Complexity of Axiom Pinpointing in the DL-Lite Family of Description Logics

Rafael Peñaloza\textsuperscript{1} and Barış Sertkaya\textsuperscript{2}

Abstract.
We investigate the complexity of axiom pinpointing for different members of the DL-Lite family of Description Logics. More precisely, we consider the problem of enumerating all minimal subsets of a given DL-Lite knowledge base that have a given consequence. We show that for the DL-Lite\textsubscript{core}, DL-Lite\textsubscript{horn} and DL-Lite\textsubscript{horN} fragments such minimal subsets are efficiently enumerable with polynomial delay, but for the DL-Lite\textsubscript{horN} fragment they cannot be enumerated in output polynomial time unless $P = NP$. We also show that interestingly, for the DL-Lite\textsubscript{horN} fragment such minimal sets can be enumerated in reverse lexicographic order with polynomial delay, but it is not possible in the forward lexicographic order since computing the first one is already coNP-hard.

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 sizes, 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 \textit{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 \textit{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 knowledge base (KB) are responsible for a given consequence. In [24] a pinpointing extension of the tableau-based satisfiability algorithm for the DL ALC has been introduced. Later in [19], this approach has been further extended to DLs that are more expressive than ALC. In [17] a pinpointing algorithm for 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 [4]. Similarly, an automata-based general approach for obtaining glass-box pinpointing algorithms has been introduced in [3].

In contrast to the glass-box approach, the idea underlying the \textit{black-box approach} is to make use of the existing highly optimized reasoning algorithms without having to modify them. The most naive 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 results in [16] demonstrate that this approach behaves quite well in practice on realistc 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 [5] it has been shown that a given consequence can have exponentially many explanations (there called MinAs, which stands for \textit{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 a fragment of $\mathcal{EL}$ that only allows for conjunction on both sides of a GCI. In [20, 21] we have investigated the complexity of axiom pinpointing in the propositional Horn fragment, and in the tractable DL $\mathcal{EL}$. We have presented 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 in the dual-Horn setting where the axioms have only 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 [25] a promising method that uses modern conflict-driven SAT solvers for axiom pinpointing in $\mathcal{EL}$ has been presented. The method generates propositional Horn formulas representing the deduction steps performed by a classification algorithm, and manipulates them with the help of a SAT solver for computing MinAs.

In the present work we investigate the complexity of axiom pinpointing in the other prominent 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 [8, 1]. For this family various aspects of finding explanations have already been considered in [7, 6]. 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 unsatisfiable, and why a particular value is not returned as answer to a conjunctive query. Complementary to the work in [7, 6] here we consider the problem of explaining TBox reasoning. We investigate in detail the complexity of enumerating MinAs in a DL-Lite TBox for unsatisfiable, and why a particular value is not returned as answer to a conjunctive query. We define our problem without mentioning a particular DL-Lite fragment but investigate its computational complexity for different fragments in the coming sections separately.

3 Complexity of Enumerating all MinAs

The main problem we consider in the present work is, given a DL-Lite TBox and a consequence that follows from it, compute all MinAs for this consequence in the given TBox.

Definition 1 (MinA). Let \( T \) be a DL-Lite TBox and \( \varphi \) a DL-Lite axiom that follows from it, i.e., \( T \models \varphi \). We call a set \( M \subseteq T \) a minimal axiom set or MinA for \( \varphi \) in \( T \) if \( M \models \varphi \) 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.

Problem: MINA-ENUM
Input: A DL-Lite TBox \( T \), and a DL-Lite axiom \( \varphi \) such that \( T \models \varphi \).
Output: The set of all MinAs for \( \varphi \) in \( T \).

3.1 MinAs in DL-Lite\textsubscript{core} and DL-Lite\textsubscript{krom} TBoxes

We start with a basic observation. In the simplest setting where we can consider MINA-ENUM, \( T \) is a DL-Lite\textsubscript{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 \( T \) becomes just a directed graph, and a MinA for \( A_n \sqsubseteq A_m \) is just a simple path, i.e., a path with no repeated vertices, between the nodes \( A_n \) and \( A_m \). Thus, 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 [26]. This observation has already been mentioned in the works [7, 6], which mainly concentrate on explaining query answering.

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

Example 2. Consider the DL-Lite\textsubscript{core} TBox \( T = \{ A \sqsubseteq \neg \exists r_1, \exists r_2 \sqsubseteq \exists r_1, D \sqsubseteq \exists r_2, D \sqsubseteq \exists r_1, A \sqsubseteq D \} \) and the axiom \( \varphi : A \sqsubseteq \neg D \) which follows from \( T \). 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 \( \varphi \) follows from \( T \). This means that we also need to find out the reasons why \( A \) is unsatisfiable. The only MinA for \( A \sqsubseteq \neg A \) in \( T \) 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 \( \varphi \), thus \( M \) is not a minimal axiom set, i.e., a MinA for \( \varphi \). This 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 \( B_1 \sqsubseteq \neg B_2 \) 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 \( \ell(\varphi) \) stands for the tail, i.e., the left
Algorithm 1: Enumerating all MinAs for $DL$-Lite$_{krom}$, TBoxes

Procedure: ALL-MINAS($T, \psi$) ($T$ a $DL$-Lite$_{krom}$ TBox, $\psi$ an axiom s.t. $T \models \psi$)

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

Procedure: ALL-MINAS-AUX($T, \psi$) ($T$ a $DL$-Lite$_{krom}$ TBox, $\psi$ an axiom, $T \models \psi$)

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

hand side, and $h(\psi)$ stands for the head, i.e., the right hand side, of axiom $\psi$.

Theorem 3. Algorithm 1 solves MINA-ENUM for $DL$-Lite$_{krom}$ TBoxes with polynomial delay.

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 $\psi$ in $T$.

The algorithm is sound. ALL-MINAS-AUX outputs an axiom $\psi_i$ only if using it can be derived. Moreover, as soon as the head and the tail of $\psi$ 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 $\psi$ in $T$. ALL-MINAS additionally checks if the tail of $\psi$ is unsatisfiable, and if this is the case also outputs the MinAs for $t(\psi) \subseteq \lnot t(\psi)$ that do not contain any of the previously output MinAs.

The algorithm is complete. ALL-MINAS-AUX iterates over the axioms in $T$ and searches for the MinA for $\psi$ in a depth-first manner. If $T \models t(\psi) \subseteq \lnot t(\psi)$, then ALL-MINAS additionally searches for MinAs for $t(\psi) \subseteq \lnot t(\psi)$, in the same manner. These are all MinAs for $\psi$ in $T$.

Note that in lines 4 and 7 of the procedure ALL-MINAS-AUX the algorithm checks whether the selected axiom $\psi$ will lead to a MinA. Clearly, for $DL$-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$_{horn}$ TBoxes

Next we show that for $DL$-Lite$_{horn}$ 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$_{horn}$ TBox $T$ a propositional Horn TBox $G_T$ as follows: for every basic concept $B$ create a propositional variable $v_B$; for every axiom $\bigwedge_{i=1}^m B_i \subseteq B$ add the Horn clause $\bigwedge_{i=1}^{m} v_{B_i} \rightarrow v_B$; and for each pair of number restrictions $q_1 < q_2$ appearing in $T$, add the Horn clause $v_{\geq q_1} \rightarrow v_{\geq q_2}$. We will call the latter ones implicit axioms. It is not difficult to see that $T \models \bigwedge_{i=1}^{m} A_i \subseteq C$ iff $G_T \models \bigwedge_{i=1}^{m} v_{A_i} \rightarrow v_C$. Furthermore, $\text{MinA } M$ in $G_T$ gives rise to a MinA in $T$ consisting of all axioms representing non implicit axioms in $M$. However, different MinAs in $G_T$ can give rise to the same MinA in $T$. For instance let $T = \{ A \subseteq \geq 2t, A \supseteq \geq 3t, \geq 1r \subseteq B \}$. Clearly $G_T$ constructed from $T$ as described has three MinAs for $v_A \rightarrow v_B$, but there are only two MinAs for $A \subseteq B$ in $T$. The reason is that the implicit subsumption $\geq 3t \supseteq \geq 1r$ is represented twice in $G_T$: one through the direct edge, and another with a path travelling along $v_{\geq 2r}$. We solve this problem by using immediate MinAs.

Definition 4 (Immediate MinA). Let $T$ be a $DL$-Lite$_{horn}$ TBox. A MinA $M$ in $G_T$ is called immediate if for every implicit axiom $\sigma \in G_T, M \models \sigma$ implies $\sigma \in M$.

Note that there is a one-to-one correspondence between MinAs for $\bigwedge_{i=1}^{m} A_i \subseteq C$ in $T$ and immediate MinAs for $\bigwedge_{i=1}^{m} v_{A_i} \rightarrow v_C$ in $G_T$. Thus, if we can enumerate all immediate MinAs in $G_T$ in polynomial time, we will be able to enumerate also all MinAs in $T$ 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 5 (Valid Ordering). Let $T$ be a propositional Horn TBox, and $\phi = \bigwedge_{i=1}^{m} A_i \rightarrow b$ be an axiom in $T$. We denote the left-hand-side (lhs) of $\phi$ with $T(\phi)$, and its right-hand-side (rhs) with $h(\phi)$, i.e., $T(\phi) := \{ a_1, \ldots, a_n \}$ and $h(\phi) := b$. With $h^{-1}(b)$ we denote the set of axioms in $T$ whose rhs are $b$. Let $M = \{ t_1, \ldots, t_m \}$ be a MinA for $\bigwedge_{a \in A} a \rightarrow c$. We call an ordering $t_1 < \cdots < t_m$ a valid ordering on $M$ if for every $1 \leq i \leq m$, $T(t_i) \subseteq \bigcup \{ h(t_1), \ldots, h(t_{i-1}) \}$ holds.

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 6 ($T_i$). Let $M$ be an immediate MinA in $G_T$ with $|M| = m$, and be a valid ordering on $M$. For each $1 \leq i \leq m$ we obtain a TBox $T_i$ from $G_T$ as follows: if $t_i$ is an implicit axiom, then $T_i = \emptyset$; otherwise, (i) for each $j$ s.t. $i < j \leq m$ and $t_j$ is not an implicit axiom, remove all axioms in $h^{-1}(h(t_j))$ except for $t_i$, i.e., remove all axioms with the same rhs as $t_j$ except for $t_i$ 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 first all non-implicit axioms for removal, and ordering the implicit ones as follows: if $t_1 := (\geq q_1 r) \subseteq (\geq q_2 r)$, and $t_2 := (\geq q'_1 r) \subseteq (\geq q'_2 r)$ are two implicit axioms and $q_1 - q_2 < q'_1 - q'_2$, then $t_1$ appears before $t_2$.

Lemma 7. Let $M$ be an immediate MinA for $\phi$ in $T$, and let $T_1, \ldots, T_m$ be constructed from $T$ and $M$ as in Definition 6. Then, 3 That is, each variable on the lhs of $t_i$ is in $A$, or it is the rhs of a previous axiom.
for every immediate MinA \( \mathcal{N} \) for \( \phi \) in \( T \) 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 \( \phi \) in \( T_i \).

Proof. Let \( t_1 < \ldots < t_m \) be a valid ordering on \( \mathcal{M} \), and \( \mathcal{N} \) an immediate MinA for \( \phi \) in \( T \) 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 \( \mathcal{N} \subseteq T_k \) and \( \mathcal{N} \nsubseteq T_i \) for all \( i \neq k \), \( 1 \leq i \leq m \).

Assume there is an axiom \( t \in \mathcal{N} \) s.t. \( t \notin T_k \). Since \( T_k \) contains all implicit axioms, \( t \) should be one of the non-implicit axioms removed from \( T \) either in step (i) or in step (ii) of Definition 6. 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 \( h(t) = h(t_j) \). 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 \( h(t) \), which contradicts with the fact that \( \mathcal{N} \) is a MinA, and thus it is minimal. Hence, \( \mathcal{N} \subseteq T_k \).

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

Lemma 7 gives an idea of how to compute the remaining MinAs from a given one in the DL-Lite\(_{horn}\) setting. Algorithm 2 describes how we can use this lemma to enumerate all MinAs in a DL-Lite\(_{horn}\) TBox \( T \) by enumerating all immediate MinAs in \( \mathcal{G}_T \).

### Theorem 8. Algorithm 2 solves MINA-ENUM for DL-Lite\(_{horn}\) TBoxes with polynomial delay.

Proof. The algorithm terminates since \( T \) is finite. Soundness is shown as follows: \( Q \) contains initially only the original TBox \( T \). Thus the first output is lexicographically the last MinA in \( T \). By Lemma 7 the MinA that comes just before the last one is contained in exactly one of the \( T_i \)s that are computed and inserted into \( Q \) in lines 8 and 9. In line 3 \( J \) 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 7 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 that takes \( O(n) \) time, where \( n = |T| \). In addition it performs at most \( n \) \( T_i \)s computations, and at most \( n \) insertions into \( \mathcal{Q} \). Each \( T_i \) 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 \( O(2 \cdot n \cdot \log (M) + n + n \cdot n \cdot \log (M)) = O(n^3) \).
**Problem: first-MN**

**Input:** A DL-Lite TBox $T$, an axiom $\varphi$ such that $T \models \varphi$, a MinA $\mathcal{M}$ for $\varphi$ in $T$ and a linear order on $T$.

**Question:** Is $\mathcal{M}$ the first MinA w.r.t. the lexicographic order induced by the linear order?

**Theorem 11.** FIRST-MINA is conp-complete for DL-Litehorn TBoxes.

**Proof.** The problem is clearly in conP, so it remains only to be shown that it is conP-hard. We do this via a reduction from the following NP-complete problem [10].

**Problem: HORN-RELEVANCE**

**Input:** Two sets of propositional variables $H$ and $M$, a set $C$ of definite Horn clauses over $H \cup M$, and a propositional variable $p \in H$.

**Question:** Is there a minimal $G \subseteq H$ such that $G \cup C \models M$ and $p \in G$?

Let an instance of HORN-RELEVANCE be given with $H, M, C$ and $p$, and assume w.l.o.g. that $H \cup C \models M$. We construct an instance of first-MINA as follows: for each propositional variable $m \in H \cup M$, we introduce a concept name $A_m$, and additionally, two fresh concept names $A_s, A_t$ and construct the DL-Litehorn TBox.

$$T := \{ A_s \subseteq A_t \mid h \in H \} \cup \left( \bigcup_{i=1}^{k} A_{q_i} \subseteq A_r \mid q_i \rightarrow r \in C \} \cup \{ \prod_{m \in M} A_m \subseteq A_t \}. \right.$$ 

It is easy to see that for $\varphi := A_s \subseteq A_t, T \models \varphi$. Let $\mathcal{M}$ be a MinA for $\varphi$ in $T$, such that, w.l.o.g. $A_s \subseteq A_t \notin \mathcal{M}$. Define a linear ordering on the axioms in $T$ as follows: first appears the axiom $A_s \subseteq A_t$, then all the axioms in $\mathcal{M}$ in any order, and finally all the other axioms in any order. Then $T, \mathcal{M}, \varphi$ forms an instance of first-MINA, and is constructed in polynomial time. Furthermore, $\mathcal{M}$ is lexicographically the first MinA w.r.t. the defined order if $G$ is no $G \subseteq H$ with $p \in G$ such that $G \cup C \models M$. Hence, FIRST-MINA is conP-hard.

Since finding lexicographically the first MinA is already intractable, we cannot expect to have an algorithm that enumerates all MinAs in lexicographical order with polynomial delay. The following is an immediate consequence of Theorem 11.

**Corollary 12.** For DL-Litehorn TBoxes, MinAs cannot be enumerated in lexicographical order with polynomial delay, unless $P = \text{NP}$.

### 3.3 MinAs in DL-Litebool 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-Litebool 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. In particular, this implies that MinAs cannot be enumerated with polynomial delay in the DL-Litebool 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-MIN**

**Input:** A DL-Lite TBox $T$ and an axiom $\varphi$ such that $T \models \varphi$, and a set of TBoxes $\mathcal{T} \subseteq \{ T \}$.

**Question:** Is $\mathcal{T}$ precisely the set of all MinAs for $\varphi$ in $T$?

Because, as Proposition 13 shows, if ALL-MINAs cannot be decided in polynomial time for DL-Litebool TBoxes, then MINA-ENUM cannot be solved in output polynomial time for DL-Litebool TBoxes. Its proof is based on a generic argument, which can also be found in [11] Theorem 4.5, but for the sake of completeness we present it here once more.

**Proposition 13.** For DL-Litebool TBoxes, if ALL-MINAs cannot be decided in polynomial time, then MINA-ENUM cannot be solved in output-polynomial time.

**Proof.** Assume we have an algorithm $A$ that solves MINA-ENUM for DL-Litebool TBoxes in output-polynomial time. Let its runtime be bounded by a polynomial $p(|S, O|) S$ where $S$ denotes the size of the input TBox and $O$ denotes the size of the output, i.e., the set of all MinAs.

In order to decide ALL-MINAs for an instance given by the DL-Litebool TBox $T$, $\varphi$, and $\mathcal{T} \subseteq \{ T \}$, we construct another algorithm $A'$ that works as follows: it runs $A$ on $T$ and $\varphi$ for at most $p(|T|, |\mathcal{T}|)$-many steps. If $A$ terminates within this many steps, then $A'$ compares the output of $A$ with $\mathcal{T}$ 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 $p(|T|, |\mathcal{T}|)$-many steps, this implies that there is at least one MinA that is not contained in $\mathcal{T}$, so $A'$ returns no. It is easy to see that the runtime of $A'$ is bounded by a polynomial in $|T|$ and $|\mathcal{T}|$, that is $A'$ decides ALL-MINAs for DL-Litebool TBoxes in polynomial time.

The proposition shows that the complexity of ALL-MINAs is indeed closely related to the complexity of MINA-ENUM. Next we show that this problem is conP-hard for DL-Litebool TBoxes.

**Lemma 14.** ALL-MINAs is conP-hard for DL-Litebool TBoxes. This already holds if the axioms in $T$ are of the form $A \subseteq C$ where $A$ is a concept name and $C$ a general concept.

**Proof.** We present a reduction from the conP-hard problem [9, 5]:

**Problem: all-MV**

**Input:** A monotone Boolean formula $\phi$ and a set $Y$ of minimal valuations satisfying $\phi$.

**Question:** Is $Y$ precisely the set of minimal valuations satisfying $\phi$?

Let $\phi, Y$ be an instance of ALL-MV. We introduce a concept name $A_p$ for each propositional variable $p$ appearing in $\phi$ and two additional concept names $A_0, A_1$. From $\phi$ we construct the general concept $C_0$ by changing each conjunction $\land$ to $\sqcap$, each disjunction $\lor$ to $\sqcup$ and each propositional variable $p$ to $\neg p$.

Using these we construct the TBox $T := \{ A_1 \subseteq \neg C_0 \} \cup \{ A_p \subseteq \neg A_0 \mid p \in \phi \}$ and the set of MinAs $\mathcal{T} := \{ \{ A_1 \subseteq \neg C_0 \} \cup \{ B_p \subseteq \neg A_0 \mid p \in \phi \} \mid \forall \varphi \}$. It is easy to see that $T$ and $\mathcal{T}$ indeed form an instance of ALL-MINAs for the axiom $A_0 \subseteq \neg A_1$. Furthermore, $\mathcal{T}$ is the set of all MinAs for $A_0 \subseteq \neg A_1$ if $Y$ is the set of all minimal valuations satisfying $\phi$.

The following is an immediate consequence of Proposition 13 and Lemma 14.

**Corollary 15.** For DL-Litebool TBoxes all MinAs cannot be computed in output-polynomial time unless $P = \text{NP}$.

---

4 We use the abbreviation $X \cup Y$ for $\neg (\neg X \cap \neg Y)$. 

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-Lite\textsubscript{core}, DL-Lite\textsubscript{horn} and DL-Lite\textsubscript{horn}\textsubscript{H\textsubscript{N}} TBoxes MinAs are efficiently enumerable with polynomial delay, but for DL-Lite\textsubscript{boost} they cannot be enumerated in output-polynomial time unless P = \textsc{NP}. We have also shown that interestingly, for DL-Lite\textsubscript{horn} TBoxes MinAs can be enumerated in reverse lexicographic order with polynomial delay but, it is not possible in the forward lexicographic order since computing the first MinA is already co\textsc{NP}-hard. This hardness result holds already for DL-Lite\textsubscript{horn} TBoxes. 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. Table 1 shows a summary of our results.

Finding explanations for query answering and ABox reasoning has already been considered in [7, 6]. 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.

<table>
<thead>
<tr>
<th>TBox</th>
<th>MinA-Enum</th>
<th>ALL-MINAS</th>
<th>MINA-ENUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL-Lite\textsubscript{core}</td>
<td>poly delay</td>
<td>poly</td>
<td>unordered</td>
</tr>
<tr>
<td>DL-Lite\textsubscript{horn}</td>
<td>poly delay</td>
<td>poly delay</td>
<td>poly delay</td>
</tr>
<tr>
<td>DL-Lite\textsubscript{horn}\textsubscript{H\textsubscript{N}}</td>
<td>coNP-c</td>
<td>poly not poly delay</td>
<td>poly delay</td>
</tr>
<tr>
<td>DL-Lite\textsubscript{horn}\textsubscript{H\textsubscript{N}}</td>
<td>coNP-c</td>
<td>poly not poly delay</td>
<td>poly delay</td>
</tr>
<tr>
<td>DL-Lite\textsubscript{boost}</td>
<td>coNP-h</td>
<td>coNP-h</td>
<td>not output poly</td>
</tr>
</tbody>
</table>

Table 1. Summary of the results

REFERENCES