Complexity of Axiom Pinpointing in the DL-Lite Family

Rafael Peñaloza[1] and Barış Sertkaya[2]

[1] Theoretical Computer Science TU Dresden, Germany
penaloza@tcs.inf.tu-dresden.de
[2] SAP Research Center Dresden, Germany
baris.sertkaya@sap.com

1 Introduction

In real world applications where ontologies are employed, often the knowledge engineer not only wants to know whether her ontology has a certain (unwanted) consequence or not, but also wants to know *why* it has this consequence. Even for ontologies of moderate size, finding explanations for a given consequence is not an easy task without getting support from an automated tool. The task of finding explanations for a given consequence, i.e., minimal subsets of the original ontology that have the given consequence is called *axiom pinpointing* in the literature.

Existing work on axiom pinpointing in DLs can be classified under two main categories, namely the glass-box approach, and the black-box approach. The idea underlying the *glass-box approach* is to extend the existing reasoning algorithms such that while doing reasoning, at the same time they can keep track of the axioms used, and detect which of the axioms in the TBox are responsible for a given consequence. In [24] a pinpointing extension of the tableau-based satisfiability algorithm for the DL $\mathcal{ALC}$ has been introduced. Later in [19], this approach has been further extended to DLs that are more expressive than $\mathcal{ALC}$. In [17] a pinpointing algorithm for $\mathcal{ALC}$ with general concept inclusions (GCIs) has been presented by following the approach in [2]. In order to overcome the problem of developing a pinpointing extension for every particular tableau-based algorithm, a general pinpointing extension for tableau algorithms has been developed in [3, 6]. Similarly, an automata-based general approach for obtaining glass-box pinpointing algorithms has been introduced in [4, 5].

In contrast to the glass-box approach, the idea underlying the *black-box approach* is to make use of the existing highly optimized reasoning algorithms wihout having to modify them. The most naïve black-box approach would of course be to generate every subset of the original TBox, and ask a DL reasoner whether this subset has the given consequence or not, which obviously is very inefficient. In [16] more efficient approaches based on Reiter's hitting set tree algorithm [23] have been presented. The experimental resuts in [16] demonstrate that this approach behaves quite well in practice on realistic TBoxes written in expressive DLs. A similar approach has successfully been used in [14] for explaining inconsistencies in OWL ontologies. The main advantages of the black-box approach are that one can use existing DL reasoners, and that it is independent of the DL reasoner being used. In [13] the black-box approach has been used for computing more fine grained explanations, i.e., not just the set of relevant axioms in the TBox but parts of these axioms that actually lead to the given consequence.

Although various methods and aspects of axiom pinpointing have been considered in the literature, its computational complexity has not been investigated in detail yet. Obviously, axiom pinpointing is at least as hard as standard reasoning. Nevertheless, especially for tractable DLs it makes sense to investigate whether explanations for a consequence can efficiently be enumerated or not. In [7] it has been shown that a given consequence can have exponentially many explanations (there called *MinAs*, which stands for *minimal axiom sets*), and checking the existence of a MinA within a cardinality bound is NP-hard even for a fragment of $\mathcal{EL}$ that only allows for conjunction on both sides of a GCI. In [20–22] we have investigated the complexity of axiom pinpointing in the propositional Horn fragment, and in the tractable DL $\mathcal{EL}$. We have given a polynomial delay algorithm for enumerating MinAs in the propositional Horn setting that works even if the MinAs are required to be enumerated in reverse lexicographic order. We have also shown that for the dual-Horn setting, where the axioms have at most one negative literal, this problem is at least as hard as the hypergraph transversal enumeration problem, whose exact complexity is a prominent open problem [12]. Moreover, we have shown that for $\mathcal{EL}$ TBoxes MinAs cannot be enumerated in output-polynomial time unless P = NP.

In the present work we investigate the complexity of axiom pinpointing in the other family of tractable DLs, namely 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, 1]. For this family various aspects of finding explanations have already been considered in [9, 8]. There the main focus is on the problem of explaining query answering and ABox reasoning, which are the most standard types of reasoning problems 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 [9, 8] here we consider the problem of explaining TBox reasoning. We investigate in detail the complexity of enumerating MinAs in a DL-Lite TBox for a given consequence of this TBox. We show that for $DL{-}Lite^{\mathcal{H}}_{core}$, $DL{-}Lite^{\mathcal{H}}_{krom}$ and $DL{-}Lite^{\mathcal{N}}_{horn}$ TBoxes MinAs are efficiently enumerable with polynomial delay, but for $DL{-}Lite_{bool}$ they cannot be enumerated in output-polynomial time unless P = NP.

2 Preliminaries

We briefly introduce the syntax of the DL-Lite family following the notation in [1]. DL-Lite concepts and roles are constructed as follows:

$$r := p \mid p^-, \qquad B := \bot \mid A \mid \ \geq q\ r, \qquad C := B \mid \neg C \mid C_1 \sqcap C_2,$$

where A is a concept name, p is a role name, and q is a natural number. Concepts of the form B are called *basic*, and those of form C are called *general* concepts.

A $DL\text{-}Lite^{\mathcal{N}}_{bool}$ TBox is a set of axioms of the form $C_1 \sqsubseteq C_2$, where C_1, C_2 are general concepts. A TBox is called *core*, denoted as $DL\text{-}Lite^{\mathcal{N}}_{core}$, 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. *Krom* TBoxes generalize core ones by allowing also axioms of the form $\neg B_1 \sqsubseteq B_2$. These TBoxes are denoted as $DL\text{-}Lite^{\mathcal{N}}_{krom}$. Finally, a *Horn* TBox $DL\text{-}Lite^{\mathcal{N}}_{horn}$ is composed only of axioms of the form $\bigsqcap_k B_k \sqsubseteq B$. We can drop the superscript $\mathcal{N}$ from the knowledge bases by allowing only number restrictions of the form $\geq 1\ r$ for constructing basic concepts. We will sometimes use the expression $\exists r$ to represent $\geq 1\ r$. To any of the previously defined TBoxes, we can add *role inclusion axioms* of the form $r_1 \sqsubseteq r_2$. This will be denoted using the superscript $\mathcal{H}$ in the name; e.g. $DL\text{-}Lite^{\mathcal{HN}}_{bool}$. Since we are not dealing with individuals in the present work, role inclusion axioms do not add any expressivity to $DL\text{-}Lite^{\mathcal{H}}_{\alpha}$ TBoxes for $\alpha \in \{core, horn, krom\}$. Indeed, a basic concept B will only make use of a role r if B is an existential restriction $\exists r$. As we are only interested in concept subsumption, we can represent the role inclusion axiom $r_1 \sqsubseteq r_2$ by the concept inclusion $\exists r_1 \sqsubseteq \exists r_2$. Thus, the complexity results we present here for for $DL\text{-}Lite_{\alpha}$ TBoxes also hold for $DL\text{-}Lite^{\mathcal{H}}_{\alpha}$ TBoxes.[3] For sake of simplicity, in the present work we do not consider inverse roles.

Finally we recall basic notions from complexity of enumeration algorithms. For analyzing the performance of algorithms where the size of the output can be exponential in the size of the input, we consider other measures of efficiency. We say that an algorithm runs with *polynomial delay* [15] if the time until the first output is generated, and thereafter the time between any two consecutive outputs is bounded by a polynomial in the size of the input. We say that it runs in *output polynomial time* [15] if it outputs all solutions in time polynomial in the size of the input *and the output*.

3 Complexity of Enumerating all MinAs

The main problem we consider in the present work is, given a DL-Lite TBox and a consequence of it, compute all MinAs for this consequence in the given TBox. We start with defining a MinA.

Definition 1. *Let $\mathcal{T}$ be a DL-Lite TBox and φ a DL-Lite axiom that follows from it, i.e., $\mathcal{T} \models \varphi$. We call a set $\mathcal{M} \subseteq \mathcal{T}$ a* minimal axiom set *or* MinA *for φ in $\mathcal{T}$ if $\mathcal{M} \models \varphi$ and it is minimal w.r.t. set inclusion.*

We define our problem without mentioning a particular DL-Lite fragment but investigate its computational complexity for different fragments in the coming sections separately. In the following, the only consequences we consider are subsumption relations that can be expressed by axioms in the corresponding DL-Lite fragment.

[3] Notice that this may not be true if number restrictions are allowed; that is, the complexity results for $DL\text{-}Lite^{\mathcal{N}}_{\alpha}$ may not transfer to $DL\text{-}Lite^{\mathcal{HN}}_{\alpha}$.

Problem: MINA-ENUM
Input: A DL-Lite TBox $\mathcal{T}$ and a DL-Lite axiom φ such that $\mathcal{T} \models \varphi$.
Output: The set of all MinAs for φ in $\mathcal{T}$.

3.1 Enumerating MinAs in $DL\!-\!Lite_{core}$ and $DL\!-\!Lite_{krom}$ TBoxes

We start with a basic observation. In the simplest setting where we can consider MINA-ENUM, $\mathcal{T}$ is a $DL\!-\!Lite_{core}$ TBox whose concept inclusion axioms are all of the form $A_1 \sqsubseteq A_2$ for atomic concepts A_1, A_2. Note that in his setting $\mathcal{T}$ becomes just a directed graph, and a MinA for $A_n \sqsubseteq A_m$ is just a simple path between the nodes A_n and A_m.[4] That is, MINA-ENUM boils down to enumerating the simple paths between two vertices in a given directed graph. This problem is well-known, and can be solved with polynomial delay, even if the simple paths are required to be output in the increasing order of their lengths [25]. This observation has already been briefly mentioned in the works [9, 8], which mainly concentrate on explaining query answering.

In $DL\!-\!Lite_{core}$ TBoxes, additionally we need to deal with unqualified existential restriction, and also with inclusion axioms that have negated basic concepts in the right hand side. Since unqualified existential restrictions cannot interact and give rise to additional MinAs in a $DL\!-\!Lite_{core}$ TBox, we can treat them as atomic concepts. We need to deal with the axioms with a negated basic concept in the right hand side separately since they can lead to additional MinAs due to contraposition. We demonstrate this with an example.

Example 1. Consider the $DL\!-\!Lite_{core}$ TBox $\mathcal{T} = \{A \sqsubseteq \neg\exists r_1,\ \exists r_2 \sqsubseteq \exists r_1,\ D \sqsubseteq \exists r_2,\ D \sqsubseteq \exists r_1,\ A \sqsubseteq D\}$ and the axiom $\varphi : A \sqsubseteq \neg D$ which follows from $\mathcal{T}$. We can treat $\exists r_1$ and $\exists r_2$ just like atomic concepts since without role inclusion axioms they cannot interact and lead to additional MinAs. That is we have the MinAs $M_1 = \{A \sqsubseteq \neg\exists r_1,\ \exists r_2 \sqsubseteq \exists r_1,\ D \sqsubseteq \exists r_2\}$, and $M_2 = \{A \sqsubseteq \neg\exists r_1,\ D \sqsubseteq \exists r_1\}$.

Note that A is actually unsatisfiable, i.e., it is subsumed by any other concept. This might also be the reason why φ follows from $\mathcal{T}$. This means that we also need to find out the reasons why A is unsatisfiable. The only MinA for $A \sqsubseteq \neg A$ in $\mathcal{T}$ is $M = \{A \sqsubseteq \neg\exists r_1,\ D \sqsubseteq \exists r_1,\ A \sqsubseteq D\}$. However, it contains M_2, which is a MinA for φ, thus M is not a *minimal* axiom set, i.e., a MinA for φ. It means that when we are looking for MinAs for an axiom $B_1 \sqsubseteq B_2$ s.t. B_1 is unsatisfiable, we also need to find MinAs for $B_1 \sqsubseteq \neg B_1$ that do *not* contain any of the MinAs for the original axiom.

Our algorithm that takes all these cases into account is described in detail in Algorithm 1 where $t(\varphi)$ stands for the tail (i.e. the left hand side), and $h(\varphi)$ stands for the head (i.e. the right hand side) of axiom φ.

Theorem 1. *Algorithm 1 solves* MINA-ENUM *for $DL\!-\!Lite_{krom}$ TBoxes with polynomial delay.*

[4] A simple path is a path with no repeated vertices.

Algorithm 1 Enumerating all MinAs for $DL\!-\!Lite_{krom}$ TBoxes

Procedure: ALL-MINAS($\mathcal{T}$,φ) ($\mathcal{T}$ a $DL\!-\!Lite_{krom}$ TBox, φ an axiom s.t. $\mathcal{T} \models \varphi$)

1: ALL-MINAS-AUX($\mathcal{T}, \varphi$)
2: **if** $\mathcal{T} \models t(\varphi) \sqsubseteq \neg t(\varphi)$ **then**
3: $\quad \mathcal{T}' := \{\psi \in \mathcal{T} \mid h(\psi) \neq h(\varphi) \text{ and } t(\psi) \neq \neg h(\varphi)\}$
4: $\quad$ ALL-MINAS-AUX($\mathcal{T}', t(\varphi) \sqsubseteq \neg t(\varphi)$) (MinAs for unsatisfiability of $t(\varphi)$)
5: **end if**

Procedure: ALL-MINAS-AUX($\mathcal{T}$,φ) ($\mathcal{T}$ a $DL\!-\!Lite_{krom}$ TBox, φ an axiom, $\mathcal{T} \models \varphi$)

1: **if** $t(\varphi) = h(\varphi)$ **then return** $\emptyset$
2: **end if**
3: **for all** $\psi \in \mathcal{T}$ **do**
4: $\quad$ **if** $t(\varphi) = t(\psi)$ and $\mathcal{T} \setminus \{\psi\} \models h(\psi) \sqsubseteq h(\varphi)$ **then**
5: $\quad\quad$ **print**$\{\psi\} \cup$ ALL-MINAS($\mathcal{T} \setminus \{\psi\}, h(\psi) \sqsubseteq h(\varphi)$)
6: $\quad$ **end if**
7: $\quad$ **if** $t(\varphi) = \neg h(\psi)$ and $\mathcal{T} \setminus \{\psi\} \models \neg t(\psi) \sqsubseteq h(\varphi)$ **then**
8: $\quad\quad$ **print**$\{\psi\} \cup$ ALL-MINAS($\mathcal{T} \setminus \{\psi\}, \neg t(\psi) \sqsubseteq h(\varphi)$)
9: $\quad$ **end if**
10: **end for**

Proof. It is not difficult to see that the algorithm terminates. Termination of the procedure ALL-MINAS depends on the termination of the procedure ALL-MINAS-AUX. ALL-MINAS-AUX terminates since the base case of the recursion is well established, and there are finitely many ψ in $\mathcal{T}$.

The algorithm is sound. ALL-MINAS-AUX outputs an axiom ψ, only if using it φ can be derived. Moreover, as soon as the head and the tail of φ become equal, it terminates in line 1. That is it does not allow 'cycles', or redundant axioms in the output. Hence, the outputs of ALL-MINAS-AUX are indeed MinAs for φ in $\mathcal{T}$. ALL-MINAS additionally checks if the tail of φ is unsatisfiable, and if this is the case also outputs the MinAs for $t(\varphi) \sqsubseteq \neg t(\varphi)$ that do not contain any of the previously output MinAs.

The algorithm is complete. ALL-MINAS-AUX iterates over the axioms in $\mathcal{T}$ and searches for the MinAs for φ in a depth-first manner. If $\mathcal{T} \models t(\varphi) \sqsubseteq \neg t(\varphi)$, then ALL-MINAS additionally searches for MinAs for $t(\varphi) \sqsubseteq \neg t(\varphi)$, in the same manner. These are all MinAs for φ in $\mathcal{T}$.

Note that in lines 4 and 7 of the procedure ALL-MINAS-AUX the algorithm checks whether the selected axiom ψ will lead to a MinA. Clearly, for $DL\!-\!Lite_{core}$ and $DL\!-\!Lite_{krom}$ this check is polynomial. Moreover, this check avoids the algorithm picking a 'wrong' axiom that will result in an exponential number of recursive calls that do not lead to a MinA. That is, it guarantees that the algorithm outputs the next MinA, or stops, after at most a polynomial number of steps, i.e., it is polynomial delay. □

3.2 MinAs in $DL\!-\!Lite^{\mathcal{N}}_{horn}$ TBoxes

Next we show that for $DL\!-\!Lite^{\mathcal{N}}_{horn}$ TBoxes, MinAs can be enumerated with polynomial delay as well. Furthermore, we show that this is true even if the

MinAs are required to be output in a given reverse lexicographic order. To do this, we construct, for every $DL\!-\!Lite^{\mathcal{N}}_{horn}$ TBox $\mathcal{T}$ a propositional Horn TBox $\mathcal{G}_\mathcal{T}$ as follows: for every basic concept B create a propositional variable v_B; for every axiom $\bigsqcap_{i=1}^n B_i \sqsubseteq B$ add the Horn clause $\bigwedge_{i=1}^n v_{B_i} \to v_B$; 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 $v_{\geq q_1 r} \to v_{\geq q_2 r}$. We will call the latter ones *implicit axioms*. It is not difficult to see that $\mathcal{T} \models \bigsqcap_{i=1}^n A_i \sqsubseteq C$ iff $\mathcal{G}_\mathcal{T} \models \bigwedge_{i=1}^n v_{A_i} \to v_C$. Furthermore, 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 $\mathcal{T}$. For instance let $\mathcal{T} = \{A \sqsubseteq\, \geq 2r, A \sqsubseteq\, \geq 3r, \geq 1r \sqsubseteq B\}$. Clearly $\mathcal{G}_\mathcal{T}$ constructed from $\mathcal{T}$ as described has three MinAs for $v_A \to v_B$, but there are only two MinAs for $A \sqsubseteq B$ in $\mathcal{T}$. The reason is that the implicit subsumption $\geq 3r \sqsubseteq\, \geq 1r$ is represented twice in $\mathcal{G}_\mathcal{T}$: one through the direct edge, and another with a path travelling along $v_{\geq 2r}$. We solve this problem by using *immediate* MinAs.

Definition 2. *Let $\mathcal{T}$ be a $DL\!-\!Lite^{\mathcal{N}}_{horn}$ TBox. A MinA $\mathcal{M}$ in $\mathcal{G}_\mathcal{T}$ is called* immediate *if for every implicit axiom $\tau \in \mathcal{G}_\mathcal{T}, \mathcal{M} \models \tau$ implies $\tau \in \mathcal{M}$.*

Note that there is a one-to-one correspondence between MinAs for $\bigsqcap_{i=1}^n A_i \sqsubseteq C$ in $\mathcal{T}$ and immediate MinAs for $\bigwedge_{i=1}^n v_{A_i} \to v_C$ in $\mathcal{G}_\mathcal{T}$. Thus, if we can enumerate all immediate MinAs in $\mathcal{G}_\mathcal{T}$ in output polynomial time, we will be able to enumerate also all MinAs in $\mathcal{T}$ within the same complexity bound. We now show how all immediate paths can be computed. For this, we first need to introduce the notion of a valid ordering on the axioms in a TBox.

Definition 3. *Let $\mathcal{T}$ be a propositional Horn TBox, and $\phi = \bigwedge_{i=1}^n a_i \to b$ be an axiom in $\mathcal{T}$. 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 $\bigwedge_{a \in A} a \to c$. We call an ordering $t_1 < \ldots < t_m$ a* valid ordering *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.*[5]

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-TBoxes that contain all and only the remaining immediate MinAs, following the ideas in [18].

Definition 4. *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 TBox $\mathcal{T}_i$ 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. $i < j \leq m$ 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.*

The naïve method for computing one MinA can be easily adapted to the computation of an immediate MinA in polynomial time by simply considering

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

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

Procedure ALL-MINAS($\mathcal{T}$,ϕ) ($\mathcal{T}$ a $DL\text{-}Lite^{\mathcal{N}}_{horn}$ TBox, ϕ an axiom s.t. $\mathcal{T} \models \phi$)

if $\mathcal{T} \not\models \phi$ **then return**
else
  $\mathcal{M}$:= an immediate MinA in $\mathcal{G}_{\mathcal{T}}$
  $\mathcal{I} := \{t \mid t$ is an implicit axiom$\}$
  output $\mathcal{M} \setminus \mathcal{I}$
  for $1 \leq i \leq |\mathcal{M}|$ **do**
    compute $\mathcal{T}_i$ from $\mathcal{M}$ as in Definition 4
    ALL-MINAS($\mathcal{T}_i \setminus \mathcal{I}$,$\phi$)
  end for
end if

first all non-implicit axioms for removal, and ordering the implicit ones as follows: if $t_1 := (\geq q_1 r) \sqsubseteq (\geq q_2 r)$, and $t_2 := (\geq q'_1 r) \sqsubseteq (\geq q'_2 r)$ are two implicit axioms and $q_1 - q_2 < q'_1 - q'_2$, then t_1 appears before t_2.

Lemma 1. *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 4. Then, for every immediate MinA $\mathcal{N}$ for ϕ in $\mathcal{T}$ that is different from $\mathcal{M}$, there exists* **exactly one** *i, where $1 \leq i \leq m$, 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 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 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 4. It cannot be step (ii) because $t_k \notin \mathcal{N}$ since $t_k \in \mathcal{M} \setminus \mathcal{N}$. Thus, it should be 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 with 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 $\mathcal{N} \not\subseteq \mathcal{T}_i$. $\square$

Lemma 1 gives an idea of how to compute the remaining MinAs from a given one in the $DL\text{-}Lite^{\mathcal{N}}_{horn}$ setting. Algorithm 2 describes how we can use this lemma to enumerate all MinAs in a $DL\text{-}Lite^{\mathcal{N}}_{horn}$ TBox $\mathcal{T}$ by enumerating all immediate MinAs in $\mathcal{G}_{\mathcal{T}}$.

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

Proof. The algorithm terminates since $\mathcal{T}$ is finite. It is sound since its outputs are MinAs for ϕ in $\mathcal{T}$. Completeness follows from Lemma 1.

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 [1]. One MinA is computed in polynomial time by iterating over the axioms in $\mathcal{T}$ and removing the redundant ones. Thus the algorithm spends at most polynomial time between each output, i.e., it is polynomial delay. $\square$

We now modify Algorithm 2 and show that it can also enumerate MinAs in reverse lexicographic order with polynomial delay. The lexicographic order we use is defined as follows:

Definition 5. *Let the elements of a set S be linearly ordered. This order induces a linear strict order on $\mathscr{P}(S)$, which is called the* lexicographic order*. We say that a set $R \subseteq S$ is* lexicographically smaller *than a set $T \subseteq S$ where $R \neq T$ if the first element at which they disagree is in R.*

The modified algorithm keeps a set of TBoxes in a priority queue $\mathcal{Q}$. These TBoxes are the "candidates" from which the MinAs are going to be computed. Each TBox can contain zero or more MinAs. They are inserted into $\mathcal{Q}$ by the algorithm at a cost of $O(|\mathcal{T}| \cdot \log(M))$ per insertion, where $\mathcal{T}$ is the original TBox and M is the total number of TBoxes inserted. Note that M can be exponentially bigger than $|\mathcal{T}|$ since there can be exponentially many MinAs. That is the algorithm uses potentially 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 TBox in $\mathcal{Q}$ that contains the lexicographically largest MinA among the MinAs contained in all other TBoxes in $\mathcal{Q}$. This operation can also be performed within $O(|\mathcal{T}| \cdot \log(M))$ time bound. Note that given a $\mathcal{T}$, the lexicographically largest MinA in $\mathcal{T}$ can be computed by starting with the axiom that is the smallest one w.r.t. the linear oder on $\mathcal{T}$, iterating over the axioms and removing an axiom if the resulting TBox still has the required consequence. Obviously this operation is in $O(|\mathcal{T}|)$. This is why the time bounds for insertion and deletion depend also on $|\mathcal{T}|$ and not only on M.

Theorem 3. *Algorithm 3 enumerates all MinAs for a $DL\!-\!Lite^{\mathcal{N}}_{horn}$ TBox in reverse lexicographic order with polynomial delay.*

Proof. The algorithm terminates since $\mathcal{T}$ is finite. Soundness is shown as follows: $\mathcal{Q}$ contains initially only the original TBox $\mathcal{T}$. Thus the first output is lexicographically the last MinA in $\mathcal{T}$. By Lemma 1 the MinA that comes just before the last one is contained in exactly one of the $\mathcal{T}_i$s that are computed and inserted into $\mathcal{Q}$ in lines 8 and 9. In line 3 $\mathcal{J}$ is assigned the TBox that contains this MinA. Thus the next output will be the MinA that comes just before the lexicographically last one. It is not difficult to see that in this way the MinAs will be enumerated in reverse lexicographic order. By Lemma 1 it is guaranteed that the algorithm enumerates all MinAs.

In one iteration, the algorithm performs one find operation and one delete operation on $\mathcal{Q}$, each of which takes time $O(n \cdot \log(M))$, and a MinA computation

Algorithm 3 Enumerating all MinAs in reverse lexicographical order

Procedure ALL-MINAS-REV-ORD($\mathcal{T}$,ϕ) ($\mathcal{T}$ a $DL{-}Lite^{\mathcal{N}}_{horn}$ TBox, ϕ an ax., $\mathcal{T} \models \phi$)

$\mathcal{Q} := \{\mathcal{T}\}$
while $\mathcal{Q} \neq \emptyset$ **do**
  $\mathcal{J}$:= maximum element of $\mathcal{Q}$
  remove $\mathcal{J}$ from $\mathcal{Q}$
  $\mathcal{M}$:= the lexicographical largest MinA in $\mathcal{J}$
  output $\mathcal{M}$
  for $1 \leq i \leq |\mathcal{M}|$ **do**
    compute $\mathcal{T}_i$ from $\mathcal{M}$ as in Definition 4
    insert $\mathcal{T}_i$ into $\mathcal{Q}$ if $\mathcal{T}_i \models \phi$
  end for
end while

that takes $O(n)$ time, where $n = |\mathcal{T}|$. In addition it performs at most n $\mathcal{T}_i$ computations, and at most n insertions into $\mathcal{Q}$. Each $\mathcal{T}_i$ requires $O(n^2)$ time to be constructed, and each insertion into $\mathcal{Q}$ takes $O(n \cdot \log(M))$ time. The total delay is thus $O(2 \cdot (n \cdot \log(M)) + n + n \cdot (n^2 + n \cdot \log(M))) = O(n^3)$. □

3.3 MinAs in $DL{-}Lite_{bool}$ TBoxes

The axioms that we have used so far allowed for only basic concepts and their negations, and we were able to show that in this restricted setting, MinAs are enumerable with polynomial delay. However, we have not yet explored the complexity of these problems if general concepts are allowed. As shown in [1], deciding whether an axiom follows from a $DL{-}Lite_{bool}$ TBox is already NP-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 in the $DL{-}Lite_{bool}$ setting. What we can ask next is whether all MinAs are computable in output polynomial time. In order to answer this, we investigate the decision version of this problem:

Problem: ALL-MINAS
Input: A DL-Lite TBox $\mathcal{T}$ and an axiom φ such that $\mathcal{T} \models \varphi$, and a set of TBoxes $\mathscr{T} \subseteq \mathscr{P}(\mathcal{T})$.
Question: Is $\mathscr{T}$ precisely the set of all MinAs for φ in $\mathcal{T}$?

Because if this problem is not solvable in polynomial time, then all MinAs cannot be computed in output-polynomial time. Due to lack of space, we cannot include the proof of this claim here. The proof is based on a general argument and can be found in [21] (Proposition 6). Next we show that ALL-MINAS is coNP-hard for $DL{-}Lite_{bool}$ TBoxes.

Lemma 2. ALL-MINAS *is coNP-hard for $DL{-}Lite_{bool}$ TBoxes. 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. We present a reduction from the following coNP-hard problem [11, 7].

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 introduce a concept name A_p for each propositional variable p appearing in ϕ and two additional concept names A_0, A_1. From ϕ we construct the general concep C_ϕ by changing each conjunction $\wedge$ to $\sqcap$, each disjunction $\vee$ to $\sqcup$ and each propositional variable p to $\neg B_p$.[6] Using these we construct the TBox $\mathcal{T} := \{A_1 \sqsubseteq \neg C_\phi\} \cup \{B_p \sqsubseteq \neg A_0 \mid p \in \mathsf{var}(\phi)\}$ and the set of MinAs $\mathscr{T} := \{\{A_1 \sqsubseteq C_\phi\} \cup \{B_p \sqsubseteq \neg A_0 \mid p \in \mathcal{V}\} \mid \mathcal{V} \in \mathscr{V}\}$. It is easy to see that $\mathcal{T}$ and $\mathscr{T}$ indeed form an instance of ALL-MINAS for the axiom $A_0 \sqsubseteq \neg A_1$. Furthermore, $\mathscr{T}$ is the set of all MinAs for $A_0 \sqsubseteq \neg A_1$ iff $\mathscr{V}$ is the set of all minimal valuations satisfying ϕ. □

The following is an immediate consequence of Lemma 2.

Corollary 1. *For $DL\text{−}Lite_{bool}$ TBoxes all MinAs cannot be computed in output-polynomial time unless* P = NP.

4 Concluding Remarks and Future Work

We have investigated the complexity of axiom pinpointing in the DL-Lite family. We have shown that for $DL\text{−}Lite^{\mathcal{H}}_{core}$, $DL\text{−}Lite^{\mathcal{H}}_{krom}$ and $DL\text{−}Lite^{\mathcal{N}}_{horn}$ TBoxes MinAs are efficiently enumerable with polynomial delay, but for $DL\text{−}Lite_{bool}$ they cannot be enumerated in output-polynomial time unless P = NP. For simplicity we did not consider inverse roles here, although we believe our results will hold in presence of inverse roles. As future work we are going to investigate whether this is the case.

Finding explanations for query answering and ABox reasoning has already been considered in [9, 8]. However, these works investigate computing only one explanation. As future work we are going to work on the problem of computing all MinAs for explaining the reasoning problems considered there.

Acknowledgements We are grateful to the anonymous reviewers for pointing out some problems, which allowed us to clarify difficult ideas and improve the overall quality of this paper.

References

1. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.

[6] We use the abbreviation $X \sqcup Y$ for $\neg(\neg X \sqcap \neg Y)$.

2. F. Baader and B. Hollunder. Embedding defaults into terminological representation systems. *Journal of Automated Reasoning*, 14:149–180, 1995.
3. F. Baader and R. Peñaloza. Axiom pinpointing in general tableaux. In *Proc. of TABLEAUX 2007*, volume 4548 of *LNCS*, pages 11–27. Springer-Verlag, 2007.
4. F. Baader and R. Peñaloza. Automata-based axiom pinpointing. In *Proc. of the 4th Int. Joint Conf. on Automated Reasoning, (IJCAR 2008)*, volume 5195 of *LNCS*, pages 226–241. Springer-Verlag, 2008.
5. F. Baader and R. Peñaloza. Automata-based axiom pinpointing. *Journal of Automated Reasoning*, 2009. To appear.
6. F. Baader and R. Peñaloza. Axiom pinpointing in general tableaux. *Journal of Logic and Computation*, 2010. To appear.
7. F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *Proc. of the 30th German Conf. on Artificial Intelligence (KI2007)*, volume 4667 of *LNAI*, pages 52–67. Springer-Verlag, 2007.
8. A. Borgida, D. Calvanese, and M. Rodriguez-Muro. Explanation in DL-Lite. In *Proc. of the 2008 Int. Workshop on Description Logics (DL 2008)*, volume 353 of *CEUR-WS*, 2008.
9. A. Borgida, D. Calvanese, and M. Rodriguez-Muro. Explanation in the DL-Lite family of Description Logics. In *Proc. of the 7th Int. Conf. on Ontologies, DataBases, and Applications of Semantics (ODBASE 2008)*, volume 5332 of *LNCS*, pages 1440–1457. Springer-Verlag, 2008.
10. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable Description Logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.
11. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. Technical Report CD-TR 91/16, TU Vienna, 1991.
12. T. Eiter, K. Makino, and G. Gottlob. Computational aspects of monotone dualization: A brief survey. *Discrete Applied Mathematics*, 156(11):2035–2049, 2008.
13. M. Horridge, B. Parsia, and U. Sattler. Laconic and precise justifications in OWL. In *Proc. of the 7th Int. Semantic Web Conf., (ISWC 2008)*, volume 5318 of *LNCS*, pages 323–338. Springer-Verlag, 2008.
14. M. Horridge, B. Parsia, and U. Sattler. Explaining inconsistencies in OWL ontologies. In *Proc. of the Third Int. Conf. on Scalable Uncertainty Management, (SUM 2009)*, volume 5785 of *LNCS*, pages 124–137. Springer-Verlag, 2009.
15. D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.
16. A. Kalyanpur, B. Parsia, M. Horridge, and E. Sirin. Finding all justifications of OWL DL entailments. In *Proc. of the 6th Int. Semantic Web Conf., 2nd Asian Semantic Web Conf., (ISWC 2007 + ASWC 2007)*, volume 4825 of *LNCS*, pages 267–280. Springer-Verlag, 2007.
17. T. Meyer, K. Lee, R. Booth, and J. Z. Pan. Finding maximally satisfiable terminologies for the description logic $\mathcal{ALC}$. In *Proc. of the 21st Natonal Conf. on Artificial Intelligence (AAAI 2006)*, pages 269–274. AAAI Press/The MIT Press, 2006.
18. L. A. Nielsen, K. A. Andersen, and D. Pretolani. Finding the K shortest hyperpaths. *Computers and Operations Research*, 32(6):1477-1497, 2005.
19. B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *Proc. of the 14th international conference on World Wide Web (WWW 2005)*, pages 633–640. ACM, 2005.
20. R. Peñaloza and B. Sertkaya. Axiom pinpointing is hard. In *Proc. of the 2009 Int. Workshop on Description Logics (DL2009)*, volume 477 of *CEUR-WS*, 2009.

21. R. Peñaloza and B. Sertkaya. On the complexity of axiom pinpointing in Description Logics. LTCS-Report LTCS-09-04, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2009. See http://lat.inf.tu-dresden.de/research/reports.html.
22. R. Peñaloza and B. Sertkaya. On the complexity of axiom pinpointing in the $\mathcal{EL}$ Family of Description Logics. In *Proc. of the Twelfth Int. Conf. on Principles and Knowledge Representation and Reasoning (KR-10)*. Morgan Kaufmann, 2010. To appear.
23. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
24. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. of the Eighteenth Int. Joint Conf. on Artificial Intelligence (IJCAI'03)*, pages 355–362. Morgan Kaufmann, 2003.
25. J. Y. Yen. Finding K shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.