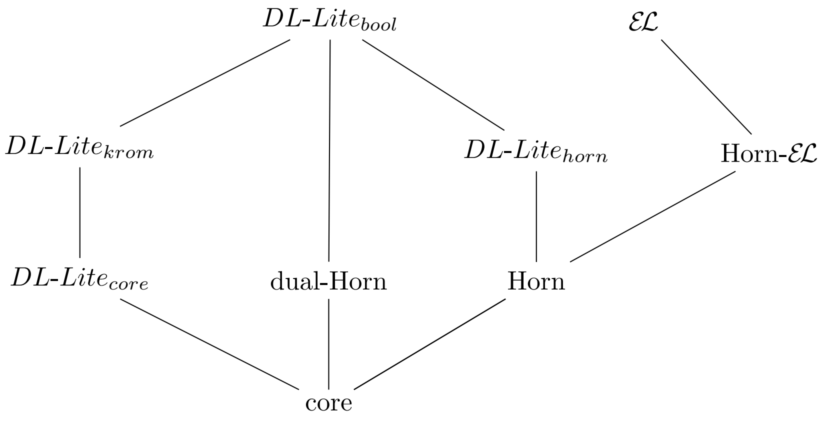

Figure 1: Relative expressivity of the different types of knowledge bases.

respectively) axioms. Throughout this paper, we formulate our problems in a generic way without referring to a specific type of KB, but show the results for each KB type separately. To minimize repetitions, we first show some basic relationships between the different types of KBs presented.

Note that core axioms are a special case of all the other types of axioms introduced above. According to the semantics of these axioms, it is easy to see that dual-Horn KBs are not more expressive than core ones: a dual-Horn axiom $p \to q_1 \land \ldots \land q_n$ can be expressed by the core axioms $p \to q_1, \ldots, p \to q_n$. Hence, any dual-Horn KB can be transformed into an equivalent core KB in linear time. However, as we show in this paper, the complexity of pinpointing-related problems is in general higher for dual-Horn KBs than for core ones. The main reason for this disparity is that in axiom pinpointing, all axioms in a KB are considered independently (see Definition 1 below). Hence, despite being *logically* equivalent, the dual-Horn axiom $p \to q_1 \land \ldots \land q_n$ and the set of core axioms $p \to q_1, \ldots, p \to q_n$ yield different pinpointing results.

Horn axioms are a special kind of Horn-$\mathcal{EL}$ ones, which are themselves a (strict) subclass of $\mathcal{EL}$ axioms; for example, the $\mathcal{EL}$ axiom $A \sqsubseteq B \sqcap C$ is not a Horn-$\mathcal{EL}$ axiom. Likewise, Horn axioms are a special case of $DL\text{-}Lite_{horn}$ ones, but these are not instances of $\mathcal{EL}$ axioms, as the bottom concept $\bot$ cannot be expressed with the $\mathcal{EL}$ constructors. All these relations are depicted in Figure 1. Notice that the figure shows the relationship between the classes of KBs that are expressible in each language, and not their logical expressivity. For example, every core KB is also a dual-Horn KB but the converse is not true—even though every dual-Horn KB can be rewritten into a logically equivalent core one.

We will make use of these relationships to transfer complexity results between the different classes of axioms in the following sections. The complexity of reasoning in the DL-Lite family has been investigated in detail in [10] and the complexity of reasoning in $\mathcal{EL}$ has been investigated in [35].

2.4. Axiom Pinpointing

Axiom pinpointing refers to the task of finding the specific axioms that are responsible for a given consequence to follow from a KB. When dealing with monotonic consequences, as in the case of DLs and propositional logic, this task equivalently corresponds to finding the minimal subsets (w.r.t. set inclusion) of the KB that entail the consequence. We call such a set a MinA.[1]

Definition 1 (MinA)**.** Let $\mathcal{K}$ be a set of axioms and φ be a logical consequence of it, i.e., $\mathcal{K} \models \varphi$. We call a set $\mathcal{M} \subseteq \mathcal{K}$ a *minimal axiom set* or *MinA* for φ in $\mathcal{K}$ if $\mathcal{M} \models \varphi$ and for every $\mathcal{M}' \subset \mathcal{M}$, it holds that $\mathcal{M}' \not\models \varphi$.

It has been shown that a single MinA for a consequence can be computed by calling a reasoner as many times as there are axioms in the KB [37, 38]. The idea is to systematically remove one axiom at a time from the KB, while the consequence is preserved. In particular this means that in every logic with polynomial-time reasoning, a MinA can be computed in polynomial time, with the degree of the polynomial increasing by one.

However, a single consequence from a KB may have several, potentially exponentially many, MinAs, and in many cases computing only one of them does not suffice. For instance, if the consequence is unwanted, then correcting one of the ways in which it can be derived does not ensure that the consequence will not follow from some other combination of axioms in the KB. It is thus important to analyse the complexity of finding all the MinAs or deciding some of their properties.

We notice that some previous work on axiom pinpointing has focused on the computation of a so-called pinpointing formula [39–41], that provides a compact encoding of all MinAs. However, extracting the specific MinAs and other information from this formula is already a hard problem. For that reason, in this paper we focus only on the computation of MinAs as subsets of the KB, and not in the construction of an alternative characterization.

In order to analyze the complexity of enumerating and counting the (potentially exponentially many) solutions to a problem, it is necessary to consider some special complexity classes. Before presenting our results, we briefly recall some of the basic enumeration and counting complexity classes and their properties.

2.5. Complexity of Enumeration

In complexity theory, we are sometimes interested not only in deciding whether a problem has a solution or not, but also in enumerating all solutions of the problem. We call such problems *enumeration problems.* For analyzing the complexity of enumeration problems where the number of solutions can be exponential in the size of the input, one needs appropriate measures. One

[1] These sets are also often called *justifications* in the DL literature [7], and MUSes in propositional logic [36].

such measure is the notion of polynomial delay. We say that an algorithm runs with *polynomial delay* [42] if the time until the first solution is generated, and thereafter the time between any two consecutive solutions, is bounded by a polynomial in the size of the input. An example of such an algorithm is the one given in [43] that enumerates all maximal independent sets of a graph with polynomial delay.

Another measure of performance is to take into account not only the size of the input, but also the size of the output. We say that an enumeration algorithm runs in *output polynomial time* (or *polynomial total time*) [42] if it outputs all solutions in time polynomial in the size of the input *and the output.* Clearly, every polynomial delay algorithm is also an output polynomial algorithm, i.e., the notion of polynomial delay is stronger than the notion of output polynomial. One advantage of an output polynomial algorithm is that it runs in polynomial time whenever the problem has polynomially many solutions. However, an output polynomial algorithm may for instance first compute all solutions and then output them all together. With polynomial delay algorithm on the other hand, a user needs to wait only polynomial time between each retrieved solution, regardless of how many solutions there are in total. Such an algorithm is especially good if one wants to enumerate the solutions one at a time and maybe stop the execution before all of them have been found; for instance after k solutions have been output.

An intermediate notion between polynomial delay and output polynomial is that of incremental polynomial. An enumeration algorithm is *incremental polynomial* if the time required for generating each new solution is polynomial in the size of the input *and the output generated so far.* Clearly, every polynomial delay algorithm is incremental polynomial, and every incremental polynomial algorithm is also ouput polynomial.

A more complicated situation is when the solutions are required to be output in some pre-specified order such as a lexicographic order (see Definition 11). In general, the complexity of enumerating solutions in a specified order is of interest only if polynomial delay or incremental polynomial algorithms exist. Indeed, if all solutions can be enumerated in output polynomial time, then they can be enumerated in output polynomial time in any desired order: one needs only to generate all solutions, sort them and output them in the new order.

Proposition 2. *If an enumeration problem can be solved in output polynomial time, then it can be solved in any polynomially-computable order in output polynomial time.*

A good example of an algorithm that provides ordered solutions with polynomial delay is the one introduced in [42] that generates maximal independent sets of a graph in lexicographic order with polynomial delay.

2.6. Counting Complexity

In applications where one is interested in computing all solutions, it might be useful to know in advance how many of them exist. In complexity theory,

Table 2: Complexity of basic decision problems.

Problem	$DL\text{-}Lite_{bool}^{[\mathcal{N}]}$		all other	
IS-MINA	D^{P}-c	[Theorem 4]	P	[Proposition 3]
MINA-IRRELEVANCE	in Σ_2^{P}	[Corollary 6]	NP-c	[Theorem 5, Corollary 6]
MINA-RELEVANCE	Σ_2^{P}-c	[Theorem 9]	NP-c	[Theorem 7, Corollary 8]

problems that ask *"how many solutions exist"* for a given problem instance are called *counting problems.* For instance the counting version of SAT, called #SAT asks how many satisfying truth assignments exist for a Boolean formula given in CNF. Obviously, a counting problem is at least as hard as its underlying decision problem. For instance, if we could solve #SAT, then we would also be able to solve SAT: an expression is satisfiable if and only if the number of truth assignments that satisfy it is non-zero.

Complexity of counting problems was first investigated by Valiant [44]. For systematically studying and classifying counting problems, he introduced the counting complexity class #P, which is the class of functions that count the number of accepting paths of nondeterministic polynomial-time Turing machines (TMs). Typical members of this class are the problems of counting the number of solutions of NP-complete problems. Valiant showed in [45] however, that there are also #P-complete problems whose underlying decision problem can be solved in polynomial time. It is well known that given a bipartite graph, whether it has a perfect matching can be decided in polynomial time. However, Valiant has shown in [45] that counting the perfect matchings of a given bipartite graph is #P-complete.

A related complexity class is #NP, which is the class of functions that count the number of solutions of nondeterministic polynomial-time TMs with an NP oracle. Intuitively, this class corresponds to counting the number of solutions of Σ_2^{P} problems. In order to prove hardness for counting complexity classes, reductions between problems must also preserve the number of solutions. In general, it suffices to consider *weakly parsimonious reductions* [46, 47]. Intuitively, these reductions transform in polynomial time an instance of the original problem to an instance of the reduced problem such that the number of solutions of the original instance is a polynomially-computable function of the number of solutions of the reduced instance.

3. Preferred and Unwanted Axioms

In this section, we analyze the complexity of the main decision problems that occur in axiom pinpointing, with a focus on the tractable DLs introduced before. Specifically, we study the problems of deciding whether a KB is a MinA (IS-MINA), whether there exists a MinA that does not contain any of a given set of KBs (MINA-IRRELEVANCE), and whether there exists a MinA containing a given axiom (MINA-RELEVANCE). These results are summarized in Table 2. As

Algorithm 1 Computing one MinA for φ in $\mathcal{K}$

```
Procedure FIND-MINA($\mathcal{K}$,$\varphi$)        ($\mathcal{K}$ a KB, $\varphi$ an axiom of the same type)
if $\mathcal{K} \not\models \varphi$ then return "no MinA"
else
    $\mathcal{M} \leftarrow \mathcal{K}$
    for $\alpha \in \mathcal{K}$ do
        if $\mathcal{M} \setminus \{\alpha\} \models \varphi$ then
            $\mathcal{M} \leftarrow \mathcal{M} \setminus \{\alpha\}$
    return $\mathcal{M}$
```

it can be seen, the complexity of these problems is governed by that of standard reasoning in the logic. In Section 4 we analyze the complexity of enumerating MinAs, and in Section 5 we study the complexity of counting MinAs.

3.1. Deciding IS-MINA

We start by formalizing a result that was already hinted in Section 2.4 regarding the problem of deciding whether a given set of axioms is already a MinA.

Problem: IS-MINA
Input: A KB $\mathcal{K}$, an axiom φ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \varphi$, and $\mathcal{M} \subseteq \mathcal{K}$.
Question: Is $\mathcal{M}$ a MinA for φ in $\mathcal{K}$?

A simple approach for computing one MinA is presented in Algorithm 1. The algorithm simply tries to remove axioms from the KB while preserving the consequence. Overall, this algorithm performs $|\mathcal{K}|+1$ entailment tests to output a MinA [37, 38]. This algorithm can also be used to decide IS-MINA: $\mathcal{M}$ is a MinA for φ in $\mathcal{K}$ iff FIND-MINA($\mathcal{M}$,φ) returns $\mathcal{M}$ as output. Notice that Algorithm 1 is agnostic to the type of axioms or the entailment relation used; it requires only an oracle that decides the entailment tests within the `for`-loop. Overall, this yields the following result.

Proposition 3. *Consider an arbitrary logic such that deciding a logical consequence is in the complexity class $\mathfrak{C}$; then,* IS-MINA *is in* $\mathrm{P}^{\mathfrak{C}}$.

A direct consequence of this proposition is that IS-MINA is polynomial in all the logics defined in the previous section, except for *DL-Lite*$_{bool}$ where this approach yields a Δ_2^{P} upper bound. This last complexity bound can be further improved. In fact, we show that IS-MINA is D^{P}-complete for *DL-Lite*$_{bool}^{\mathcal{N}}$ KBs, where D^{P} is the class of problems that can be solved by one NP and one coNP test [48]. Notice that D^{P} contains NP and coNP, and is contained in Δ_2^{P}.

Theorem 4. IS-MINA *is D^{P}-complete for DL-Lite$_{bool}^{\mathcal{N}}$ KBs.*

Proof. To show that the problem is in D^{P} one only needs to observe that $\mathcal{M}$ is a MinA for ϕ if (i) $\mathcal{M} \models \phi$ and (ii) for all $\mathcal{N} \subsetneq \mathcal{M}$, $\mathcal{N} \not\models \phi$. The task (i) is in coNP [10] while (ii) is equivalent to deciding for a polynomial number of TBoxes $\{\mathcal{N}_1, \ldots, \mathcal{N}_n\}$ (where $n = |\mathcal{M}|$) that none of these TBoxes entails ϕ. This can be verified by guessing a model for each $\mathcal{N}_i$ and verifying in polynomial time that none of these models satisfies ϕ. Thus, task (ii) is in NP, and IS-MINA is hence in D^{P}.

To prove hardness, we provide a reduction from the IS-MUS problem, which is known to be D^{P}-hard [49]. Given an inconsistent set of propositional clauses $\mathcal{F}$, a *MUS* is a minimal subset of $\mathcal{F}$ that preserves inconsistency.

Problem: IS-MUS
Input: An inconsistent set of propositional clauses $\mathcal{F}$, and $\mathcal{E} \subseteq \mathcal{F}$.
Question: Is $\mathcal{E}$ a MUS for $\mathcal{F}$?

Given an instance of IS-MUS, construct a *DL-Lite*$_{bool}$ TBox as follows. For each variable x appearing in $\mathcal{F}$, introduce a concept name B_x, and define the function ncon mapping literals to concepts as: $\mathsf{ncon}(x) = \neg B_x$; $\mathsf{ncon}(\neg x) = B_x$. For a clause $\phi = (\ell_1 \vee \ldots \vee \ell_k)$, let $t_\phi = A \sqsubseteq \neg(\mathsf{ncon}(\ell_1) \sqcap \ldots \sqcap \mathsf{ncon}(\ell_k))$ where A is a fresh concept name. Given a set of clauses $\mathcal{G}$, then define $\mathcal{T}_\mathcal{G} = \{t_\phi \mid \phi \in \mathcal{G}\}$. It is easy to see that $\mathcal{T}_\mathcal{E}$ is a MinA for $A \sqsubseteq \bot$ in $\mathcal{T}_\mathcal{F}$ iff $\mathcal{E}$ is a MUS for $\mathcal{F}$. □

We now consider two problems that try to identify the axioms that are relevant for a consequence. The first problem, called MINA-IRRELEVANCE, corresponds to checking the existence of a MinA that does not contain any of the given sets of axioms. The second one, called MINA-RELEVANCE, is its dual problem, i.e., the problem of checking the existence of a MinA that contains a given axiom. MINA-RELEVANCE is of interest in a setting where the knowledge engineer suspects an axiom for being the reason of the unwanted consequence and wants to verify whether this axiom appears in any of the MinAs. On the other hand, MINA-IRRELEVANCE is of interest if one wants to avoid certain combinations of axioms in MinAs. This might be the case, for instance, if the knowledge engineer has already identified responsible axioms for an unwanted consequence and she wants to check whether an additional MinA that does not contain these axioms exist or not.

3.2. The Existence of New MinAs

In this section, we focus on MINA-IRRELEVANCE, which is formally defined next.

Problem: MINA-IRRELEVANCE
Input: A KB $\mathcal{K}$ and an axiom φ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \varphi$, and a set $\mathscr{K} \subseteq \mathscr{P}(\mathcal{K})$.
Question: Is there a MinA $\mathcal{M}$ for φ in $\mathcal{K}$ such that $\mathcal{S} \not\subseteq \mathcal{M}$ for every $\mathcal{S} \in \mathscr{K}$?

MINA-IRRELEVANCE refers to the problem of deciding whether there is a MinA that does not contain any of the sets in $\mathscr{K}$. Intuitively, one can consider $\mathscr{K}$ as a collection of sets of axioms that are already known to be faulty. Hence, any MinA that is a superset of any element of $\mathscr{K}$ will give no further information about the causes of an erroneous consequence. In order to decide MINA-IRRELEVANCE, it does not suffice to remove the axioms that appear in one or all the sets that form $\mathscr{K}$. There can still be a MinA that has a non-empty intersection with each element of $\mathscr{K}$, but is not a superset of any of them. The most direct approach for solving MINA-IRRELEVANCE is to test for each (minimal) hitting set $\mathcal{S}$ of $\mathscr{K}$, whether there is a MinA that does not contain any of the axioms in $\mathcal{S}$.[2] However, there can be exponentially many such hitting sets in the size of $\mathscr{K}$, which means that this simple approach cannot avoid an exponential execution time in the worst case. We now show that the problem is in fact NP-complete already for core KBs.

Theorem 5. MINA-IRRELEVANCE *is* NP*-complete for core KBs.*

Proof. The problem is clearly in NP: a nondeterministic algorithm for solving it first guesses a set $\mathcal{M} \subseteq \mathcal{K}$, and then tests in polynomial time whether it is a MinA that does not contain any of the $\mathcal{S}$ in $\mathscr{K}$. For showing hardness we give a reduction from the NP-hard path with forbidden pairs problem [50].

Problem: PATH WITH FORBIDDEN PAIRS
Input: A graph $\mathcal{G} = (V, \mathcal{E})$, two vertices $s, t \in V$ and a set $\mathcal{F} \subseteq \mathcal{E} \times \mathcal{E}$.
Question: Is there a simple path P from s to t in $\mathcal{G}$ such that for every $(e, e') \in \mathcal{F}$, $\{e, e'\} \not\subseteq P$?[3]

Let an instance of PATH WITH FORBIDDEN PAIRS be given through the graph $\mathcal{G} = (V, \mathcal{E})$, $s, t \in V$ and $\mathcal{F} \subseteq \mathcal{E} \times \mathcal{E}$. We use a propositional variable p_v for every $v \in V$, and define the core KB

$$\mathcal{K} := \{p_v \to p_w \mid (v, w) \in \mathcal{E}\}.$$

Additionally we set $\varphi := p_s \to p_t$, and define

$$\mathscr{K} := \{\{p_v \to p_w, p_{v'} \to p_{w'}\} \mid ((v, w), (v', w')) \in \mathcal{F}\}.$$

It is easy to see that $\mathcal{K}, \varphi$, and $\mathscr{K}$ form an instance of MINA-IRRELEVANCE, and are built in polynomial time. We now prove that there is a MinA $\mathcal{M}$ for φ in $\mathcal{K}$ not containing any set in $\mathscr{K}$ iff there is a simple path in $\mathcal{G}$ from s to t not using any pair of edges appearing in $\mathcal{F}$.
($\Rightarrow$) Let $\mathcal{M}$ be such a MinA, and set $P_{\mathcal{M}} := \{(v, w) \mid p_v \to p_w \in \mathcal{M}\}$. It is a simple induction argument to show that $P_{\mathcal{M}}$ is a path from s to t in $\mathcal{G}$, and hence contains a simple sub-path. Moreover, as $\mathcal{M}$ does not contain any pair of axioms in $\mathscr{K}$, it follows that for every pair of edges $(e, e') \in \mathcal{F}$, $\{e, e'\} \notin P_{\mathcal{M}}$.

[2] Given a collection of sets $\mathscr{K}$, a *hitting set for* $\mathscr{K}$ is a set $\mathcal{S}$ that satisfies $\mathcal{S} \cap \mathcal{K} \neq \emptyset$ for every $\mathcal{K} \in \mathscr{K}$.

[3] The original description of the PATH WITH FORBIDDEN PAIRS problem uses pairs of vertices, rather than pairs of edges, as forbidden elements [51]. However, the variant presented here has also been shown to be NP-hard [50].

($\Leftarrow$) Let P be a path avoiding the forbidden pairs, and define the set of axioms $\mathcal{M}_P := \{p_v \to p_w \mid (v, w) \in P\}$. By construction, $\mathcal{M}_P \models p_s \to p_t$. Thus, $\mathcal{M}_P$ contains a MinA for φ. Moreover, if there is $\{p_v \to p_w, p_{v'} \to p_{w'}\} \in \mathscr{K}$ such that $\{p_v \to p_w, p_{v'} \to p_{w'}\} \subseteq \mathcal{M}_P$, then it follows that $\{(v, w), (v', w')\} \subseteq P$, which is a contradiction, since $((v, w), (v', w')) \in \mathcal{F}$. □

Recall that core KBs are special cases of all the other kinds of $\mathscr{K}$ that we have introduced. This yields the following corollary.

Corollary 6. *For all kinds of KBs introduced in Section 2,* MINA-IRRELEVANCE *is* NP*-hard. Moreover, for all of them, except for DL-Lite$_{bool}$, the problem is* NP*-complete.*

Proof. NP-hardness was shown in Theorem 5. For proving that the problem is in NP, we use the same argument as in the proof of that theorem: a nondeterministic algorithm for solving it first guesses a set $\mathcal{M} \subseteq \mathcal{K}$, then tests in polynomial time whether it is a MinA that does not contain any of the $\mathcal{S}$ in $\mathscr{K}$. □

Notice that the "in NP" argument used in this corollary does not hold for *DL-Lite$_{bool}$* KBs since testing whether a consequence follows from a *DL-Lite$_{bool}$* KB is already NP-hard [10]. Thus, the argument from the proof yields a Σ_2^{P} algorithm for deciding MINA-IRRELEVANCE in this logic.

3.3. The Case of MINA-RELEVANCE

We now consider the dual problem of MINA-IRRELEVANCE, which corresponds to checking the existence of a MinA that contains a given axiom.

Problem: MINA-RELEVANCE
Input: A KB $\mathcal{K}$ and an axiom φ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \varphi$, and an axiom $\psi \in \mathcal{K}$.
Question: Is there a MinA $\mathcal{M}$ for φ in $\mathcal{K}$ such that $\psi \in \mathcal{M}$?

If we identify a specific axiom ψ as a possible culprit for an erroneous consequence φ from a KB, MINA-RELEVANCE would allow us to decide whether ψ indeed appears in at least one MinA, and hence influences the deduction of the consequence φ from the KB. We now show that this problem is NP-complete already for core KBs.

Theorem 7. MINA-RELEVANCE *is* NP*-complete for core KBs.*

Proof. The problem is clearly in NP: a nondeterministic algorithm for solving it first guesses a subset of $\mathcal{K}$, and then tests in polynomial time whether it is a MinA containing ψ. For showing hardness we provide a reduction from the following NP-complete problem [52]:

Problem: PATH-VIA-NODE
Input: A directed graph $\mathcal{G} = (V, \mathcal{E})$ and three vertices $s, t, m \in V$
Question: Is there a simple path from s to t in $\mathcal{G}$ that passes through m?

Let $\mathcal{G}, s, t, m$ be an instance of PATH-VIA-NODE. We build an instance of MINA-RELEVANCE as follows. We introduce a propositional variable p_v for every node $v \in (V \setminus \{m\}) \cup \{m_1, m_2\}$, build the KB

$$\begin{aligned}\mathcal{K} \quad := \quad & \{p_v \to p_w \mid (v,w) \in \mathcal{E}, v, w \neq m\} \cup \\ & \{p_v \to p_{m_1} \mid (v,m) \in \mathcal{E}, v \neq m\} \cup \{p_{m_2} \to p_v \mid (m,v) \in \mathcal{E}, v \neq m\} \cup \\ & \{p_{m_1} \to p_{m_2}\},\end{aligned}$$

and set $\varphi := p_s \to p_t, \psi := p_{m_1} \to p_{m_2}$. We now show that there is a MinA for φ in $\mathcal{K}$ containing ψ iff there is a simple path in $\mathcal{G}$ from s to t that crosses through m.
($\Rightarrow$) Let $\mathcal{M}$ be a MinA for φ containing ψ. Construct the path

$$\begin{aligned}P_{\mathcal{M}} \quad := \quad & \{(v,w) \mid p_v \to p_w \in \mathcal{M}, v, w \in V\} \cup \\ & \{(v,m) \mid p_v \to p_{m_1} \in \mathcal{M}, v \in V\} \cup \{(m,v) \mid p_{m_2} \to p_v \in \mathcal{M}, v \in V\}.\end{aligned}$$

Since $\mathcal{M} \models \varphi$, it follows that $P_{\mathcal{M}}$ contains a simple path P from s to t. Suppose that $P \neq P_{\mathcal{M}}$; that is, there is an edge (v,w) in $P_{\mathcal{M}}$ that does not appear in P. As P is a path from s to t, this implies that $\mathcal{M} \setminus \{p_v \to p_w\} \models \varphi$, which contradicts the assumption that $\mathcal{M}$ is a MinA for φ. Thus, $P_{\mathcal{M}}$ is a simple path from s to t. Moreover, as $\mathcal{M}$ contains the axiom ψ, there must be an edge in $P_{\mathcal{M}}$ of the form (v,m) or (m,v). This implies that $P_{\mathcal{M}}$ passes through m.
($\Leftarrow$) Assume that there is such a simple path P. We construct the sub-KBs

$$\begin{aligned}\mathcal{M}_P \quad := \quad & \{p_v \to p_w \mid (v,w) \in P, v, w \neq m\} \cup \\ & \{p_v \to p_{m_1} \mid (v,m) \in P\} \cup \{p_{m_2} \to p_v \mid (m,v) \in P\},\end{aligned}$$

and $\mathcal{M} := \mathcal{M}_P \cup \psi$. As P is a path from s to t crossing through m, it follows that $\mathcal{M}_P \models p_s \to p_{m_1}$ and $\mathcal{M}_P \models p_{m_2} \to p_t$, and hence $\mathcal{M} \models \varphi$. Since P is a simple path, $\mathcal{M}_P \not\models \varphi$. This in particular means that every MinA for φ in $\mathcal{M}$ must contain ψ. □

As was the case for MINA-IRRELEVANCE, hardness of the core case implies hardness for all the other types of KBs we are interested in.

Corollary 8. *For all kinds of KBs introduced in Section 2,* MINA-RELEVANCE *is* NP*-hard. Moreover, for all of them, except for DL-Lite$_{bool}$, the problem is* NP*-complete.*

Proof. NP-hardness is a consequence of Theorem 7. For proving that the problem is in NP, we use the same argument as in the proof of that theorem: a nondeterministic algorithm for solving it first guesses a set $\mathcal{M} \subseteq \mathcal{K}$, then tests in polynomial time whether it is a MinA containing ψ. □

Once again, the algorithm proposed in the proof of this corollary yields a Σ_2^{p} upper bound for the case of *DL-Lite$_{bool}$* KBs. A matching lower bound for this logic can be obtained through a reduction from the Σ_2^{p}-complete problem MUS-MEMBERSHIP [53].

Problem: MUS-MEMBERSHIP
Input: An inconsistent set of propositional clauses $\mathcal{F}$ and a clause $\phi \in \mathcal{F}$.
Question: Is there a MUS $\mathcal{M}$ for $\mathcal{F}$ such that $\phi \in \mathcal{M}$?

Theorem 9. MINA-RELEVANCE *is* Σ_2^p*-complete for DL-Lite*$_{bool}^{\mathcal{N}}$ *KBs*

Proof. The upper bound was argued before. To show hardness, let $\mathcal{F}, \phi$ be an instance of MUS-MEMBERSHIP, and $\mathcal{T}_\mathcal{F}, t_\phi$ be constructed as in the proof of Theorem 4. Then, there is a MinA for $A \sqsubseteq \bot$ in $\mathcal{T}_\mathcal{F}$ that contains t_ϕ iff there is a MUS for $\mathcal{F}$ that contains ϕ. □

We now change our attention to the problem of computing *all* the MinAs for a given consequence, and analyse the complexity of enumerating them with or without a specific ordering.

4. Complexity of Enumerating All MinAs

In the previous section, we considered three decision problems related to the computation of one MinA satisfying some additional properties. To fully understand the axioms responsible for a consequence, however, it is important to find all possible MinAs. The main problem we consider in this section is thus, given a KB $\mathcal{K}$ and a consequence φ of $\mathcal{K}$, to compute all MinAs for φ in $\mathcal{K}$. This MinA enumeration problem is formally defined as follows.

Problem: MINA-ENUM
Input: A KB $\mathcal{K}$ and an axiom φ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \varphi$.
Output: The set of all MinAs for φ in $\mathcal{K}$.

Note that for core KBs, which, as we have shown in the previous section, are essentially directed graphs, a MinA is a simple path between two given vertices, and enumerating all MinAs corresponds to enumerating all simple paths between two given vertices, which can easily be done with polynomial delay [54]. However, the situation is not so clear for Horn or other more expressive types of KBs. To the best of our knowledge, only [55] considers a problem related to ours on directed hypergraphs, but it is not exactly the one considered here. We analyse first the case where the MinAs can be enumerated in any arbitrary order, and then see how the complexity is affected if a specific ordering is required, by considering lexicographic and cardinality orderings. These enumeration complexity results are summarized in Table 3.

The negative results stating that MinAs cannot be enumerated in incremental polynomial or in output polynomial time are obtained by showing hardness of some associated decision problems. Most notably, inability to enumerate all MinAs without order in output polynomial time arises from the hardness of the problem ALL-MINAS, which decides whether a given set of KBs is exactly the set of all MinAs for a given consequence (see Proposition 25). We also check whether a set of axioms is the smallest or largest MinA (w.r.t. their cardinality), or the first or last w.r.t. a lexicographic order. The complexity of these

Table 3: Complexity of enumerating all MinAs. PD stands for polynomial delay, OP for output polynomial (but not incremental polynomial), and NIP for not incremental polynomial.

		lexicographic		size	
type	unordered	forward	backward	increasing	decreasing
core, $DL\text{-}Lite^{[\mathcal{N}]}_{[core\|krom]}$	PD [Cor. 24]	OP [Cor. 33]	PD [Thm. 35]	PD [Thm. 40]	OP [Cor. 37]
Horn, $DL\text{-}Lite^{[\mathcal{N}]}_{horn}$	PD [Thm. 17]	OP [Cor. 33]	PD [Thm. 35]	OP [Prop. 38]	OP [Cor. 37]
dual-Horn	TE-h [Cor. 27]	NIP [Cor. 33]	TE-h [Cor. 27]	TE-h [Cor. 27]	NIP [Cor. 37]
(Horn-)$\mathcal{EL}$, $DL\text{-}Lite^{[\mathcal{N}]}_{bool}$	not ouput polynomial [Cors. 29, 31]				

Table 4: Complexity of decision problems associated with enumeration.

type	ALL-MINAS	FIRST-MINA	LAST-MINA	LARGEST-MINA	SMALLEST-MINA
core, $DL\text{-}Lite^{[\mathcal{N}]}_{[core\|krom]}$	P [Prop. 25]	coNP-c [Thm. 32]	P	NP-c [Thm. 36]	P [Thm. 39]
Horn, $DL\text{-}Lite^{[\mathcal{N}]}_{horn}$	P [Prop. 25]	coNP-c [Thm. 32]	P	NP-c [Thm. 36]	NP-c [37]
dual-Horn	TH-h [Thm. 26]	coNP-c [Thm. 32]	P	NP-c [Thm. 36]	P
Horn-$\mathcal{EL}$, $\mathcal{EL}$	coNP-c [Thm. 28]	coNP-c [Thm. 32]	P	NP-c [Thm. 36]	NP-c [37]
$DL\text{-}Lite^{[\mathcal{N}]}_{bool}$	D^{p}-h [Lem. 30]	D^{p}-h [Thm. 34]	D^{p}-c	D^{p}-h	Σ_2^{p}-c [Thm. 41]

decision problems, which are also of interest on their own, is summarized in Table 4. In the table, results lacking a reference to where they can be found are consequences of previous results, and explained in the text.

4.1. Enumeration without a Specific Order

We first consider the $DL\text{-}Lite^{\mathcal{N}}_{horn}$ setting, and show that for this kind of KBs MinAs can be efficiently enumerated through a polynomial delay algorithm. To achieve this goal, we start by constructing for every $DL\text{-}Lite^{\mathcal{N}}_{horn}$ TBox $\mathcal{T}$ a propositional Horn KB $\mathcal{G}_{\mathcal{T}}$ as follows: for every basic concept B create a propositional variable p_B; for every axiom $\bigsqcap_{i=1}^{n} B_i \sqsubseteq B$ add the Horn clause $\bigwedge_{i=1}^{n} p_{B_i} \rightarrow p_B$; and for each pair of number restrictions $\geq q_1 r, \geq q_2 r$ with $q_1 > q_2$ appearing in $\mathcal{T}$, add the Horn clause $p_{\geq q_1 r} \rightarrow p_{\geq q_2 r}$. We will call these latter clauses *implicit axioms*. For simplicity, in the following we will assume that $\bot$ does not appear in $\mathcal{T}$. For handling the case where $\bot$ appears in $\mathcal{T}$, one only needs to add the implicit axiom $p_\bot \rightarrow p_C$ to $\mathcal{G}_{\mathcal{T}}$ whenever we are interested in deciding a subsumption of the form $\bigsqcap_{i=1}^{n} A_i \sqsubseteq C$

It is not difficult to see that $\mathcal{T} \models \bigsqcap_{i=1}^{n} A_i \sqsubseteq C$ iff $\mathcal{G}_{\mathcal{T}} \models \bigwedge_{i=1}^{n} p_{A_i} \rightarrow p_C$. Furthermore, every MinA $\mathcal{M}$ in $\mathcal{G}_{\mathcal{T}}$ gives rise to a MinA in $\mathcal{T}$ consisting of all axioms representing non implicit axioms in $\mathcal{M}$. However, different MinAs in $\mathcal{G}_{\mathcal{T}}$ can give rise to the same MinA in $\mathcal{T}$, as shown in the following example.

Example 10. Let $\mathcal{T} = \{A \sqsubseteq\ \geq 2r, A \sqsubseteq\ \geq 3r, \geq 1r \sqsubseteq B\}$. Using the construction described above, we obtain

$$\begin{aligned} \mathcal{G}_{\mathcal{T}} \quad = \quad & \{p_A \rightarrow p_{\geq 2r}, \quad p_A \rightarrow p_{\geq 3r}, \quad p_{\geq 1r} \rightarrow p_B, \\ & p_{\geq 3r} \rightarrow p_{\geq 1r}, \quad p_{\geq 3r} \rightarrow p_{\geq 2r}, \quad p_{\geq 2r} \rightarrow p_{\geq 1r}\}, \end{aligned}$$

where the implicit axioms are those appearing in the second row. It is easy to see that there are two MinAs for $A \sqsubseteq B$ in $\mathcal{T}$, namely $\{A \sqsubseteq\ \geq 3r, \geq 1r \sqsubseteq B\}$ and $\{A \sqsubseteq\ \geq 2r, \geq 1r \sqsubseteq B\}$. However, $\mathcal{G}_\mathcal{T}$ contains three MinAs for $p_A \rightarrow p_B$. The reason for this superfluous MinA is that the implicit subsumption $\geq 3r \sqsubseteq\ \geq 1r$ is represented twice in $\mathcal{G}_\mathcal{T}$: one through the direct edge $p_{\geq 3r} \rightarrow p_{\geq 1r}$, and another with a path travelling along $p_{\geq 2r}$. This yields two different MinAs in $\mathcal{G}_\mathcal{T}$ for the MinA $\{A \sqsubseteq\ \geq 3r, \geq 1r \sqsubseteq B\}$ in $\mathcal{T}$.

As this example shows, the transformation to Horn clauses may introduce some artificial MinAs in the constructed Horn KB $\mathcal{G}_\mathcal{T}$ that must not be confused with MinAs of the original KB $\mathcal{T}$. To solve this problem, we choose a representative for all the MinAs in $\mathcal{G}_\mathcal{T}$ that correspond to the same MinA in $\mathcal{T}$. We define this representative with the help of a lexicographic ordering.

Definition 11 (Lexicographic Order)**.** Let the elements of a set S be linearly ordered. This order induces a linear strict order on $\mathscr{P}(S)$, called the *lexicographic order*, as follows. A set $R \subseteq S$ is *lexicographically smaller* than a set $T \subseteq S$ if $R = T$ or the first element at which they disagree is in R.

For example, if $S = \{x, y, z\}$ with $x < y < z$, then $\{x, y\}$ is lexicographically smaller than $\{x, z\}$, which is itself lexicographically smaller than $\{y\}$. Consider now an arbitrary but fixed total ordering on the set of implicit axioms $\mathcal{I}_\mathcal{T}$ appearing in $\mathcal{G}_\mathcal{T}$. When two or more MinAs in $\mathcal{G}_\mathcal{T}$ agree on all non-implicit axioms, then we choose only that one that is the lexicographically largest. This is what we call an immediate MinA.

Definition 12 (Immediate MinA)**.** Let $\mathcal{T}$ be a $\mathit{DL\text{-}Lite}^{\mathcal{N}}_{horn}$ TBox and $\mathcal{I}_\mathcal{T}$ be the set of implicit axioms obtained from $\mathcal{T}$ using the construction described above. A MinA $\mathcal{M}$ in $\mathcal{G}_\mathcal{T}$ is called *immediate* if for every $\mathcal{J} \subseteq \mathcal{I}_\mathcal{T}$ the following holds: if $(\mathcal{M} \setminus \mathcal{I}_\mathcal{T}) \cup \mathcal{J}$ is a MinA in $\mathcal{G}_\mathcal{T}$, then $\mathcal{J}$ is lexicographically smaller than $\mathcal{M} \cap \mathcal{I}_\mathcal{T}$.

For the TBox $\mathcal{T}$ from Example 10, suppose that the implicit axioms are ordered as follows:

$$p_{\geq 2r} \rightarrow p_{\geq 1r} \ < \ p_{\geq 3r} \rightarrow p_{\geq 1r} \ < \ p_{\geq 3r} \rightarrow p_{\geq 2r}.$$

The MinA $\mathcal{M} = \{p_A \rightarrow p_{\geq 3r},\ p_{\geq 3r} \rightarrow p_{\geq 2r},\ p_{\geq 2r} \rightarrow p_{\geq 1r},\ p_{\geq 1r} \rightarrow p_B\}$ is *not* immediate, since the set of implicit axioms $\mathcal{J} = \{p_{\geq 3r} \rightarrow p_{\geq 1r}\}$ is lexicographically larger than $\mathcal{M} \cap \mathcal{I}_\mathcal{T} = \{p_{\geq 3r} \rightarrow p_{\geq 2r},\ p_{\geq 2r} \rightarrow p_{\geq 1r}\}$ according to this ordering, and $(\mathcal{M} \setminus \mathcal{I}_\mathcal{T}) \cup \mathcal{J} = \{p_A \rightarrow p_{\geq 3r},\ p_{\geq 3r} \rightarrow p_{\geq 1r},\ p_{\geq 1r} \rightarrow p_B\}$ is also a MinA.

In fact, there exists a bijection between MinAs for $\bigsqcap_{i=1}^n A_i \sqsubseteq C$ in $\mathcal{T}$ and immediate MinAs for $\bigwedge_{i=1}^n p_{A_i} \rightarrow p_C$ in $\mathcal{G}_\mathcal{T}$: from an immediate MinA in $\mathcal{G}_\mathcal{T}$ we obtain a MinA in $\mathcal{T}$ by removing all implicit axioms; dually, from a MinA in $\mathcal{T}$ we can build an immediate MinA in $\mathcal{G}_\mathcal{T}$ by adding the lexicographically largest set of implicit axioms that forms a MinA. Thus, if we can enumerate all immediate MinAs in $\mathcal{G}_\mathcal{T}$ in output polynomial time, we will also be able to enumerate all MinAs in $\mathcal{T}$ within the same complexity bound. We now show how to compute all immediate MinAs using the notion of a *valid ordering* on the axioms in a Horn KB.

Definition 13 (Valid Ordering)**.** Let $\mathcal{T}$ be a propositional Horn KB. Given a Horn axiom $\phi = \bigwedge_{i=1}^{n} a_i \to b$, we denote the left-handside (lhs) of ϕ with $\mathsf{T}(\phi)$, and its right-handside (rhs) with $\mathsf{h}(\phi)$, i.e., $\mathsf{T}(\phi) := \{a_1, \ldots, a_n\}$ and $\mathsf{h}(\phi) := b$. With $\mathsf{h}^{-1}(b)$ we denote the set of axioms in $\mathcal{T}$ whose rhs are b.

Let $\mathcal{M} = \{t_1, \ldots, t_m\}$ be a MinA for $\bigwedge_{a \in A} a \to c$. An ordering $t_1 < \ldots < t_m$ is *valid* on $\mathcal{M}$ if for every $1 \leq i \leq m$, $\mathsf{T}(t_i) \subseteq A \cup \{\mathsf{h}(t_1), \ldots, \mathsf{h}(t_{i-1})\}$ holds.[4]

Intuitively, a valid ordering describes the steps that need to be made to deduce the atom b from the set of atoms A using only the clauses in the MinA. It is easy to see that for every immediate MinA there is always at least one such valid ordering. In the following, we use this fact to construct a set of sub-KBs that contain all and only the remaining immediate MinAs. Our approach is based on the ideas originally presented in [56].

Definition 14 ($\mathcal{T}_i$)**.** Let $\mathcal{M}$ be an immediate MinA in $\mathcal{G}_\mathcal{T}$ with $|\mathcal{M}| = m$, and $<$ be a valid ordering on $\mathcal{M}$. For each $1 \leq i \leq m$ we obtain a KB $\mathcal{T}_i$ from $\mathcal{G}_\mathcal{T}$ as follows: if t_i is an implicit axiom, then $\mathcal{T}_i = \emptyset$; otherwise, (i) for each j s.t. $i < j \leq m$ and t_j is not an implicit axiom, remove all axioms in $\mathsf{h}^{-1}(\mathsf{h}(t_j))$ except for t_j, i.e., remove all axioms with the same rhs as t_j except for t_j itself, (ii) remove t_i, and (iii) add all implicit axioms.

Example 15. Consider again the KBs $\mathcal{T}$ and $\mathcal{G}_\mathcal{T}$ from Example 10, where the implicit axioms $\mathcal{I}_\mathcal{T}$ are ordered as

$$p_{\geq 2r} \to p_{\geq 1r} \;<\; p_{\geq 3r} \to p_{\geq 1r} \;<\; p_{\geq 3r} \to p_{\geq 2r}.$$

Then, $\mathcal{M} = \{p_A \to p_{\geq 3r},\ p_{\geq 3r} \to p_{\geq 1r},\ p_{\geq 1r} \to p_B\}$ is an immediate MinA with three elements. The only possible valid ordering for this MinA is

$$t_1 := p_A \to p_{\geq 3r}, \quad t_2 := p_{\geq 3r} \to p_{\geq 1r}, \quad t_3 := p_{\geq 1r} \to p_B.$$

Notice that t_2 is in fact an implicit axiom. Thus, the construction from Definition 14 yields the three Horn KBs

$$\begin{aligned} \mathcal{T}_1 &:= \{p_A \to p_{\geq 2r}, p_{\geq 1r} \to p_B\} \cup \mathcal{I}_\mathcal{T} \\ \mathcal{T}_2 &:= \emptyset \\ \mathcal{T}_3 &:= \{p_A \to p_{\geq 2r}, p_A \to p_{\geq 3r}\} \cup \mathcal{I}_\mathcal{T}. \end{aligned}$$

The naïve method for computing one MinA sketched in Section 2.4 (see Algorithm 1) can be easily adapted to the computation of an immediate MinA in polynomial time by specifying an ordering in which the axioms are selected

[4] That is, each variable on the lhs of t_i is in A, or it is the rhs of a previous axiom.

for the execution of the **for**-loop: first considering all non-implicit axioms, and afterwards the $\mathcal{I}_\mathcal{T}$, i.e. the implicit axioms, according to the fixed ordering. By first testing all non-implicit axioms, one minimizes the set of axioms of this kind that will remain in the computed MinA. At that point, one needs only to remove the superfluous implicit axioms to reach a MinA. Analyzing these axioms in order guarantees that the set of implicit axioms kept is the lexicographically largest such set. Hence, the computation yields an immediate MinA.

Once that we have found an immediate MinA, Definition 14 introduces a class of sub-KBs $\mathcal{T}_i, 1 \leq i \leq m$ of $\mathcal{G}_\mathcal{T}$. We show that this class induces a partition on the set of all other MinAs in $\mathcal{G}_\mathcal{T}$.

Lemma 16. *Let $\mathcal{M}$ be an immediate MinA for ϕ in $\mathcal{T}$, and let $\mathcal{T}_1, \ldots, \mathcal{T}_m$ be constructed from $\mathcal{T}$ and $\mathcal{M}$ as in Definition 14. Then, for every immediate MinA $\mathcal{N}$ for ϕ in $\mathcal{T}$, if $\mathcal{N} \neq \mathcal{M}$, then there exists* **exactly one** *i, $1 \leq i \leq m$, such that $\mathcal{N}$ is a MinA for ϕ in $\mathcal{T}_i$.*

Proof. Let $t_1 < \ldots < t_m$ be a valid ordering on $\mathcal{M}$, and $\mathcal{N}$ an immediate MinA for ϕ in $\mathcal{T}$ such that $\mathcal{N} \neq \mathcal{M}$. Then, $\mathcal{M} \setminus \mathcal{N} \neq \emptyset$. Let t_k be the largest non-implicit axiom in $\mathcal{M} \setminus \mathcal{N}$ w.r.t. the valid ordering $<$. We show that $\mathcal{N} \subseteq \mathcal{T}_k$ and $\mathcal{N} \not\subseteq \mathcal{T}_i$ for all $i \neq k$, $1 \leq i \leq m$.

Assume first that there is an axiom $t \in \mathcal{N}$ s.t. $t \notin \mathcal{T}_k$. Since $\mathcal{T}_k$ contains all implicit axioms, t should be one of the non-implicit axioms removed from $\mathcal{T}$ either in step (i) or in step (ii) of Definition 14. If it was in step (ii), then $t = t_k$, but by construction, $t_k \in \mathcal{M} \setminus \mathcal{N}$ and hence $t = t_k \notin \mathcal{N}$, which is a contradiction. Thus, it should have been removed by step (i). This implies that there exists a j, $k < j \leq m$, such that t_j satisfies $\mathsf{h}(t) = \mathsf{h}(t_j)$. Recall that we chose k to be the largest axiom in $\mathcal{M} \setminus \mathcal{N}$ w.r.t. the valid ordering $<$ on $\mathcal{M}$. Then this t_j should be in $\mathcal{N}$. But then $\mathcal{N}$ contains two axioms with the rhs $\mathsf{h}(t)$, which contradicts the fact that $\mathcal{N}$ is a MinA, and thus it is minimal. Hence, $\mathcal{N} \subseteq \mathcal{T}_k$.

Now take an i s.t. $i \neq k$. If $i > k$, then $t_i \in \mathcal{N}$ but $t_i \notin \mathcal{T}_i$, and hence $\mathcal{N} \not\subseteq \mathcal{T}_i$. If $i < k$, then there is an axiom $t \in \mathcal{N}$ such that $\mathsf{h}(t) = \mathsf{h}(t_k)$ since otherwise $\mathcal{M}$ and $\mathcal{N}$ would not be MinAs. By construction, $t \notin \mathcal{T}_i$, hence $\mathcal{N} \not\subseteq \mathcal{T}_i$. □

Returning to Example 15, the only remaining immediate MinA for $p_A \to p_B$ is contained in $\mathcal{T}_3$. Lemma 16 provides a strategy for computing all the remaining MinAs, starting from a known one, in the *DL-Lite*$^\mathcal{N}_{horn}$ setting. Algorithm 2 describes how we can use this lemma to enumerate all MinAs in a *DL-Lite*$^\mathcal{N}_{horn}$ TBox $\mathcal{T}$ by enumerating all immediate MinAs in $\mathcal{G}_\mathcal{T}$.

Theorem 17. *Algorithm 2 solves* MINA-ENUM *for DL-Lite$^\mathcal{N}_{horn}$ TBoxes with polynomial delay.*

Proof. The algorithm terminates since $\mathcal{T}$ is finite. It is sound since its outputs are all MinAs for ϕ in $\mathcal{T}$. By Lemma 16, every MinA that has not been computed is contained in exactly one of the KBs $\mathcal{T}_i$ constructed, and will hence be computed in the recursive call using $\mathcal{T}_i$ as input. This implies that the algorithm is complete.

Algorithm 2 Enumerating all MinAs for $DL\text{-}Lite^{\mathcal{N}}_{horn}$ TBoxes

```
Procedure ALL-MINAS($\mathcal{T}$,$\phi$)        ($\mathcal{T}$ $DL\text{-}Lite^{\mathcal{N}}_{horn}$ TBox, $\phi$ axiom s.t. $\mathcal{T} \models \phi$)
1: if $\mathcal{T} \not\models \phi$ then return
2: else
3:     $\mathcal{M} \leftarrow$ an immediate MinA in $\mathcal{G}_{\mathcal{T}}$
4:     $\mathcal{I} \leftarrow \{t \in \mathcal{M} \mid t$ is an implicit axiom$\}$
5:     output $\mathcal{M} \setminus \mathcal{I}$
6:     for $1 \leq i \leq |\mathcal{M}|$ do
7:         compute $\mathcal{T}_i$ from $\mathcal{M}$ as in Definition 14
8:         ALL-MINAS($\mathcal{T}_i \setminus \mathcal{I}$,$\phi$)
```

We now analyse the time required to output each successive answer. Every time the procedure ALL-MINAS is called, it either concludes that the consequence does not follow from the KB (in polynomial time) and stops, or computes one immediate MinA, also in polynomial time. In the latter case, it makes linearly many recursive calls to ALL-MINAS via the sub-KBs $\mathcal{T}_i$. By Lemma 16, any other immediate MinA appears in exactly one of these $\mathcal{T}_i$s. Thus, after linearly many calls, the algorithm either finds out that no more MinAs exist, or outputs a new MinA. More precisely, in each recursive call of the algorithm there is one consequence check (line 1), and one MinA computation (line 3). The consequence check can be done in polynomial time [10]. One MinA is computed in polynomial time using Algorithm 1. Thus the algorithm spends at most polynomial time between each output, i.e., it is polynomial delay. □

This theorem shows that MinAs can be enumerated in polynomial delay for several classes of KBs considered in this paper.

Corollary 18. *All MinAs can be enumerated with polynomial delay for core, Horn, $DL\text{-}Lite_{horn}$ and $DL\text{-}Lite^{\mathcal{N}}_{horn}$ KBs.*

However, this corollary does not apply to $DL\text{-}Lite_{core}$ and $DL\text{-}Lite_{krom}$ KBs. The main reason for this is the presence of negation in the axioms. We will adapt the previous reduction to Horn KBs by abstracting away from the negations and considering each concept $\neg B$ as *atomic*. However, this requires some technical modifications in the construction of the Horn KB. Observe first that the axiom $B_1 \sqsubseteq \neg B_2$ is in fact equivalent to $B_2 \sqsubseteq \neg B_1$. In addition, notice that a subsumption $B_1 \sqsubseteq B_2$ may follow from B_1 being unsatisfiable (i.e., $B_1^{\mathcal{I}} = \emptyset$ for all models of the KB), which is the case iff $\mathcal{K} \models B_1 \sqsubseteq \neg B_1$.

Given a $DL\text{-}Lite^{\mathcal{N}}_{krom}$ TBox $\mathcal{T}$, we construct a core KB as follows: for every basic concept B create two propositional variables p_B and $p_{\neg B}$; for every axiom $t \in \mathcal{T}$ define a pair of core clauses $\mathcal{C}_t$ as follows:

$$\begin{aligned}
\mathcal{C}_{B_1 \sqsubseteq B_2} &:= \{p_{B_1} \rightarrow p_{B_2}, p_{\neg B_2} \rightarrow p_{\neg B_1}\}, \\
\mathcal{C}_{B_1 \sqsubseteq \neg B_2} &:= \{p_{B_1} \rightarrow p_{\neg B_2}, p_{B_2} \rightarrow p_{\neg B_1}\}, \text{ and} \\
\mathcal{C}_{\neg B_1 \sqsubseteq B_2} &:= \{p_{\neg B_1} \rightarrow p_{B_2}, p_{\neg B_2} \rightarrow p_{B_1}\};
\end{aligned}$$

and for each pair of number restrictions $\geq q_1 r, \geq q_2 r$ with $q_1 > q_2$ appearing in $\mathcal{T}$, define the pair $\mathcal{C}_{q_1,q_2} := \{p_{\geq q_1 r} \to p_{\geq q_2 r}, p_{\geq \neg q_2 r} \to p_{\geq \neg q_1 r}\}$. As before, the latter clauses are called *implicit*, and $\mathcal{I}_\mathcal{T}$ denotes the set of all implicit axioms. Given a $\textit{DL-Lite}^{\mathcal{N}}_{krom}$ axiom $\phi = B_1 \sqsubseteq B_2$, define the core KBs

$$\mathcal{H}^{\phi}_{\mathcal{T}} := \bigcup_{t \in \mathcal{T}} \mathcal{C}_t \cup \mathcal{I}_\mathcal{T} \cup \{p_{B_2} \to p_{\neg B_1}\}.$$

One can see that $\mathcal{T} \models \phi$ iff $\mathcal{H}^{\phi}_{\mathcal{T}} \models p_{B_1} \to p_{\neg B_1}$. However, contrary to the case of $\textit{DL-Lite}^{\mathcal{N}}_{horn}$ TBoxes, the MinAs in $\mathcal{H}^{\phi}_{\mathcal{T}}$ do not necessarily correspond to MinAs in $\mathcal{T}$.

Example 19. Let $\mathcal{T} = \{A \sqsubseteq B_1, A \sqsubseteq B_2, B_1 \sqsubseteq C, B_2 \sqsubseteq C, C \sqsubseteq \neg C\}$ and $\phi = A \sqsubseteq D$. Using the previous construction, we obtain

$$\begin{aligned}\mathcal{H}^{\phi}_{\mathcal{T}} = \{&p_A \to p_{B_1},\ p_A \to p_{B_2},\ p_{B_1} \to p_C,\ p_{B_2} \to p_C,\ p_C \to p_{\neg C}\\ &p_{\neg B_1} \to p_{\neg A},\ p_{\neg B_2} \to p_{\neg A},\ p_{\neg C} \to p_{\neg B_1},\ p_{\neg C} \to p_{\neg B_2},\ p_D \to p_{\neg A}\}.\end{aligned}$$

Clearly, $\mathcal{M} = \{p_A \to p_{B_1}, p_{B_1} \to p_C, p_C \to p_{\neg C}, p_{\neg C} \to p_{\neg B_2}, p_{\neg B_2} \to p_{\neg A}\}$ is a MinA for $p_A \to p_{\neg A}$ in $\mathcal{H}^{\phi}_{\mathcal{T}}$. However, this set represents all the axioms in $\mathcal{T}$ and thus, contains superfluous axioms for the consequence.

In order to avoid this behaviour, we consider a different kind of MinA, in which the two clauses corresponding to the same axiom are always required to appear together.

Definition 20 (Paired MinA)**.** Consider a $\textit{DL-Lite}^{\mathcal{N}}_{krom}$ TBox $\mathcal{T}$ and an axiom $\phi = B_1 \sqsubseteq B_2$. We call a subset $\mathcal{M} \subseteq \mathcal{H}^{\phi}_{\mathcal{T}}$ a *dualized set* if for every $t \in \mathcal{T}$ either (i) $\mathcal{C}_t \subseteq \mathcal{M}$, or (ii) $\mathcal{C}_t \cap \mathcal{M} = \emptyset$ holds. The dualized set $\mathcal{M}$ is a *paired MinA* if $\mathcal{M} \models p_{B_1} \to p_{\neg B_1}$ and for every $t \in \mathcal{T}$, if $\mathcal{C}_t \subseteq \mathcal{M}$, then $\mathcal{M} \setminus \mathcal{C}_t \not\models p_{B_1} \to p_{\neg B_1}$.

Notice that a paired MinA is not a MinA in the strict sense since it is not necessarily minimal. For instance for the TBox in Example 19 and $\phi = A \sqsubseteq B_1$, $\{p_A \to p_{B_1}, p_{\neg B_1} \to p_{\neg A}, p_{B_1} \to p_{\neg A}\}$ is a paired MinA, even though the second axiom is superfluous for the consequence. However, every paired MinA $\mathcal{M}$ in $\mathcal{H}^{\phi}_{\mathcal{T}}$ corresponds to a MinA in $\mathcal{T}$, defined by the set of axioms t such that $\mathcal{C}_t \subseteq \mathcal{M}$. To improve readability, in the following we will often disregard the implicit axioms in $\mathcal{H}^{\phi}_{\mathcal{T}}$. They can be treated analogously as in the $\textit{DL-Lite}^{\mathcal{N}}_{horn}$ case, by fixing an ordering on them and considering only the lexicographically largest subset of $\mathcal{I}_\mathcal{T}$ needed.

One paired MinA can be computed in polynomial time through a slight variant of Algorithm 1. To ensure that the resulting set of axioms is paired, the **for** loop is changed to try to remove a set $\mathcal{C}_t$ in a single iteration, rather than one axiom at a time. As before, we will use one paired MinA to build a set of sub-KBs that partitions the class of all remaining paired MinAs in $\mathcal{H}^{\phi}_{\mathcal{T}}$. Given a paired MinA $\mathcal{M}$, it is always possible to choose a representative $\overline{\alpha}$ of each $\mathcal{C}_\alpha$, where α can either be a $t \in \mathcal{T}$ or denote an implicit axiom, and order these representatives as $\overline{t_1} < \ldots < \overline{t_m}$ such that $\mathsf{T}(\overline{t_1}) = p_{B_1}$ and for all $i, 1 < i \leq m$ $\mathsf{T}(\overline{t_i}) = \mathsf{h}(\overline{t_{i-1}})$. Abusing the terminology, we will call this a *valid ordering*. The construction of the KBs $\mathcal{T}_i$ is very similar to Definition 14, but considering this new notion of valid ordering, and the axiom derived from ϕ.

Definition 21 ($\mathcal{T}_i$)**.** Let $\mathcal{M}$ be a paired MinA in $\mathcal{H}^{\phi}_{\mathcal{T}}$ and $<$ be a valid ordering on $\mathcal{M}$ of length m. For each $1 \leq i \leq m$ we obtain a KB $\mathcal{T}_i$ from $\mathcal{H}^{\phi}_{\mathcal{T}}$ as follows: if t_i is an implicit axiom or $p_{B_2} \to p_{\neg B_1}$, then $\mathcal{T}_i = \emptyset$; otherwise, (i) for each j s.t. $i < j \leq m$ and t_j is not an implicit axiom or $p_{B_2} \to p_{\neg B_1}$, remove all axioms in $\mathsf{h}^{-1}(\mathsf{h}(t_j))$ except for t_j, i.e., remove all axioms with the same rhs as t_j except for t_j itself, (ii) remove t_i, and (iii) add all implicit axioms and $p_{B_2} \to p_{\neg B_1}$.

Example 22. Consider again the KBs $\mathcal{T}$ and $\mathcal{H}^{\phi}_{\mathcal{T}}$ from Example 19. Then $\mathcal{M} = \{p_A \to p_{B_1}, p_{B_1} \to p_C, p_C \to p_{\neg C}, p_{\neg C} \to p_{\neg B_1}, p_{\neg B_1} \to p_{\neg A}\}$ is a paired MinA, where the last two elements are the duals for the first two. The only possible valid ordering for this paired MinA is

$$p_A \to p_{B_1} \ < \ p_{B_1} \to p_C \ < \ p_C \to p_{\neg C}.$$

The construction from Definition 21 yields the following three KBs:

$$\begin{aligned}
\mathcal{T}_1 = \{&p_A \to p_{B_2},\ p_{B_1} \to p_C,\ p_C \to p_{\neg C}\\
&p_{\neg B_1} \to p_{\neg A},\ p_{\neg B_2} \to p_{\neg A},\ p_{\neg C} \to p_{\neg B_1},\ p_{\neg C} \to p_{\neg B_2},\ p_D \to p_{\neg A}\},\\
\mathcal{T}_2 = \{&p_A \to p_{B_1},\ p_A \to p_{B_2},\ p_{B_2} \to p_C,\ p_C \to p_{\neg C}\\
&p_{\neg B_1} \to p_{\neg A},\ p_{\neg B_2} \to p_{\neg A},\ p_{\neg C} \to p_{\neg B_1},\ p_{\neg C} \to p_{\neg B_2},\ p_D \to p_{\neg A}\},\\
\mathcal{T}_3 = \{&p_A \to p_{B_1},\ p_A \to p_{B_2},\ p_{B_1} \to p_C,\ p_{B_2} \to p_C,\\
&p_{\neg B_1} \to p_{\neg A},\ p_{\neg B_2} \to p_{\neg A},\ p_{\neg C} \to p_{\neg B_1},\ p_{\neg C} \to p_{\neg B_2},\ p_D \to p_{\neg A}\}.
\end{aligned}$$

The proof of the following lemma is very similar to that of Lemma 16.

Lemma 23. *Let $\mathcal{M}$ be a paired MinA for $p_{B_2} \to p_{\neg B_1}$ in $\mathcal{H}^{\phi}_{\mathcal{T}}$, and let $\mathcal{T}_1, \ldots, \mathcal{T}_m$ be constructed from $\mathcal{H}^{\phi}_{\mathcal{T}}$ and $\mathcal{M}$ as in Definition 21. Then, for every paired MinA $\mathcal{N}$ for $p_{B_2} \to p_{\neg B_1}$ in $\mathcal{H}^{\phi}_{\mathcal{T}}$, if $\mathcal{N} \neq \mathcal{M}$, then there exists* **exactly one** *i, $1 \leq i \leq m$, such that $\mathcal{N}$ is a paired MinA for $p_{B_2} \to p_{\neg B_1}$ in $\mathcal{T}_i$.*

Proof. Let $t_1 < \ldots < t_m$ be a valid ordering on $\mathcal{M}$, and $\mathcal{N}$ a paired MinA for $p_{B_2} \to p_{\neg B_1}$ in $\mathcal{H}^{\phi}_{\mathcal{T}}$ such that $\mathcal{N} \neq \mathcal{M}$. Then, $\mathcal{M} \setminus \mathcal{N} \neq \emptyset$. Let t_k be the largest non-implicit axiom in $\mathcal{M} \setminus \mathcal{N}$ w.r.t. the valid ordering $<$. We show that $\mathcal{N} \subseteq \mathcal{T}_k$ and $\mathcal{N} \not\subseteq \mathcal{T}_i$ for all $i \neq k$, $1 \leq i \leq m$.

Assume first that there is an axiom $t \in \mathcal{N}$ s.t. $t \notin \mathcal{T}_k$. Since $\mathcal{T}_k$ contains all implicit axioms, t should be one of the non-implicit axioms removed from $\mathcal{T}$ either in step (i) or in step (ii) of Definition 21. If it was in step (ii), then $t = t_k$, but by construction, $t_k \in \mathcal{M} \setminus \mathcal{N}$ and hence $t = t_k \notin \mathcal{N}$, which is a contradiction. Thus, it should have been removed by step (i). This implies that there exists a j, $k < j \leq m$, such that t_j satisfies $\mathsf{h}(t) = \mathsf{h}(t_j)$. Recall that we chose k to be the largest axiom in $\mathcal{M} \setminus \mathcal{N}$ w.r.t. the valid ordering $<$ on $\mathcal{M}$.

Then this t_j should be in $\mathcal{N}$. But then $\mathcal{N}$ contains two different axioms with the rhs $\mathsf{h}(t)$. These two axioms cannot belong to the same set $\mathcal{C}_\alpha$. But then, $\mathcal{N}$ cannot be a paired MinA, as it contradicts the minimality criterion. Hence, $\mathcal{N} \subseteq \mathcal{T}_k$.

Now take an i s.t. $i \neq k$. If $i > k$, then $t_i \in \mathcal{N}$ but $t_i \notin \mathcal{T}_i$, and hence $\mathcal{N} \not\subseteq \mathcal{T}_i$. If $i < k$, then there is an axiom $t \in \mathcal{N}$ such that $\mathsf{h}(t) = \mathsf{h}(t_k)$ since otherwise $\mathcal{M}$ and $\mathcal{N}$ would not be paired MinAs. By construction, $t \notin \mathcal{T}_i$, hence $\mathcal{N} \not\subseteq \mathcal{T}_i$. □

Using this result, and Algorithm 2 it follows directly that all the paired MinAs in $\mathcal{H}^{\phi}_{\mathcal{T}}$ can be enumerated with polynomial delay. In particular, this means that MinAs for $DL\text{-}Lite^{\mathcal{N}}_{krom}$ and $DL\text{-}Lite^{\mathcal{N}}_{core}$ KBs are also polynomial delay enumerable.

Corollary 24. *All MinAs can be enumerated with polynomial delay for KBs in $DL\text{-}Lite_{core}$, $DL\text{-}Lite^{\mathcal{N}}_{core}$, $DL\text{-}Lite_{krom}$ and $DL\text{-}Lite^{\mathcal{N}}_{krom}$.*

We now consider MINA-ENUM for dual-Horn KBs. For this, we first investigate the following decision problem which is closely related to MINA-ENUM. As we will see, determining its complexity is important for determining the complexity of MINA-ENUM.

Problem: ALL-MINAS
Input: A KB $\mathcal{K}$ and an axiom φ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \varphi$, and a set of KBs $\mathscr{K} \subseteq \mathscr{P}(\mathcal{K})$.
Question: Is $\mathscr{K}$ precisely the set of all MinAs for φ in $\mathcal{K}$?

As Proposition 25 below shows, if ALL-MINAS cannot be decided in polynomial time, then MINA-ENUM cannot be solved in output polynomial time. The proof of this fact is based on a generic argument, which can also be found in Theorem 4.5 of [57], but for the sake of completeness and clarity we present it here once more.

Proposition 25. *If* ALL-MINAS *cannot be decided in polynomial time, then unless* P = NP, MINA-ENUM *cannot be solved in output-polynomial time.*

Proof. Assume we have an algorithm A that solves MINA-ENUM in output-polynomial time. Let its runtime be bounded by a polynomial $p(IS, OS)$ where IS denotes the size of the input KB and OS denotes the size of the output, i.e., the set of all MinAs.

In order to decide ALL-MINAS for an instance given by $\mathcal{K}$, φ, and $\mathscr{K} \subseteq \mathscr{P}(\mathcal{K})$, we construct another algorithm A' that works as follows: it runs A on $\mathcal{K}$ and φ for at most $p(|\mathcal{K}|, |\mathscr{K}|)$-many steps. If A terminates within this many steps, then A' compares the output of A with $\mathscr{K}$ and returns *yes* if and only if they are equal. If they are not equal, A' returns *no*. If A has not yet terminated after $p(|\mathcal{K}|, |\mathscr{K}|)$-many steps, this implies that there is at least one MinA that is not contained in $\mathscr{K}$, so A' returns *no*. It is easy to see that the runtime of A' is bounded by a polynomial in $|\mathcal{K}|$ and $|\mathscr{K}|$, that is A' decides ALL-MINAS in polynomial time. □

This proposition shows that the complexity of ALL-MINAS is indeed closely related to the complexity of MINA-ENUM. We now present some hardness results for enumerating MinAs when other types of KBs different from *DL-Lite*$_{horn}$ or *DL-Lite*$_{krom}$ are used. It is not difficult to see that, for all types of axioms considered in this paper except for the *DL-Lite*$_{bool}$ family, ALL-MINAS is in coNP: given an instance of ALL-MINAS, a nondeterministic algorithm can guess a subset of $\mathcal{K}$ that is not in $\mathscr{K}$, and in polynomial time verify that this is a MinA, thus proving that $\mathscr{K}$ is *not* the set of all MinAs. In the following we show that for dual-Horn KBs ALL-MINAS is at least as hard as recognizing the set of all minimal transversals of a given hypergraph. Whether the problem is coNP-hard remains unfortunately open. We later show that ALL-MINAS is coNP-complete if Horn-$\mathcal{EL}$ axioms are considered and D^{p}-hard for *DL-Lite*$_{bool}$ TBoxes.

First we briefly recall some basic notions on hypergraphs. A *hypergraph* $\mathcal{H} = (V, \mathcal{E})$ consists of a set of *vertices* $V = \{v_i \mid 1 \leq i \leq n\}$, and a set of *(hyper)edges* $\mathcal{E} = \{E_j \mid 1 \leq j \leq m\}$ where $E_j \subseteq V$. We assume w.l.o.g. that the set of edges as well as the set of vertices is nonempty, and the union of all edges yields the vertex set. A set $W \subseteq V$ is called a *transversal* of $\mathcal{H}$ if it intersects all edges of $\mathcal{H}$, i.e., $\forall E \in \mathcal{E}.\ E \cap W \neq \emptyset$. A transversal is called *minimal* if no proper subset of it is a transversal. The set of all minimal transversals of $\mathcal{H}$ constitutes another hypergraph on V called the *transversal hypergraph* of $\mathcal{H}$, which is denoted by $Tr(\mathcal{H})$. Generating $Tr(\mathcal{H})$ is an important problem which has applications in many fields of computer science [58]. The well-known decision problem associated to this computation problem is defined as follows:

Problem: TRANSVERSAL HYPERGRAPH (TRANS-HYP)
Input: Two hypergraphs $\mathcal{H} = (V, \mathcal{E}_{\mathcal{H}})$ and $\mathcal{G} = (V, \mathcal{E}_{\mathcal{G}})$.
Question: Is $\mathcal{G}$ the transversal hypergraph of $\mathcal{H}$, i.e., does $Tr(\mathcal{H}) = \mathcal{G}$ hold?

TRANS-HYP is known to be in coNP, but its lower bound is a prominent open problem. More precisely, so far neither a polynomial time algorithm has been found, nor has it been proved to be coNP-hard. In a landmark paper [59] Fredman and Khachiyan proved that TRANS-HYP can be solved in $n^{o(\log n)}$ time, which implies that this problem is most likely not coNP-hard. More recently, Gottlob and Malizia have further improved this upper bound [60]. It is conjectured that this problem, together with several computationally equivalent problems, forms a class properly contained between P and coNP [59].

Theorem 26. ALL-MINAS *is* TRANS-HYP*-hard for dual-Horn KBs.*

Proof. Let an instance of TRANS-HYP be given by the hypergraphs $\mathcal{H} = (V, \mathcal{E}_{\mathcal{H}})$ and $\mathcal{G} = (V, \mathcal{E}_{\mathcal{G}})$. From $\mathcal{H}$ and $\mathcal{G}$ we construct an instance of ALL-MINAS as follows: for every vertex $v \in V$ we introduce a propositional variable p_v, for every edge $E \in \mathcal{E}_{\mathcal{H}}$ a propositional variable p_E, and finally one additional propositional variable a. For constructing a dual-Horn KB from $\mathcal{H}$ and a set of vertices $W \subseteq V$, we define the following operator, which is also going to be used in later proofs:

$$\mathcal{K}_{W,\mathcal{H}} := \{p_v \to \bigwedge_{v \in E, E \in \mathcal{E}_\mathcal{H}} p_E \mid v \in W\} \cup \{a \to \bigwedge_{v \in V} p_v\}.$$

Using these sets, we can then construct the KB $\mathcal{K} := \mathcal{K}_{V,\mathcal{H}}$, a set of KBs $\mathscr{K} := \{\mathcal{K}_{E,\mathcal{H}} \mid E \in \mathcal{E}_\mathcal{G}\} \subseteq \mathscr{P}(\mathcal{K})$, and the axiom $\varphi := a \to \bigwedge_{E \in \mathcal{E}_\mathcal{H}} p_E$ that follows from $\mathcal{K}$. Obviously this construction creates an instance of ALL-MINAS for dual-Horn KBs and it can be done in time polynomial in the sizes of $\mathcal{H}$ and $\mathcal{G}$.

We claim that $\mathcal{G}$ is the transversal hypergraph of $\mathcal{H}$ if and only if $\mathscr{K}$ is precisely the set of all MinAs for φ in $\mathcal{K}$. Note that $a \to \bigwedge_{v \in V} p_v$ is the only axiom in $\mathcal{K}$ such that a appears on the lhs, which implies that every MinA must contain this axiom. Hence, every MinA is of the form $\mathcal{K}_{W,\mathcal{H}}$ for some $W \subseteq V$. To prove our claim, it suffices to show that a set of vertices $W \subseteq V$ is a minimal transversal of $\mathcal{H}$ if and only if the set of axioms $\mathcal{K}_{W,\mathcal{H}}$ is a MinA.
($\Rightarrow$) Assume that W is a minimal transversal of $\mathcal{H}$. By definition W satisfies $W \cap E \neq \emptyset$ for every $E \in \mathcal{E}_H$. This implies that $\mathcal{K}_{W,\mathcal{H}} \models \varphi$ holds. Moreover, $\mathcal{K}_{W,\mathcal{H}}$ is minimal since W is minimal, i.e., $\mathcal{K}_{W,\mathcal{H}}$ is a MinA.
($\Leftarrow$) Now assume that $\mathcal{K}_{W,\mathcal{H}}$ is a MinA. Then every p_E where $E \in \mathcal{E}_\mathcal{H}$ appears on the rhs of at least one of the axioms in $\mathcal{K}_{W,\mathcal{H}}$. This implies that W intersects every E, i.e., it is a transversal of $\mathcal{H}$. Moreover it is minimal since $\mathcal{K}_{W,\mathcal{H}}$ is minimal. □

A direct consequence of this theorem is that the enumeration of all MinAs in a dual-Horn KB is at least as hard as the enumeration of the transversals of a hypergraph.[5]

Corollary 27. MINA-ENUM *for dual-Horn KBs is at least as hard as enumerating hypergraph transversals.*

Up to now we have investigated the complexity of MINA-ENUM for the propositional and the simple DL-Lite cases. In particular, we have presented a polynomial delay algorithm for enumerating all MinAs in a Horn KB. However, whether such an algorithm exists for dual-Horn KBs remained open. We now turn our attention to $\mathcal{EL}$ KBs, and show that there is no output polynomial algorithm that enumerates all MinAs in a Horn-$\mathcal{EL}$ KB, unless P = NP. As a first step to this result, we show that ALL-MINAS is intractable for Horn-$\mathcal{EL}$ KBs.

Theorem 28. ALL-MINAS *is* coNP*-complete for Horn-$\mathcal{EL}$ and $\mathcal{EL}$ TBoxes.*

Proof. We have already shown that it is in coNP for $\mathcal{EL}$ TBoxes. It then suffices to show coNP-hardness for Horn-$\mathcal{EL}$. We present a reduction from the following coNP-hard problem [37, 61].

[5] MINA-ENUM is likely harder than TRANS-ENUM, although this claim has not yet been proven.

Problem: ALL-MV
Input: A monotone Boolean formula ϕ and a set $\mathscr{V}$ of minimal valuations satisfying ϕ.
Question: Is $\mathscr{V}$ precisely the set of all minimal valuations satisfying ϕ?

Let $\phi, \mathscr{V}$ be an instance of ALL-MV; we denote as $\mathsf{sub}(\phi)$ the set of all subformulas of ϕ, and define $\mathsf{csub}(\phi) := \mathsf{sub}(\phi) \setminus \{p \in \mathsf{sub}(\phi) \mid p \text{ is a propositional variable}\}$. We introduce three concept names B_ψ, C_ψ, D_ψ, and two role names r_ψ, s_ψ for every subformula ψ of ϕ and two additional concept names A and E. For each $\psi \in \mathsf{sub}(\phi)$ we define a TBox $\mathcal{T}_\psi$ as follows: if ψ is the propositional variable p, then $\mathcal{T}_\psi := \{A \sqsubseteq B_p\}$; if $\psi = \psi_1 \wedge \psi_2$, then

$$\mathcal{T}_\psi := \{A \sqsubseteq \exists r_\psi.C_\psi,\ C_\psi \sqsubseteq B_{\psi_1},\ C_\psi \sqsubseteq B_{\psi_2},\ \exists r_\psi.B_\psi \sqsubseteq D_\psi,\ B_{\psi_1} \sqcap B_{\psi_2} \sqsubseteq B_\psi\},$$

and if $\psi = \psi_1 \vee \psi_2$, then

$$\begin{aligned}\mathcal{T}_\psi := \{&A \sqsubseteq \exists r_\psi.B_{\psi_1},\ A \sqsubseteq \exists s_\psi.B_{\psi_2},\\ &\exists r_\psi.B_\psi \sqcap \exists s_\psi.B_\psi \sqsubseteq D_\psi,\ B_{\psi_1} \sqsubseteq B_\psi,\ B_{\psi_2} \sqsubseteq B_\psi\}.\end{aligned}$$

Finally, we set

$$\mathcal{T} := \bigcup_{\psi \in \mathsf{sub}(\phi)} \mathcal{T}_\psi \cup \{ \bigsqcap_{\psi \in \mathsf{csub}(\phi)} D_\psi \sqcap B_\phi \sqsubseteq E\}.$$

Notice that for every $\mathcal{T}' \subseteq \mathcal{T}$, if $\mathcal{T}' \models A \sqsubseteq E$, then also $A \sqsubseteq D_\psi$ for every $\psi \in \mathsf{csub}(\phi)$. But in order to have $A \sqsubseteq D_\psi$, all the axioms in $\mathcal{T}_\psi$ are necessary, and thus $\mathcal{T}_\psi \subseteq \mathcal{T}'$. In particular, if $\psi = \psi_1 \wedge \psi_2$, then $B_{\psi_1} \sqcap B_{\psi_2} \sqsubseteq B_\psi \in \mathcal{T}'$, and if $\psi = \psi_1 \vee \psi_2$, then $\{B_{\psi_1} \sqsubseteq B_\psi, B_{\psi_2} \sqsubseteq B_\psi\} \subseteq \mathcal{T}'$. Thus, a valuation $\mathcal{V}$ satisfies ϕ iff the KB

$$\mathcal{T}_\mathcal{V} := \{A \sqsubseteq B_p \mid p \in \mathcal{V}\} \cup \bigcup_{\psi \in \mathsf{csub}(\phi)} \mathcal{T}_\psi \cup \{ \bigsqcap_{\psi \in \mathsf{csub}(\phi)} D_\psi \sqcap B_\phi \sqsubseteq E\}$$

entails $A \sqsubseteq E$. This in particular shows that $\mathscr{V}$ is the set of all minimal valuations satisfying ϕ iff $\{\mathcal{T}_\mathcal{V} \mid \mathcal{V} \in \mathscr{V}\}$ is the set of all MinAs for $A \sqsubseteq E$ in $\mathcal{T}$. □

The following is an immediate consequence of Theorem 28 and Proposition 25.

Corollary 29. *For Horn-$\mathcal{EL}$ and $\mathcal{EL}$ TBoxes* MINA-ENUM *cannot be solved in output polynomial time, unless* P =NP.

Having shown intractability results for Horn-$\mathcal{EL}$ and $\mathcal{EL}$, we now turn our attention again to the DL-Lite-family. The only case remaining here is the complexity of enumerating MinAs if general DL-Lite concepts are allowed when forming axioms. As shown in [10], deciding whether an axiom follows from a *DL-Lite$_{bool}$* TBox is already coNP-hard. Since computing a MinA is at least as hard as doing a consequence check, we cannot expect to find a single MinA in polynomial time. This in particular implies that MinAs cannot be enumerated with polynomial delay, or even in incremental polynomial time, in the *DL-Lite*$_{bool}$ setting. However, it could still be the case that all MinAs can be computed in output polynomial time, e.g. if all cases where finding a MinA is hard happen to contain exponentially many MinAs. We show now that, unfortunately, this is not the case.

Lemma 30. ALL-MINAS *is* D^{P}*-hard for DL-Lite*$_{bool}$ *KBs. This already holds if the axioms in* $\mathcal{T}$ *are of the form* $A \sqsubseteq C$ *where* A *is a concept name and* C *a general concept.*

Proof. This result is a simple consequence of the construction from Theorem 4 together with the fact that $\mathcal{M}$ is a MinA for φ in $\mathcal{K}$ iff $\{\mathcal{M}\}$ is the set of all MinAs for φ in $\mathcal{M}$. □

It is easy to show that ALL-MINAS w.r.t. *DL-Lite*$^{\mathcal{N}}_{bool}$ TBoxes is in Π^{P}_2: if $\mathscr{K}$ is not the set of all MinAs, one just needs to guess a $\mathcal{M} \subseteq \mathcal{K}$ and verify with a coNP oracle that $\mathcal{M} \models \phi$ and for all $\mathcal{N} \in \mathscr{K}$, $\mathcal{N} \not\subseteq \mathcal{M}$. The following is an immediate consequence of Proposition 25 and Lemma 30.

Corollary 31. *For DL-Lite*$_{bool}$ *TBoxes all MinAs cannot be computed in output-polynomial time if* P $\neq$ NP*.*

Notice that Proposition 25 can also be used in the converse direction; that is, if MINA-ENUM can be solved in output polynomial time, then ALL-MINAS is in P. We thus have that, for all types of KBs studied in this paper, for which MINA-ENUM is output-polynomial, the decision problem ALL-MINAS is decidable in polynomial time.

4.2. Enumeration in a Specified Order

We now consider the case when MinAs are required to be output in a specified order. First recall that if an enumeration problem can be solved without ordering in output polynomial time, then it can also be solved with *any* ordering in output polynomial time: one needs only compute all the solutions of the problem, which takes polynomial time in the size of the output, and then display them in the desired order. In particular, this means that enumerating all MinAs for *DL-Lite*$^{\mathcal{N}}_{horn}$ and *DL-Lite*$^{\mathcal{N}}_{krom}$ KBs (and their sublogics) in any arbitrary ordering can be done in output polynomial time. Obviously, this also means that for Horn-$\mathcal{EL}$ and *DL-Lite*$_{bool}$ KBs the ordered enumeration problem is not output polynomial.

In this section we study whether the upper bounds can be improved for some orders. We first consider enumerations based on the lexicographical ordering, and afterwards look at the case where MinAs are ordered according to their size.

If we want to enumerate MinAs in some given order using an incremental polynomial algorithm, a necessary condition is that the first MinA, according to that ordering, can be computed in polynomial time. In particular, if we consider the lexicographical order, one would need to compute the lexicographical first MinA in polynomial time. For that reason, we look first at the complexity of this problem and its decision variant.

Problem: FIRST-MINA
Input: A KB $\mathcal{K}$ and an axiom φ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \varphi$, a MinA $\mathcal{M}$ for φ in $\mathcal{K}$, and a linear order on $\mathcal{K}$.
Question: Is $\mathcal{M}$ the first MinA w.r.t. the lexicographic order induced by the given linear order?

This problem is of particular interest when, for instance, one can assign a degree of trust to the axioms in the KB. In this setting if we order the axioms in such a way that less trusted axioms appear before the more trusted ones, the first lexicographical MinA will be the one that has the most distrusted axioms, and hence the most likely cause of an error. As we show now, finding the first lexicographical MinA is coNP-complete already for core KBs.

Theorem 32. FIRST-MINA *is co*NP*-complete for core KBs.*

Proof. The problem is in coNP. If $\mathcal{M}$ is not the first MinA, a proof of this can be given by guessing a subset of $\mathcal{K}$ and verifying in polynomial time that it is a MinA, and it is lexicographically smaller than $\mathcal{M}$. To show coNP-hardness, we present a reduction from the NP-hard MINA-RELEVANCE problem of core KBs (see Theorem 7) to the negation of FIRST-MINA.

Let $\mathcal{K}, \varphi, \psi$ be an instance of MINA-RELEVANCE, and let $\mathcal{M}$ be an arbitrary but fixed MinA for φ in $\mathcal{K}$. $\mathcal{M}$ can be computed time polynomial on the size of $\mathcal{K}$ as described in the proof of Theorem 17. We can assume w.l.o.g. that $\psi \notin \mathcal{M}$ since otherwise, we would already know that this is a positive instance of the problem. Consider an ordering of the axioms in $\mathcal{K}$ such that:

- $\psi < \chi$ for every $\chi \neq \psi$, and
- for every $\chi \in \mathcal{M}$ and every $\chi' \in \mathcal{K} \setminus (\mathcal{M} \cup \{\psi\})$, $\chi < \chi'$.

$\mathcal{K}, \mathcal{M}$, and φ form an instance of FIRST-MINA, and as explained above is built in polynomial time. We show that $\mathcal{M}$ is *not* the lexicographical first MinA according to the ordering described above iff there is a MinA for ϕ in $\mathcal{K}$ that contains ψ.
($\Rightarrow$) Suppose that there is a MinA $\mathcal{M}'$ for φ that is lexicographically before $\mathcal{M}$. Then, there must exist an axiom $\chi \in \mathcal{M}' \setminus \mathcal{M}$ that is smaller than any axiom in $\mathcal{M}$ according to the ordering provided. By construction, ψ is the only axiom that is smaller than any axiom in $\mathcal{M}$. Thus, $\psi \in \mathcal{M}'$.
($\Leftarrow$) If there is a MinA $\mathcal{M}'$ such that $\psi \in \mathcal{M}'$, then by definition $\mathcal{M}'$ is lexicographically smaller than $\mathcal{M}$. □

Since generating the first lexicographical MinA is already intractable, Theorem 32 has the following consequence:

Corollary 33. *Unless* P = NP*, MinAs cannot be enumerated for core KBs in lexicographic order in incremental polynomial time.*

Notice that the construction provided in the proof of Theorem 32 cannot be used to polynomially reduce MINA-RELEVANCE to the negation of FIRST-MINA in $DL\text{-}Lite^{\mathcal{N}}_{bool}$ KBs. Indeed, the first step in this construction requires the computation of a MinA, which cannot be done in polynomial time for this logic. On the other hand, FIRST-MINA is at least as hard as IS-MINA, thus yielding the following bounds.

Theorem 34. FIRST-MINA *for $DL\text{-}Lite^{\mathcal{N}}_{bool}$ KBs is D^{P}-hard and in Π^{P}_2.*

Proof. For the upper bound, if $\mathcal{M}$ is not the first MinA, then one can guess a subset $\mathcal{N}$ of $\mathcal{K}$ and verify with polynomially many calls to an NP oracle that $\mathcal{N}$ is a MinA, and lexicographically smaller than $\mathcal{M}$. For the lower bound, we reduce IS-MINA to FIRST-MINA as follows: $\mathcal{M}$ is a MinA for φ in $\mathcal{K}$ iff $\mathcal{M}$ is the first lexicographic MinA for φ in $\mathcal{M}$. □

Although computing the first MinA is coNP-hard for core KBs, interestingly computing the last MinA is polynomial for all types of KBs we consider here where reasoning is polynomial. To see this, notice that Algorithm 1 can be modified such that the **for** loop selects the axioms according to the specified linear order on $\mathcal{K}$ increasingly. It is easy to see that the MinA obtained through this strategy is necessarily the last according to the induced lexicographical ordering. It is also easy to see that deciding whether a subset of a $DL\text{-}Lite^{\mathcal{N}}_{krom}$ TBox is the last lexicographical MinA remains in D^{P}. More interestingly, it is also possible to modify Algorithm 2 to enumerate all MinAs from a $DL\text{-}Lite^{\mathcal{N}}_{horn}$ or $DL\text{-}Lite^{\mathcal{N}}_{krom}$ KB in reverse lexicographical order with polynomial delay.

The modified algorithm keeps a set of KBs in a priority queue $\mathcal{Q}$. These KBs are the *candidates* from which the remaining MinAs are going to be computed. Each KB can contain zero or more MinAs. They are inserted into $\mathcal{Q}$ by the algorithm at a cost of $O(n \cdot log(M))$ per insertion, where n is the size of the original KB and M is the total number of such KBs inserted. Note that M can be exponentially larger than n since there can be exponentially many MinAs. That is, the algorithm potentially uses exponential space. The other operation that the algorithm performs on $\mathcal{Q}$ is to find and delete the maximum element of $\mathcal{Q}$. The maximum element of $\mathcal{Q}$ is the KB in $\mathcal{Q}$ that contains the lexicographically largest MinA among the MinAs contained in all other KBs in $\mathcal{Q}$. This operation can also be performed within $O(n \cdot log(M))$ time bound. The time bounds for insertion and deletion depend also on n since they require a computation of the last MinA. This approach is presented in Algorithm 3. In line 9 of this algorithm the computation of the KB $\mathcal{K}_i$ depends on the type of the input KB. That is, if the input $\mathcal{K}$ is a $DL\text{-}Lite^{\mathcal{N}}_{horn}$ KB, then use Definition 14; if it is a $DL\text{-}Lite^{\mathcal{N}}_{krom}$ KB, then use Definition 21.

Theorem 35. *Algorithm 3 enumerates all the MinAs for $DL\text{-}Lite^{\mathcal{N}}_{horn}$ and $DL\text{-}Lite^{\mathcal{N}}_{krom}$ KBs in reverse lexicographic order with polynomial delay.*

Proof. The algorithm terminates since $\mathcal{K}$ is finite. Soundness is shown as follows: $\mathcal{Q}$ contains initially only the original KB $\mathcal{K}$. Thus the first output is lexicographically the last MinA in $\mathcal{K}$. By Lemma 16 the MinA that comes just

Algorithm 3 Enumerating all MinAs in reverse lexicographical order

```
 1: ALL-MINAS-REV-ORDER($\mathcal{K}$,$\phi$)        ▷ ($\mathcal{K}$ a KB, $\phi$ an axiom s.t. $\mathcal{K} \models \phi$)
 2: $\mathcal{Q} := \{\mathcal{K}\}$
 3: while $\mathcal{Q} \neq \emptyset$ do
 4:     $\mathcal{J}$ := maximum element of $\mathcal{Q}$
 5:     remove $\mathcal{J}$ from $\mathcal{Q}$
 6:     $\mathcal{M}$ := the lexicographical largest MinA in $\mathcal{J}$
 7:     output $\mathcal{M}$
 8:     for $1 \leq i \leq |\mathcal{M}|$ do
 9:         compute $\mathcal{K}_i$ from $\mathcal{M}$ as in Definition 14 or Definition 21
10:         insert $\mathcal{K}_i$ into $\mathcal{Q}$ if $\mathcal{K}_i \models \phi$
```

before the last one is contained in exactly one of the $\mathcal{K}_i$s that are computed and inserted into $\mathcal{Q}$ in lines 9 and 10. In line 4 $\mathcal{J}$ is assigned the KB that contains this MinA. Thus the next output will be the MinA that comes just before lexicographically the last one. It is not difficult to see that in this way the MinAs will be enumerated in reverse lexicographic order. By Lemma 16 it is guaranteed that the algorithm enumerates all MinAs.

In one iteration, the algorithm performs one find operation and one delete operation on $\mathcal{Q}$, which both take time $O(n \cdot log(M))$, and a MinA computation that takes $O(n)$ time. In addition it performs at most n $\mathcal{K}_i$ computations, and at most n insertions into $\mathcal{Q}$. Each $\mathcal{K}_i$ computation takes $O(n^2)$ time, and each insertion takes $O(n \cdot log(M))$ time. The total delay between outputs is thus $O(2 \cdot (n \cdot log(M)) + n + n \cdot (n^2 + n \cdot log(M))) = O(n^3)$. □

Notice that the polynomial delay running time of Algorithm 3 does not depend on our use of the reverse lexicographical ordering. In fact, this algorithm can be modified to enumerate MinAs in any order with polynomial delay, as long as the first MinA according to this ordering is computable in polynomial time. We will use this fact next, as we study enumeration by size.

We start by showing that is not possible to enumerate all the MinAs in decreasing size order in incremental polynomial time (unless P = NP). As it was the case for the enumeration in lexicographical order, this claim is a direct consequence of the hardness of the following decision problem.

Problem: LARGEST-MINA
Input: A KB $\mathcal{K}$ and an axiom φ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \varphi$, and $n \geq 1$.
Question: Is there a MinA $\mathcal{M}$ such that $|\mathcal{M}| \geq n$?

Theorem 36. LARGEST-MINA *is* NP*-complete for core KBs.*

Proof. An algorithm that shows NP membership simply guesses a subset $\mathcal{M}$ of the KB and verifies in polynomial time that $\mathcal{M}$ is a MinA and $|\mathcal{M}| \geq n$. Hardness is shown through a reduction from the well known NP-hard HAMILTONIAN PATH decision problem [50].

Let $\mathcal{G} = (V, E)$ be a directed graph, and $m = |V|$. For each $v \in V$, consider a concept name A_v and construct the core KB

$$\mathcal{K}_\mathcal{G} := \{A_v \sqsubseteq A_w \mid (v, w) \in E\} \cup \{A \sqsubseteq A_v, A_v \sqsubseteq B \mid v \in V\},$$

where A, B are two additional (new) concept names. Then, there is a Hamiltonian path in $\mathcal{G}$ (that is, a simple path that visits all nodes in V) if and only if there is a MinA for $A \sqsubseteq B$ in $\mathcal{K}_\mathcal{G}$ of length $m + 1$. □

This theorem in particular means that the largest MinA, which would be the first MinA to be returned in decreasing size order, cannot be computed in polynomial time. Thus, no incremental polynomial algorithm exists for any of the languages considered here.

Corollary 37. *Unless* P = NP*, MinAs cannot be enumerated for core KBs in decreasing size order in incremental polynomial time.*

Clearly, for *DL-Lite*$_{bool}$ KBs LARGEST-MINA is D^P-hard since it can be used to decide whether a set $\mathcal{M}$ is a MinA: $\mathcal{M}$ is a MinA iff there is a MinA in $\mathcal{M}$ of size at least $|\mathcal{M}|$. Moreover, the non-deterministic algorithm sketched in the proof of Theorem 36 also provides a Σ_2^P upper bound for the problem in *DL-Lite*$^{\mathcal{N}}_{bool}$ KBs.

We have thus shown that it is hard to enumerate all MinAs in decreasing size. Conversely, we can consider the enumeration by increasing size. Such a case is important since smaller MinAs are typically easier to understand by users, thus providing more informative explanations for the causes of the consequence to follow.[6] For this case, it will be important to consider the following decision problem.

Problem: SMALLEST-MINA
Input: A KB $\mathcal{K}$ and an axiom φ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \varphi$, and $n \geq 1$.
Question: Is there a MinA $\mathcal{M}$ such that $|\mathcal{M}| \leq n$?

It was previously shown in [37] that SMALLEST-MINA is NP-hard already for Horn KBs. This immediately yields a negative result for enumerating in increasing size for this class of KBs.

Proposition 38. *Unless* P = NP*, MinAs cannot be enumerated for Horn KBs in increasing size order in incremental polynomial time.*

In addition, Corollary 27 shows that enumerating MinAs in dual-Horn KBs in increasing size is likely not to be incremental polynomial, since it must be at least as hard as enumerating the transversals of a hypergraph. This negative result arises despite the fact that for this class of KBs, SMALLEST-MINA is decidable in polynomial time, slightly generalizing the ideas of computing shortest

[6] Although the size of a MinA is not the only factor defining its cognitive complexity [62, 63], it provides a simple and effective heuristic for understanding the causes of an error.

Algorithm 4 Finding a smallest MinA for $\mathit{DL\text{-}Lite}^{\mathcal{N}}_{krom}$ KBs

```
Procedure smallest-MinAs($\mathcal{T}, A \sqsubseteq B$)          ▷ ($\mathcal{T}$ $\mathit{DL\text{-}Lite}^{\mathcal{N}}_{krom}$ TBox)
if $\mathcal{T} \not\models A \sqsubseteq B$ then return $\infty$
else
    $\mathcal{M}_0 \leftarrow \{\phi \mid \mathcal{I}_{\mathcal{T}} \cup \{p_B \rightarrow p_{\neg A}\} \models \phi\}$
    $k \leftarrow 0$, $n \leftarrow -1$
    repeat
        if $p_A \rightarrow p_{\neg A} \in \mathcal{M}_k$ then
            $n \leftarrow k$
        $\mathcal{M}_{k+1} \leftarrow \mathcal{M}_k \cup \bigcup_{\psi \in \mathcal{M}_k, t \in \mathcal{T}} \{\phi \mid \mathcal{I}_{\mathcal{T}} \cup \mathcal{C}_t \cup \{p_B \rightarrow p_{\neg A}, \psi\} \models \phi\}$
        $k \leftarrow k + 1$
    until $n \geq 0$
    return $n$
```

paths in directed graphs. We now focus on the remaining cases, between core and $\mathit{DL\text{-}Lite}^{\mathcal{N}}_{krom}$ KBs. In this context, we first show that SMALLEST-MINA is decidable in polynomial time for these logics.

Following the idea of computing the shortest path in a graph, Algorithm 4 computes the least number of (non-implicit) axioms that are used in the construction of a paired MinA in $\mathcal{H}^{\phi}_{\mathcal{T}}$ (recall Definition 20). The algorithm follows a layered approach in computing all the consequences from the input TBox $\mathcal{T}$ through the core KB $\mathcal{H}^{\phi}_{\mathcal{T}}$. $\mathcal{M}_0$ contains all the consequences of $\mathcal{H}^{\phi}_{\mathcal{T}}$ that can be derived from implicit axioms only, and hence correspond to a MinA without any original axiom. $\mathcal{M}_1$ then includes all the consequences that are derivable from one axiom in $\mathcal{T}$. It can be shown through a simple induction argument that for every $k \geq 0$, $\phi \in \mathcal{M}_k$ iff there is a paired MinA for ϕ in $\mathcal{H}^{\phi}_{\mathcal{T}}$ that uses at most k axioms from $\mathcal{T}$. To avoid an infinite computation, all the sets $\mathcal{M}_k$ are restricted to the relevant core clauses, using only the variables obtained from the transformation. It is easy to see that all the sets $\mathcal{M}_k$ have polynomial size, and can be computed in polynomial time.

Theorem 39. *Algorithm 4 decides* SMALLEST-MINA *in polynomial time.*

Clearly, this algorithm can be modified to provide one smallest MinA for the desired consequence. Moreover, Algorithm 3 can be modified to consider the smallest MinA in place of the lexicographical largest one without affecting its performance. This means that all MinAs for a $\mathit{DL\text{-}Lite}^{\mathcal{N}}_{krom}$ KBs can be enumerated in increasing size order with polynomial delay.

Theorem 40. *All MinAs for a $\mathit{DL\text{-}Lite}^{\mathcal{N}}_{krom}$ KB can be enumerated in increasing size with polynomial delay.*

This finishes our analysis of the complexity of enumerating MinAs in different tractable DLs. Before turning our attention to the problem of *counting* the MinAs of a consequence that follows from a KB, we analyse the complexity of SMALLEST-MINA for $\mathit{DL\text{-}Lite}^{\mathcal{N}}_{bool}$ KBs.

Theorem 41. SMALLEST-MINA *is Σ_2^{P}-complete for DL-Lite$_{bool}$ KBs.*

Proof. The problem can be decided by guessing a subset $\mathcal{M}$ of the KB $\mathcal{K}$ and verifying with an NP oracle that $\mathcal{M}$ entails the consequence φ. Thus, it is in Σ_2^{P}. The lower bound can be obtained by a reduction from the following Σ_2^{P}-complete problem [64].

Problem: SMALLEST-MUS
Input: An inconsistent set of propositional clauses $\mathcal{F}$ and $n \geq 1$.
Question: Is there a MUS $\mathcal{M}$ for $\mathcal{F}$ such that $|\mathcal{M}| \leq n$?

Given an instance of SMALLEST-MUS, we construct $\mathcal{T}_{\mathcal{F}}$ and t_ϕ as in the proof of Theorem 4. Then, there is a MUS of size at most n iff there is a MinA for $A \sqsubseteq \bot$ in $\mathcal{T}_{\mathcal{F}}$ of size at most n. □

5. Complexity of Counting MinAs

In applications where one is interested in computing all MinAs, it might also be useful to know in advance how many of them exist. For that reason, we consider the following counting problem.

Problem: #MINA
Input: A KB $\mathcal{K}$ and an axiom ϕ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \phi$.
Output: The number of all MinAs for ϕ in $\mathcal{K}$.

As already described in the previous sections, if $\mathcal{K}$ is a core KB, the problem #MINA boils down to the problem of counting simple paths between two vertices of a given directed graph. This problem called S-T CONNECTEDNESS was one of the first counting problems considered in [45].

Problem: S-T PATHS
Input: A directed graph $G = (V, E)$, and two vertices $s, t \in V$.
Output: The number of simple paths from s to t in G.

In [45] it was shown that this problem is #P-complete. Recall that #P is defined [44] as the class of functions counting the accepting paths of nondeterministic Turing machines. Typical members of this class are the problems of counting the number of solutions of NP-complete problems. Among them, the most well-known one is #SAT, which is the problem of counting the distinct truth assignments that satisfy a given Boolean formula in CNF. Intuitively, #P is the counting-complexity analogous of the class NP for decision problems.

Since core KBs are the simplest type of KB, the hardness result applies to the other KB types we consider here. Moreover for most of the types of axioms considered in this paper, the problem of deciding whether a given set of axioms is a MinA is polynomial. This implies that for these fragments #MINA is in #P, thus it is #P-complete.

Corollary 42. #MINA *is #P-complete for core, Horn, dual-Horn, Horn-$\mathcal{EL}$, $\mathcal{EL}$, DL-Lite$_{core}^{\mathcal{N}}$ and DL-Lite$_{horn}^{\mathcal{N}}$ KBs.*

Next we consider a variant of this counting problem. Instead of the number of all MinAs, one can also be interested in the number of MinAs that contain a specific axiom. We call this problem #MINA-RELEVANCE.

Problem: #MINA-RELEVANCE
Input: A KB $\mathcal{K}$ and an axiom ϕ of the same type as $\mathcal{K}$ such that $\mathcal{K} \models \phi$, and an axiom $\psi \in \mathcal{K}$.
Output: The number of all MinAs for ϕ in $\mathcal{K}$ that contain ψ.

If we are trying to explain an unwanted consequence, the solution of this counting problem will allow us to detect axioms that are most likely to be faulty, i. e. those that appear in the most MinAs. This idea has been proposed in [65] as a heuristic for correcting an error while minimizing the changes in the set of axioms. However, this heuristic requires the solution of a #P-complete problem.

Theorem 43. #MINA-RELEVANCE *is #*P*-complete for core KBs.*

Proof. The problem is in #P since given a core KB $\mathcal{K}$, an axiom ϕ that follows from $\mathcal{K}$, an axiom $\psi \in \mathcal{K}$, and a candidate solution $\mathcal{K}' \subseteq \mathcal{K}$, we can in polynomial time verify that $\mathcal{K}$' is a MinA and it contains ψ.

For showing #P-hardness we give a parsimonious reduction from #MINA for core KBs, which has been shown to be #P-hard above. Given an instance of #MINA with the core KB $\mathcal{K}$ and the axiom $p \to q$ we construct the core KB $\mathcal{K}' := \mathcal{K} \cup \mathcal{S}$, where $\mathcal{S} = \{q \to x\}$, and x is a fresh propositional variable not occurring in $\mathcal{K}$. It is not difficult to see that a set $\mathcal{M} \subseteq \mathcal{K}$ is a MinA for $p \to q$ in $\mathcal{K}$ if and only if $\mathcal{M} \cup \mathcal{S}$ is a MinA for $p \to x$ in $\mathcal{K}'$. Moreover, every MinA for $p \to x$ in $\mathcal{K}'$ must contain the only axiom in $\mathcal{S}$. Thus, there are exactly as many MinAs for $p \to q$ in $\mathcal{K}$ as there are for $p \to x$ in $\mathcal{K}$' containing the axiom $q \to x$. □

Obviously, Theorem 43 implies that #MINA-RELEVANCE is #P-complete also for *DL-Lite*$^{\mathcal{N}}_{horn}$, *DL-Lite*$^{\mathcal{N}}_{krom}$, and $\mathcal{EL}$ KBs, as well as all the classes in between. The only remaining case is that of *DL-Lite*$_{bool}$ KBs. For this logic, both counting problems are #NP-complete.

Theorem 44. #MINA *and* #MINA-RELEVANCE *are #*NP*-complete for KBs in DL-Lite*$_{bool}$ *and DL-Lite*$^{\mathcal{N}}_{bool}$*.*

Proof. Both problems are in #NP since all candidate solutions can be checked with an NP oracle. We show only #NP-hardness for #MINA. The hardness of #MINA-RELEVANCE follows from this result using the same arguments from the proof of Theorem 43.

The prove that #MINA is #NP-hard, we provide a weakly parsimoniuos reduction from the canonical #NP-complete problem #Π_1SAT [66, 67].

Problem: #Π_1SAT
Input: A formula $\psi = \forall \mathbf{x}.\phi(\mathbf{x}, \mathbf{y})$, with $\mathbf{x}, \mathbf{y}$ sets of variables and ϕ a Boolean formula.
Output: The number of truth assignments of $\mathbf{y}$ that satisfy ψ.

For every variable z appearing in ϕ, build a concept name B_z; in addition, for every $y \in \mathbf{y}$ build the variables A_y and A'_y. Using these variables, for each subformula ϕ' of ϕ, construct a concept $C_{\phi'}$ by induction as follows.

- for all $x \in \mathbf{x}$, $C_x := B_x$;
- for all $y \in \mathbf{y}$, $C_y := A_y$;
- $C_{\neg\phi_1} := \neg C_{\phi_1}$;
- $C_{\phi_1 \wedge \phi_2} := C_{\phi_1} \sqcap C_{\phi_2}$; and
- $C_{\phi_1 \vee \phi_2} := C_{\phi_1} \sqcup C_{\phi_2}$.

Consider now the TBox $\mathcal{T}_\psi := \{A_y \sqsubseteq B_y \sqcap A'_y, A_y \sqsubseteq \neg B_y \sqcap A'_y \mid y \in \mathbf{y}\}$. Recall that valuations are represented as the set of variables they make true. Given a valuation $\mathcal{V}$ of $\mathbf{y}$, let $\mathcal{M}_\mathcal{V} := \{A_y \sqsubseteq B_y \sqcap A'_y \mid y \in \mathcal{V}\} \cup \{A_y \sqsubseteq \neg B_y \sqcap A'_y \mid y \notin \mathcal{V}\}$. We first show that for every such $\mathcal{V}$ that satisfies ψ, $\mathcal{M}_\mathcal{V}$ is a MinA for

$$\bigsqcap_{y \in \mathbf{y}} A_y \sqsubseteq C_\phi \sqcap \bigsqcap_{y \in \mathbf{y}} A'_y \tag{1}$$

in $\mathcal{T}_\psi$. Since $\mathcal{V}$ satisfies ψ, C_ϕ is a tautology. Moreover, by construction $\mathcal{M}_\mathcal{V}$ entails $A_y \sqsubseteq A'_y$ for all $y \in \mathbf{y}$, and hence $\mathcal{M}_\mathcal{V}$ entails the GCI (1). In every strict subset $\mathcal{N}$ of $\mathcal{M}_\mathcal{V}$ there is a variable $y \in \mathbf{y}$ such that $\mathcal{N}$ does not entail $A_y \sqsubseteq A'_y$, and hence does not entail (1). Thus, $\mathcal{M}_\mathcal{V}$ is a MinA.

Conversely, if $\mathcal{M}$ is a MinA for (1) in $\mathcal{T}_\psi$, then $\mathcal{M} = \mathcal{M}_\mathcal{V}$ for some valuation $\mathcal{V}$ satisfying ψ, or $\mathcal{M}$ is of the form $\{A_y \sqsubseteq B_y \sqcap A'_y, A_y \sqsubseteq \neg B_y \sqcap A'_y\}$ for some $y \in \mathbf{y}$. Notice that there are exactly $|\mathbf{y}|$ MinAs of the latter form. Thus, the number of MinAs of (1) in $\mathcal{T}_\psi$ is exactly the number of truth assignments of $\mathbf{y}$ satisfying ψ plus $|\mathbf{y}|$. □

6. Summary of Results

All the complexity results obtained are summarized in Table 5. In the table, all decision and counting problems are complete for the given class, except when prefixed with "in," or suffixed with "-h" (-hard). For the complexity of enumeration, PD, NOP, and NIP stand for *polynomial delay*, *not output polynomial*, and *not incremental polynomial*, respectively. The cells marked with OP denote problems that can be enumerated in output polynomial time, but *not* in incremental polynomial time.

As it can be seen from the table, we have obtained an almost complete picture of the complexity of axiom pinpointing in the prominent families of lightweight DLs. The most relevant open problems remaining correspond to the dual-Horn case, where hardness w.r.t. hypergraph transversal decision and enumeration has been shown, but no matching upper bound has been provided. We conjecture that all these problems are in fact harder; i.e., that ALL-MINAS is coNP-complete, and none of the enumeration problems can be solved in output polynomial time.

Table 5: Complexity of related decision, counting, and enumeration problems

Problem		[core, $DL\text{-}Lite^{\mathcal{N}}_{krom}$]	[Horn, $DL\text{-}Lite^{\mathcal{N}}_{horn}$]	dual-Horn	Horn-$\mathcal{EL}$, $\mathcal{EL}$	$DL\text{-}Lite_{bool}$, $DL\text{-}Lite^{\mathcal{N}}_{bool}$
IS-MINA		P	P	P	P	D^{P}
FIRST-MINA		coNP	coNP	coNP	coNP	D^{P}-h, in Π^{P}_2
LAST-MINA		P	P	P	P	D^{P}
LARGEST-MINA		NP	NP	NP	NP	D^{P}-h, in Σ^{P}_2
SMALLEST-MINA		P	NP	P	NP	Σ^{P}_2
ALL-MINAS		P	P	TH-h	coNP	D^{P}-h, in Π^{P}_2
MINA-RELEVANCE		NP	NP	NP	NP	Σ^{P}_2
MINA-IRREL		NP	NP	NP	NP	in Σ^{P}_2
#MINA		#P	#P	#P	#P	#NP
#MINA-RELEVANCE		#P	#P	#P	#P	#NP
MINA-ENUM	unordered	PD	PD	TE-h	NOP	NOP
	lex forward	OP	OP	NIP	NOP	NOP
	lex backward	PD	PD	TE-h	NOP	NOP
	size increasing	PD	OP	TE-h	NOP	NOP
	size decreasing	OP	OP	NIP	NOP	NOP

The remaining open problems refer to the precise complexity of some problems w.r.t. $DL\text{-}Lite_{bool}$. Notably, to the best of our efforts, we were unable to prove any relevant lower bound for MINA-IRRELEVANCE in this case, beyond the obvious NP-hardness that follows from its sublogics. Notice that to solve any of the problems where no tight complexity bounds have been found, it is necessary to find a MinA with some additional properties; in particular, one would need to prove that the set constructed is indeed a MinA. This leads us to believe that all those problems are hard for the second level of the polynomial hierarchy. These cases are of less interest in the context of lightweight DLs, as the basic reasoning problem is already NP-hard. However, we intend to work on closing those gaps.

7. Conclusions

We have studied the complexity of axiom pinpointing and several related decision, counting and enumeration problems for lightweight description logics.

Using different reductions and novel algorithms, we were able to find tight complexity bounds for most of the problems and languages considered. All these results are summarised in Table 5.

One important thing to consider is that our focus here is on the computational complexity classes, and not on the specific resource consumption needed to solve the problem. For instance, for most of our logics, we have proven a polynomial upper bound for IS-MINA using the black-box method described in Algorithm 1. Since entailment of Horn formulas is decidable in linear time, it is easy to see that our algorithm yields a quadratic decision procedure in this setting. However, it was recently shown, using specialised techniques, that this problem is in fact decidable in linear time [68]. Similarly, a more fine-grained run-time analysis might be required to achieve efficient algorithms even in the tractable cases. A promising approach in this direction is to exploit the ideas already developed for efficient MUS enumeration [14, 69].

Similarly, one should keep in mind that all the hardness results presented correspond to the *worst-case* behaviour of the problems. Indeed, different tools have been implemented for enumerating MinAs in Horn [68], $\mathcal{EL}$ [70–74], and even more expressive logics [7, 9, 38, 75]. Empirical evaluations over such systems have shown that axiom pinpointing and other related tasks are feasible in practice, for realistic ontologies.

As future work, we intend to close the few remaining gaps in the complexity table. We are also interested in dealing with other enumeration orderings and preference relationships between the MinAs. Another important path of research is to understand how the panorama changes in the presence of knowledge about individuals (known as ABoxes in the DL community) or data, and with more complex entailments such as queries. In such cases, a parameterised study of the complexity may become relevant.

Acknowledgements

The authors want to thank Joao Marques-Silva for useful pointers to the complexity of MUS computation.

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.

[2] M. A. Musen, The Protégé project: A look back and a look forward, AI Matters 1 (4) (2015) 4–12. `doi:10.1145/2757001.2757003`.

[3] L. Halilaj, N. Petersen, I. Grangel-González, C. Lange, S. Auer, G. Coskun, S. Lohmann, Vocol: An integrated environment to support version-controlled vocabulary development, in: E. Blomqvist, P. Ciancarini, F. Poggi, F. Vitali (Eds.), Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW

2016), Vol. 10024 of Lecture Notes in Computer Science, 2016, pp. 303–319. doi:10.1007/978-3-319-49004-5_20.
URL http://dx.doi.org/10.1007/978-3-319-49004-5_20

[4] A. Alobaid, D. Garijo, M. Poveda-Villalón, I. S. Pérez, Ó. Corcho, Ontoology, a tool for collaborative development of ontologies, in: F. M. Couto, J. Hastings (Eds.), Proceedings of the International Conference on Biomedical Ontology, (ICBO 2015), Vol. 1515 of CEUR Workshop Proceedings, CEUR-WS.org, 2015.
URL http://ceur-ws.org/Vol-1515/demo3.pdf

[5] M. Horridge, B. Parsia, U. Sattler, Explaining inconsistencies in OWL ontologies, in: L. Godo, A. Pugliese (Eds.), Proceedings of the Third International Conference on Scalable Uncertainty Management, (SUM 2009), Vol. 5785 of Lecture Notes in Computer Science, Springer-Verlag, 2009, pp. 124–137.

[6] F. Baader, B. Suntisrivaraporn, Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$, in: Proceedings of the International Conference on Representing and Sharing Knowledge Using SNOMED (KR-MED'08), Phoenix, Arizona, 2008.

[7] A. Kalyanpur, B. Parsia, M. Horridge, E. Sirin, Finding all justifications of OWL DL entailments, in: Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, (ISWC 2007 + ASWC 2007), Vol. 4825 of Lecture Notes in Computer Science, Springer-Verlag, 2007, pp. 267–280.

[8] M. Horridge, B. Parsia, U. Sattler, Laconic and precise justifications in OWL, in: A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, K. Thirunarayan (Eds.), Proceedings of the 7th International Semantic Web Conference, (ISWC 2008), Vol. 5318 of Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 323–338.

[9] B. Suntisrivaraporn, G. Qi, Q. Ji, P. Haase, A modularization-based approach to finding all justifications for OWL DL entailments, in: J. Domingue, C. Anutariya (Eds.), Proceedings of the 3rd Asian Semantic Web Conference (ASWC'08), Vol. 5367 of Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 1–15.

[10] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The DL-Lite family and relations, Journal of Artificial Intelligence Research 36 (2009) 1–69.

[11] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope, in: L. P. Kaelbling, A. Saffiotti (Eds.), Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, (IJCAI 05), Professional Book Center, 2005, pp. 364–369.

[12] N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M. D. Storey, C. G. Chute, M. A. Musen, Bioportal: ontologies and integrated data resources at the click of a mouse, Nucleic Acids Research 37 (Web-Server-Issue) (2009) 170–173. doi:10.1093/nar/gkp440.
URL http://dx.doi.org/10.1093/nar/gkp440

[13] R. Cote, D. Rothwell, J. Palotay, R. Beckett, L. Brochu, The systematized nomenclature of human and veterinary medicine, Tech. rep., International, Northfield, IL: College of American Pathologists (1993).

[14] A. Belov, I. Lynce, J. Marques-Silva, Towards efficient MUS extraction, AI Commun. 25 (2) (2012) 97–116. doi:10.3233/AIC-2012-0523.
URL http://dx.doi.org/10.3233/AIC-2012-0523

[15] C. Ansótegui, M. Bonet, J. Levy, SAT-based MaxSAT algorithms, Artificial Intelligence 196 (2013) 77–105. doi:10.1016/j.artint.2013.01.002.
URL http://dx.doi.org/10.1016/j.artint.2013.01.002

[16] A. Morgado, F. Heras, M. H. Liffiton, J. Planes, J. Marques-Silva, Iterative and core-guided MaxSAT solving: A survey and assessment, Constraints 18 (4) (2013) 478–534. doi:10.1007/s10601-013-9146-2.
URL http://dx.doi.org/10.1007/s10601-013-9146-2

[17] R. Dividino, S. Sizov, S. Staab, B. Schueler, Querying for provenance, trust, uncertainty and other meta knowledge in rdf, Web Semantics 7 (3) (2009) 204–219. doi:10.1016/j.websem.2009.07.004.
URL http://dx.doi.org/10.1016/j.websem.2009.07.004

[18] F. Baader, M. Knechtel, R. Peñaloza, A generic approach for large-scale ontological reasoning in the presence of access restrictions to the ontology's axioms, in: A. B. et al. (Ed.), Proceedings of the 8th International Semantic Web Conference (ISWC 2009), Vol. 5823 of Lecture Notes in Computer Science, 2009, pp. 49–64.

[19] M. Ludwig, R. Peñaloza, Error-tolerant reasoning in the description logic EL, in: E. Fermé, J. Leite (Eds.), Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'14), Vol. 8761 of Lecture Notes in Artificial Intelligence, Springer-Verlag, Madeira, Portugal, 2014, pp. 107–121.

[20] T. Eiter, T. Lukasiewicz, L. Predoiu, Generalized consistent query answering under existential rules, in: C. Baral, J. P. Delgrande, F. Wolter (Eds.), Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2016), AAAI Press, 2016, pp. 359–368.
URL http://www.aaai.org/ocs/index.php/KR/KR16/paper/view/12888

[21] R. Peñaloza, Inconsistency-tolerant instance checking in tractable description logics, in: Proceedings of the International Joint Conference on Rules and Reasoning (RuleML+RR'17), Vol. 10364 of Lecture Notes in Computer Science, Springer, 2017, to appear.

[22] F. Baader, M. Knechtel, R. Peñaloza, Context-dependent views to axioms and consequences of semantic web ontologies, Journal of Web Semantics 12–13 (2012) 22–40, available at http://dx.doi.org/10.1016/j.websem.2011.11.006.

[23] F. Riguzzi, E. Bellodi, E. Lamma, R. Zese, Probabilistic description logics under the distribution semantics, Semantic Web 6 (5) (2015) 477–501. `doi:10.3233/SW-140154`.
URL `http://dx.doi.org/10.3233/SW-140154`

[24] I. I. Ceylan, R. Peñaloza, The bayesian description logic BEL, in: S. Demri, D. Kapur, C. Weidenbach (Eds.), Proceedings of the 7th International Joint Conference on Automated Reasoning (IJCAR'14), Vol. 8562 of Lecture Notes in Computer Science, Springer, 2014, pp. 480–494.

[25] I. I. Ceylan, R. Peñaloza, The bayesian ontology language BEL, Journal of Automated Reasoning 58 (1) (2017) 67–95. `doi:10.1007/s10817-016-9386-0`.
URL `http://dx.doi.org/10.1007/s10817-016-9386-0`

[26] G. Stoilos, G. B. Stamou, J. Z. Pan, Classifying fuzzy subsumption in fuzzy-EL+, in: F. Baader, C. Lutz, B. Motik (Eds.), Proceedings of the 21st International Workshop on Description Logics (DL2008), Vol. 353 of CEUR Workshop Proceedings, CEUR-WS.org, 2008.
URL `http://ceur-ws.org/Vol-353/StoilosStamouPan.pdf`

[27] R. Peñaloza, B. Sertkaya, Axiom pinpointing is hard, in: B. C. Grau, I. Horrocks, B. Motik, U. Sattler (Eds.), Proceedings of the 2009 International Workshop on Description Logics (DL2009), Vol. 477 of CEUR-WS, 2009.

[28] R. Peñaloza, B. Sertkaya, On the complexity of axiom pinpointing in the $\mathcal{EL}$ family of Description Logics, in: F. Lin, U. Sattler, M. Truszczynski (Eds.), Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning, (KR 2010), AAAI Press, 2010, pp. 280–289.

[29] R. Peñaloza, B. Sertkaya, Complexity of axiom pinpointing in the DL-Lite family, in: V. Haarslev, D. Toman, G. Weddell (Eds.), Proceedings of the 2010 International Workshop on Description Logics (DL2010), Vol. 573 of CEUR-WS, 2010, pp. 173–184.

[30] R. Peñaloza, B. Sertkaya, Complexity of axiom pinpointing in the DL-Lite family of description logics, in: H. Coelho, R. Studer, M. Wooldridge

(Eds.), Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), Vol. 215 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2010, pp. 29–34.

[31] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, DL-Lite: Tractable description logics for ontologies, in: Proceedings of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005), 2005, pp. 602–607.

[32] A. Borgida, D. Calvanese, M. Rodriguez-Muro, Explanation in DL-Lite, in: F. Baader, C. Lutz, B. Motik (Eds.), Proceedings of the 2008 International Workshop on Description Logics (DL 2008), Vol. 353 of CEUR-WS, 2008.

[33] A. Borgida, D. Calvanese, M. Rodriguez-Muro, Explanation in the DL-Lite family of Description Logics, in: Proceedings of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2008), Vol. 5332 of Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 1440–1457.

[34] F. Baader, Terminological cycles in a description logic with existential restrictions, in: G. Gottlob, T. Walsh (Eds.), Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03), Morgan Kaufmann, 2003, pp. 325–330.

[35] S. Brandt, Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and - what else?, in: R. L. de Mántaras, L. Saitta (Eds.), Proceedings of the 16th European Conference on Artificial Intelligence, (ECAI 2004), IOS Press, 2004, pp. 298–302.

[36] M. H. Liffiton, K. A. Sakallah, Algorithms for computing minimal unsatisfiable subsets of constraints, Journal of Automated Reasoning (JAR) 40 (1) (2008) 1–33. doi:10.1007/s10817-007-9084-z.
URL http://dx.doi.org/10.1007/s10817-007-9084-z

[37] F. Baader, R. Peñaloza, B. Suntisrivaraporn, Pinpointing in the description logic $\mathcal{EL}^+$, in: J. Hertzberg, M. Beetz, R. Englert (Eds.), Proceedings of the 30th German Conference on Artificial Intelligence (KI2007), Vol. 4667 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2007, pp. 52–67.

[38] A. Kalyanpur, B. Parsia, E. Sirin, B. C. Grau, Repairing unsatisfiable concepts in OWL ontologies., in: Y. Sure, J. Domingue (Eds.), The Semantic Web: Research and Applications. Proceedings of the 3rd European Semantic Web Conference (ESWC 2006), Vol. 4011 of Lecture Notes in Computer Science, Springer-Verlag, 2006, pp. 170–184.

[39] R. Peñaloza, Axiom pinpointing in description logics and beyond, Ph.D. dissertation, Institute for Theoretical Computer Science, TU Dresden, Germany (2009).

[40] F. Baader, R. Peñaloza, Automata-based axiom pinpointing, Journal of Automated Reasoning 45 (2) (2010) 91–129.

[41] F. Baader, R. Peñaloza, Axiom pinpointing in general tableaux, Journal of Logic and Computation 20 (1) (2010) 5–34.

[42] D. S. Johnson, M. Yannakakis, C. H. Papadimitriou, On generating all maximal independent sets, Information Processing Letters 27 (3) (1988) 119–123.

[43] S. Tsukiyama, M. Ide, H. Ariyoshi, I. Shirakawa, A new algorithm for generating all maximal independent sets, SIAM Journal on Computing 6 (1977) 505–517.

[44] L. G. Valiant, The complexity of computing the permanent, Theoretical Computer Science 8 (2) (1979) 189–201.

[45] L. G. Valiant, The complexity of enumeration and reliability problems, SIAM Journal on Computing 8 (3) (1979) 410–421.

[46] D. C. Kozen, Counting Problems and #P, Springer New York, New York, NY, 1992, pp. 138–143. doi:10.1007/978-1-4612-4400-4_26.
URL http://dx.doi.org/10.1007/978-1-4612-4400-4_26

[47] N. Creignou, M. Hermann, Complexity of generalized satisfiability counting problems, Information and Computation 125 (1) (1996) 1 – 12. doi:http://dx.doi.org/10.1006/inco.1996.0016.
URL http://www.sciencedirect.com/science/article/pii/S0890540196900164

[48] C. H. Papadimitriou, M. Yannakakis, The complexity of facets (and some facets of complexity), Journal of Computer and System Sciences 28 (2) (1984) 244 – 259. doi:http://dx.doi.org/10.1016/0022-0000(84)90068-0.

[49] C. H. Papadimitriou, D. Wolfe, The complexity of facets resolved, Journal of Computer and System Sciences 37 (1) (1988) 2 – 13. doi:http://dx.doi.org/10.1016/0022-0000(88)90042-6.

[50] M. R. Garey, D. S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness, W. H. Freeman & Company, New York, NY, USA, 1990.

[51] H. N. Gabow, S. N. Maheshwari, L. J. Osterweil, On two problems in the generation of program test paths, IEEE Trans. Softw. Eng. 2 (3) (1976) 227–231.
URL http://dx.doi.org/10.1109/TSE.1976.233819

[52] A. S. Lapaugh, C. H. Papadimitriou, The even-path problem for graphs and digraphs, Networks 14 (4) (1984) 507–513. doi:10.1002/net.3230140403.
URL http://dx.doi.org/10.1002/net.3230140403

[53] O. Kullmann, Constraint satisfaction problems in clausal form: Autarkies and minimal unsatisfiability, Electronic Colloquium on Computational Complexity (ECCC) 14 (055).
URL http://eccc.hpi-web.de/eccc-reports/2007/TR07-055/index.html

[54] J. Y. Yen, Finding K shortest loopless paths in a network, Management Science 17 (11) (1971) 712–716.

[55] L. R. Nielsen, D. Pretolani, K. A. Andersen, Finding the shortest hyperpaths using reoptimization, Oper. Res. Lett. 34 (2) (2006) 155–164.

[56] L. R. Nielsen, K. A. Andersen, D. Pretolani, Finding the K shortest hyperpaths, Computers and Operations Research 32 (6) (2005) 1477–1497.

[57] T. Eiter, G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, SIAM Journal on Computing 24 (6) (1995) 1278–1304.

[58] M. Hagen, Algorithmic and computational complexity issues of monet, Ph.D. dissertation, Institut für Informatik, Friedrich-Schiller-Universität Jena (2008).

[59] M. L. Fredman, L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms, Journal of Algorithms 21 (3) (1996) 618–628.

[60] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, The Computing Research Repository (CoRR) abs/1407.2912.
URL http://arxiv.org/abs/1407.2912

[61] T. Eiter, G. Gottlob, 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).

[62] M. Horridge, Justification based explanation in ontologies, Ph.D. thesis, University of Manchester, UK (2011).
URL http://www.manchester.ac.uk/escholar/uk-ac-man-scw:131699

[63] M. Horridge, S. Bail, B. Parsia, U. Sattler, The cognitive complexity of OWL justifications, in: L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. F. Noy, E. Blomqvist (Eds.), Proceedings of the 10th International Semantic Web Conference, (ISWC 2011), Part I, Vol. 7031 of Lecture Notes in Computer Science, Springer, 2011, pp. 241–256. doi:10.1007/978-3-642-25073-6_16.
URL http://dx.doi.org/10.1007/978-3-642-25073-6_16

[64] P. Liberatore, Redundancy in logic I: CNF propositional formulae, Artificial Intelligence 163 (2) (2005) 203–232. doi:10.1016/j.artint.2004.11.002.
URL http://dx.doi.org/10.1016/j.artint.2004.11.002

[65] S. Schlobach, Z. Huang, R. Cornet, F. Harmelen, Debugging incoherent terminologies, Journal of Automated Reasoning 39 (3) (2007) 317–349.

[66] C. Wrathall, Complete sets and the polynomial-time hierarchy, Theoretical Computer Science 3 (1) (1976) 23–33. doi:10.1016/0304-3975(76)90062-1.
URL http://dx.doi.org/10.1016/0304-3975(76)90062-1

[67] A. Durand, M. Hermann, P. G. Kolaitis, Subtractive reductions and complete problems for counting complexity classes, Theoretical Computer Science 340 (3) (2005) 496–513.

[68] J. Marques-Silva, A. Ignatiev, C. Mencía, R. Peñaloza, Efficient reasoning for inconsistent Horn formulae, in: L. Michael, A. C. Kakas (Eds.), Proceedings of the 15th European Conference On Logics In Artificial Intelligence (JELIA 2016), Vol. 10021 of Lecture Notes in Computer Science, Springer-Verlag, 2016. doi:10.1007/978-3-319-48758-8_22.
URL https://doi.org/10.1007/978-3-319-48758-8_22

[69] J. Marques-Silva, M. Janota, A. Belov, Minimal sets over monotone predicates in Boolean formulae, in: N. Sharygina, H. Veith (Eds.), Proceedings of the 25th International Conference on Computer Aided Verification (CAV 2013), Vol. 8044 of Lecture Notes in Computer Science, Springer, 2013, pp. 592–607.

[70] R. Sebastiani, M. Vescovi, Axiom pinpointing in lightweight description logics via Horn-SAT encoding and conflict analysis, in: R. A. Schmidt (Ed.), Proceedings of the 22nd International Conference on Automated Deduction, (CADE-22), Vol. 5663 of Lecture Notes in Computer Science, Springer-Verlag, 2009, pp. 84–99.

[71] N. Manthey, R. Peñaloza, S. Rudolph, Efficient axiom pinpointing in EL using SAT technology, in: M. Lenzerini, R. Peñaloza (Eds.), Proceedings of the 29th International Workshop on Description Logics (DL 2016), Vol. 1577 of CEUR Workshop Proceedings, CEUR-WS.org, 2016.
URL http://ceur-ws.org/Vol-1577/paper_33.pdf

[72] M. F. Arif, C. Mencía, J. Marques-Silva, Efficient axiom pinpointing with EL2MCS, in: S. Hölldobler, M. Krötzsch, R. Peñaloza, S. Rudolph (Eds.), Proceedings of the 38th Annual German Conference on AI (KI 2015), Vol. 9324 of Lecture Notes in Computer Science, Springer, 2015, pp. 225–233. doi:10.1007/978-3-319-24489-1_17.
URL http://dx.doi.org/10.1007/978-3-319-24489-1_17

[73] M. F. Arif, C. Mencía, A. Ignatiev, N. Manthey, R. Peñaloza, J. Marques-Silva, BEACON: an efficient sat-based tool for debugging EL+ ontologies, in: N. Creignou, D. L. Berre (Eds.), Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT 2016), Vol. 9710 of Lecture Notes in Computer Science, Springer,

2016, pp. 521–530. doi:10.1007/978-3-319-40970-2_32.
URL http://dx.doi.org/10.1007/978-3-319-40970-2_32

[74] M. Ludwig, Just: a tool for computing justifications w.r.t. ELH ontologies, in: S. Bail, B. Glimm, E. Jiménez-Ruiz, N. Matentzoglu, B. Parsia, A. Steigmiller (Eds.), Informal Proceedings of the 3rd International Workshop on OWL Reasoner Evaluation (ORE 2014), Vol. 1207 of CEUR Workshop Proceedings, CEUR-WS.org, 2014, pp. 1–7.
URL http://ceur-ws.org/Vol-1207/paper_2.pdf

[75] M. H. Liffiton, A. Previti, A. Malik, J. Marques-Silva, Fast, flexible MUS enumeration, Constraints 21 (2) (2016) 223–250. doi:10.1007/s10601-015-9183-0.
URL http://dx.doi.org/10.1007/s10601-015-9183-0