Axiom Pinpointing is Hard

Rafael Peñaloza and Barış Sertkaya*

Theoretical Computer Science TU Dresden, Germany
{penaloza,sertkaya}@tcs.inf.tu-dresden.de

Abstract. We investigate the complexity of several decision, enumeration and counting problems in axiom pinpointing in Description Logics. We prove hardness results that already hold for the propositional Horn fragment. We show that for this fragment, unless P = NP, all minimal subsets of a given TBox that have a given consequence, i.e. MinAs, cannot be enumerated in a specified lexicographic order with polynomial delay. Moreover, we show that recognizing the set of all MinAs is at least as hard as recognizing the set of all minimal transversals of a given hypergraph, however whether this problem is intractable remains open. We also show that checking the existence of a MinA that does not contain any of the given sets of axioms, as well as checking the existence of a MinA that contains a specified axiom are both NP-hard. In addition we show that counting all MinAs and counting the MinAs that contain a certain axiom are both #P-hard.

1 Introduction

As the number and the size of DL-based ontologies grow, tools that support knowledge engineers in improving and maintaining the quality of these ontologies become more important. In real world applications often the knowledge engineer not only wants to know whether her knowledge base has a certain (unwanted) consequence or not, but also wants to know the reasons of this consequence. Even for knowledge bases 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., finding minimal subsets of the original knowledge base that have the given consequence is called *axiom pinpointing* in the literature.

Axiom pinpointing has been considered by several authors. In [18] Schlobach and Cornet have described an extension of the tableau-based satisfiability algorithm for $\mathcal{ALC}$ that given an $\mathcal{ALC}$ knowledge base and a consequence of this knowledge base computes all minimal subsets of the original knowledge base that have the given consequence. The same problem had been addressed earlier in [1] in the context of extending DLs by default rules. Later in [15] Parsia et al. have extended the approach in [18] to more expressive DLs, and in [14] Lee et al. have extended the approach in [1] to $\mathcal{ALC}$ knowledge bases with GCIs. In contrast to

* Supported by the German Research Foundation (DFG) under grant BA 1122/12-1

these works on axiom pinpointing in expressive DLs, in [2] Baader et al. have addressed axiom pinpointing in the less expressive DL $\mathcal{EL}^+$. They have investigated the complexity of finding minimal subsets of a given $\mathcal{EL}^+$ TBox (there called *MinA*s) that have a given consequence. They have shown that checking the existence of a MinA within a specified cardinality bound is NP-complete. Moreover, they have shown that a given consequence can have exponentially many MinAs in a given $\mathcal{EL}^+$ TBox. Given this fact, it is definitely not possible to compute all MinAs in polynomial time. Baader et al. have considered this problem in a setting where the TBox has a so-called static part which is always present, and a refutable part from which MinAs are computed. They have shown that in this setting computing all MinAs cannot be performed in output polynomial time unless P = NP. In fact, all these hardness results are shown for the propositional Horn fragment (there called $\mathcal{HL}$), that allows for only conjunction and GCIs.

In the present paper we investigate the complexity of several decision, enumeration and counting problems related to MinAs. Following [2], we show our hardness results for the logic $\mathcal{HL}$. First we examine whether MinAs can be enumerated in a specified lexicographic order with polynomial delay. It turns out that even for $\mathcal{HL}$ already generating the lexicographically first MinA is intractable, thus unless P = NP, MinAs cannot be enumerated in a specified lexicographic order with polynomial delay. Next we examine whether all MinAs can be computed in output polynomial time without the static part considered in the setting in [2]. We show that for this setting recognizing the set of all MinAs is at least as hard as recognizing the set of all minimal transversals of a given hypergraph, which is a prominent open problem [8] in hypergraph theory [4]. However, whether this problem is intractable remains open. Next we investigate the problem of existence of a MinA that does not contain any of the given sets of axioms. This problem, which can be useful in practical applications where one wants to avoid certain combinations of axioms in the MinAs, turns out to be NP-hard. We also consider the problem of checking the existence of a MinA that contains a specified axiom. We show that this problem is also NP-hard. Finally, we show that both counting all MinAs and counting the MinAs that contain a certain axiom are #P-hard problems.

2 Complexity of computing all MinAs

In the present section we examine the complexity of computing all MinAs. The hardness results we show in this work actually already hold for the propositional Horn fragment $\mathcal{HL}$. For this reason, from now on we are going to formulate our problems for $\mathcal{HL}$ TBoxes. Note that $\mathcal{HL}$ is a sublogic of $\mathcal{EL}$, which implies that the lower bounds for the complexity found for $\mathcal{HL}$ also hold for $\mathcal{EL}$ and $\mathcal{EL}^+$. We start with formally defining some basic notions.

Definition 1. *Let $\mathcal{T}$ be an $\mathcal{HL}$ TBox and C and D be two $\mathcal{HL}$ concept descriptions such that $C \sqsubseteq_{\mathcal{T}} D$. We call a set $\mathcal{T}' \subseteq \mathcal{T}$ a* minimal axiom set *or* MinA *for $\mathcal{T}$ w.r.t. $C \sqsubseteq D$ if $C \sqsubseteq_{\mathcal{T}'} D$ and it is minimal w.r.t. set inclusion. In this case we also say that $\mathcal{T}$'* explains *$C \sqsubseteq D$.*

Based on this, our problem is formulated as follows:

Problem: MINA-ENUM
Input: An $\mathcal{HL}$ TBox $\mathcal{T}$ and a GCI $C \sqsubseteq D$ such that $\mathcal{T} \models C \sqsubseteq D$.
Output: The set of all MinAs for $\mathcal{T}$ w.r.t. $C \sqsubseteq D$.

2.1 Enumerability with polynomial delay

In [2] it has been shown that there can be exponentially many MinAs for a given $\mathcal{HL}$ TBox w.r.t. a given consequence of this TBox. Given this fact, it is definitely not possible to solve MINA-ENUM in polynomial time. In complexity theory, for analyzing the performance of enumeration algorithms where the number of solutions can be exponential in the size of the input, one considers other measures. One such measure is the time the algorithm spends between two consequent solutions. An algorithm is said to run with *polynomial delay* [13] 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.

In the following we investigate whether it is possible to enumerate all MinAs in a specified lexicographic order with polynomial delay. The lexicographic order we will use is defined as follows:

Definition 2. *Let the elements of a set S be linearly ordered. This order induces another 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.*

Problem: FIRST-MINA
Input: An $\mathcal{HL}$ TBox $\mathcal{T}$, a GCI $C \sqsubseteq D$ such that $\mathcal{T} \models C \sqsubseteq D$, a set $\mathcal{S} \subseteq \mathcal{T}$ such that $\mathcal{S} \models C \sqsubseteq D$, and a linear order on $\mathcal{T}$.
Question: Is $\mathcal{S}$ the first MinA w.r.t. the lexicographic order induced by the given linear order?

Theorem 1. FIRST-MINA *is* coNP*-complete.*

Proof. The problem is in coNP since if $\mathcal{S}$ is not the lexicographically first MinA, a proof of this can be given by nondeterministically guessing a subset of $\mathcal{T}$ and verifying in polynomial time that it is a MinA that is lexicographically smaller than $\mathcal{S}$.

In order to show coNP-hardness, we are going to give a reduction from the problem of checking whether a given maximal independent set is the lexicographically last maximal independent set of a given graph. Recall that a *maximal independent set* of a graph $\mathcal{G} = (V, \mathcal{E})$ is a subset $V' \subseteq V$ of the vertices such that no two vertices in V' are joined by an edge in $\mathcal{E}$, and each vertex in $V \setminus V'$ is joined by an edge to some vertex in V'. This problem has been shown to be coNP-complete in [13].

Problem: LAST MAXIMAL INDEPENDENT SET (LAST-MIS)
Input: A graph $\mathcal{G} = (V, \mathcal{E})$, a maximal independent set $S \subseteq V$, and a linear order on V.
Question: Is S the last maximal independent set w.r.t. the lexicographic order induced by the given linear order?

Let an instance of LAST-MIS be given with the graph $\mathcal{G} = (V, \mathcal{E})$ and the maximal independent set S. From $\mathcal{G}$ and S we construct an instance of FIRST-MINA as follows: For every $v \in V$ we introduce a concept name P_v, for every edge $E \in \mathcal{E}$ we introduce a concept name P_E, and finally one more concept name A. Using these we construct the TBox

$$\mathcal{T} := \{P_v \sqsubseteq \bigsqcap_{v \in E, E \in \mathcal{E}} P_E \mid v \in V\} \cup \{\bigsqcap_{E \in \mathcal{E}} P_E \sqsubseteq A\},$$

and the GCI $\bigsqcap_{v \in V} P_v \sqsubseteq A$ that obviously follows from $\mathcal{T}$. Additionally, by using S we construct a set $\mathcal{S} \subseteq \mathcal{T}$ as:

$$\mathcal{S} := \{P_v \sqsubseteq \bigsqcap_{v \in E, E \in \mathcal{E}} P_E \mid v \in (V \setminus S)\} \cup \{\bigsqcap_{E \in \mathcal{E}} P_E \sqsubseteq A\}.$$

Note that $\mathcal{T}$ contains exactly $|V| + 1$ axioms. We order these axioms such that an axiom with premise P_v comes before the axiom with premise $P_{v'}$ if and only if the vertex v comes before the vertex v' in the originally given linear order on V. Finally we place the axiom $\bigsqcap_{E \in \mathcal{E}} P_E \sqsubseteq A$ as the last one. It is easy to see that this construction indeed creates an instance of FIRST-MINA and the TBox $\mathcal{T}$, as well as the MinA $\mathcal{S}$ and the GCI $\bigsqcap_{v \in V} P_v \sqsubseteq A$ can be created in time polynomial in the sizes of $\mathcal{G}$ and S. We claim that S is lexicographically the last maximal independent set if and only if $\mathcal{S}$ is the lexicographically first MinA.

($\Rightarrow$) Assume S is the lexicographically last maximal independent set. Then $V \setminus S$ contains at least one vertex from every edge (i.e., it is a vertex cover), since otherwise S would not be an independent set. Thus every P_E, for $E \in \mathcal{E}$, appears on the righthand side of at least one axiom in $\mathcal{S}$. That is $\bigsqcap_{v \in V} P_v \sqsubseteq_{\mathcal{S}} \bigsqcap_{E \in \mathcal{E}} P_E$, thus $\bigsqcap_{v \in V} P_v \sqsubseteq_{\mathcal{S}} A$, i.e., $\mathcal{S}$ explains this axiom. Since S is maximal, $V \setminus S$ is minimal, and thus $\mathcal{S}$ is minimal, i.e., $\mathcal{S}$ is a MinA. Moreover, it is the lexicographically first one since S is the lexicographically last maximal independent set.

($\Leftarrow$) Assume $\mathcal{S}$ is the lexicographically first MinA. Then every P_E, for $E \in \mathcal{E}$, appears on the righthand side of at least one axiom in $\mathcal{S}$, since otherwise it would not be an explanation. That is, $V \setminus S$ contains at least one vertex from every edge. Then S contains at most one vertex from every edge, i.e., it is an independent set. Since $\mathcal{S}$ is minimal $V \setminus S$ is minimal, and thus S is maximal, i.e., S is a maximal independent set. Moreover it is the lexicographically last one since $\mathcal{S}$ is the lexicographically first MinA. $\square$

Theorem 1 has the following consequence: Since generating the lexicographically first MinA is already intractable, unless P = NP, all MinAs cannot be enumerated in lexicographic order with polynomial delay.

Corollary 1. *Unless* P = NP*, there is no algorithm that enumerates MinAs in a given lexicographic order with polynomial delay.*

Interestingly, it is easy to generate the lexicographically last MinA for a given TBox w.r.t. a given GCI. We start with the axioms in the specified order, and remove an axiom if the resulting set of axioms still explains the given GCI. Once we have scanned all axioms in this way, the resulting set of axioms is a MinA, and it is the lexicographically last one. For a TBox of size n, i. e. having n axioms, this operation requires at most n subsumption tests each of which can be done in polynomial time for $\mathcal{EL}$ and $\mathcal{EL}^+$. Nonetheless, it is still unclear whether this fact can be used to enumerate all MinAs in the *reverse* lexicographical order.

2.2 Computability in output polynomial time

One other measure for analyzing the performance of enumeration algorithms is to take into account not only the size of the input, but also the size of the output. An algorithm is said to run in *output polynomial time* (or *polynomial total time*) [13] if it outputs all solutions in time polynomial in the size of the input *and the output*. One advantage of an output polynomial algorithm is that it runs in polynomial time (in the size of the input) when there are only polynomially many solutions.

In the following we investigate whether MINA-ENUM can be solved in output polynomial time. For solving this enumeration problem, the following decision problem has crucial importance:

Problem: ALL-MINAS
Input: An $\mathcal{HL}$ TBox $\mathcal{T}$, a GCI $C \sqsubseteq D$ such that $\mathcal{T} \models C \sqsubseteq D$, and $T \subseteq \mathscr{P}(\mathcal{T})$.
Question: Is T precisely the set of all MinAs of $\mathcal{T}$ w.r.t. $C \sqsubseteq D$?

As Proposition 1 below shows, if this problem cannot be decided in polynomial time, then unless P = NP, MINA-ENUM cannot be solved in output polynomial time.

Proposition 1. *If* ALL-MINAS *cannot be decided in polynomial time, then unless* P = NP*,* MINA-ENUM *cannot be solved in output-polynomial time.*

Proof. Assume that 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 TBox 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 the TBox $\mathcal{T}$, the GCI $C \sqsubseteq D$, and a set $T \subseteq \mathscr{P}(\mathcal{T})$, we construct another algorithm A' that works as follows: it runs A on $\mathcal{T}$ and $C \sqsubseteq D$ for at most $p(|\mathcal{T}|, |T|)$-many steps. If A terminates within this many steps, then A' compares the output of A with 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 $p(|\mathcal{T}|, |T|)$-many steps, this implies that there is at least one MinA that is not contained in T, so A' returns *no*. It is easy to see that the runtime of A' is bounded by a polynomial in $|\mathcal{T}|$ and $|T|$, that is A' decides ALL-MINAS in time polynomial in the size of the input. □

The proposition shows that determining the complexity of ALL-MINAS is indeed crucial for determining the enumeration complexity of MINA-ENUM. It is not difficult to see that ALL-MINAS is in coNP. Given an instance of ALL-MINAS, a nondeterministic algorithm can guess a subset of $\mathcal{T}$ that is not in T, and in polynomial time verify that this is a MinA, thus T is *not* the set of all MinAs. In the following we show that ALL-MINAS is at least as hard as recognizing the set of all minimal transversals of a given hypergraph. However, whether it is coNP-hard remains unfortunately open.

First we briefly recall some notions of 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 $E_j \subseteq V$. Following the convention in [4], we assume 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 $\mathcal{H}$ constitute 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 (see e.g. [9]). 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 so far neither a polynomial time algorithm has been found, nor has it been proved to be coNP-hard. In a landmark paper [10] 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. It is conjectured that this problem, together with several computationally equivalent problems, forms a class properly contained between P and coNP [10].

Theorem 2. ALL-MINAS *is* TRANS-HYP*-hard.*

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 concept name P_v, for every edge $E \in \mathcal{E}_\mathcal{H}$ of $\mathcal{H}$ a concept name P_E, and finally one additional concept name A. For a set of vertices $F \subseteq V$, we define the TBox

$$\mathcal{T}_F := \{P_v \sqsubseteq \bigsqcap_{v \in E, E \in \mathcal{E}_\mathcal{H}} P_E \mid v \in F\} \cup \{\bigsqcap_{E \in \mathcal{E}_\mathcal{H}} P_E \sqsubseteq A\}.$$

Using these we construct the TBox $\mathcal{T} := \mathcal{T}_V$, a set of TBoxes $T := \{\mathcal{T}_F \mid F \in \mathcal{E}_\mathcal{G}\} \subseteq \mathscr{P}(\mathcal{T})$, and the GCI $\bigsqcap_{v \in V} P_v \sqsubseteq A$ that obviously follows from $\mathcal{T}$.

It is easy to see that this construction indeed creates an instance of ALL-MINAS, and the TBox $\mathcal{T}$, as well as the set T and the GCI $\bigsqcap_{v \in V} P_v \sqsubseteq A$ can be constructed 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 T is precisely the set of all MinAs. Note that $\bigsqcap_{E \in \mathcal{E}_{\mathcal{H}}} P_E \sqsubseteq A$ is the only axiom in $\mathcal{T}$ that has A in its right-hand side; this implies that every MinA must contain this axiom. Hence, every MinA is of the form $\mathcal{T}_F$ for some $F \subseteq V$. To prove our claim, it suffices to show that a set of vertices $F \subseteq V$ is a minimal transversal of $\mathcal{H}$ if and only if the TBox $\mathcal{T}_F$ is a MinA.

($\Rightarrow$) Assume that F is a minimal transversal of $\mathcal{H}$. By definition F satisfies $F \cap E \neq \emptyset$ for every $E \in \mathcal{E}_H$. Then $\mathcal{T}_F$ is an axiom set explaining the above GCI. Moreover, it is minimal since F is minimal. Thus, it is indeed a MinA for $\mathcal{T}$ w.r.t. $\bigsqcap_{v \in V} P_v \sqsubseteq A$.

($\Leftarrow$) Now assume that $\mathcal{T}_F$ is a MinA. Then for every P_E (where $E \in \mathcal{E}_{\mathcal{H}}$) there is at least one axiom in $\mathcal{T}_F$ with lefthand side P_v (where $v \in F$ and $F \in \mathcal{E}_G$) such that P_v is subsumed by P_E. That is F intersects every $E \in \mathcal{E}_{\mathcal{H}}$, i.e., F is a transversal. Moreover, it is minimal since the original set of axioms $\mathcal{T}_F$ is a *minimal* axiom set.

It is not difficult to see that from the above it follows that $\mathcal{G}$ is the transversal hypergraph of $\mathcal{H}$ if and only if $T := \{\mathcal{T}_F \mid F \in \mathcal{E}_{\mathcal{G}}\}$ is the set of all MinAs. $\square$

3 Preferred and unwanted axioms

Next we consider the problem of computing MinAs that contain a preferred axiom and MinAs that do not contain some unwanted combination of axioms. First we investigate the problem of existence of a MinA that does not contain any of the given sets of axioms. This problem can be useful in applications where one wants to avoid certain combinations of axioms in the MinAs. It is defined as follows:

Problem: ADD-MINA
Input: An $\mathcal{HL}$ TBox $\mathcal{T}$, a GCI $C \sqsubseteq D$ such that $\mathcal{T} \models C \sqsubseteq D$, and $T \subseteq \mathscr{P}(\mathcal{T})$.
Question: Is there a MinA $\mathcal{T}'$ that satisfies $\mathcal{S} \not\subseteq \mathcal{T}'$ for every $\mathcal{S} \in T$?

Theorem 3. ADD-MINA *is* NP-*complete.*

Proof. The problem is clearly in NP. A nondeterministic algorithm for solving it first guesses a $\mathcal{T}' \subseteq \mathcal{T}$, then tests in polynomial time whether $\mathcal{T}'$ is a MinA that does not contain any of the $\mathcal{S}$ in T.

Hardness is shown by a reduction from the following NP-hard problem [12]:

Problem: HYPERGRAPH 2-COLORING
Input: A hypergraph $\mathcal{H} = (V, \mathcal{E})$.
Question: Is $\mathcal{H}$ 2-colorable, i.e., is there a $W \subseteq V$ such that for all $E \in \mathcal{E}$, $W \cap E \neq \emptyset$ and $(V \setminus W) \cap E \neq \emptyset$?

Let an instance of HYPERGRAPH 2-COLORING be given with the hypergraph $\mathcal{H} = (V, \mathcal{E})$. From $\mathcal{H}$ we construct an instance of ADD-MINA as follows: just like in the construction in the proof of Theorem 2, we introduce a concept name P_v for every $v \in V$, a concept name P_E for every edge $E \in \mathcal{E}$, and finally one more concept name A. We also construct exactly the same TBox $\mathcal{T}$ and the GCI $\bigsqcap_{v\in V} P_v \sqsubseteq A$ constructed there. In addition to these, for an edge $E \in \mathcal{E}$ we define the set

$$\mathcal{V}_E := \{P_v \sqsubseteq \bigsqcap_{v\in F, F\in\mathcal{E}} P_F \mid v \in E\} \cup \{\bigsqcap_{F\in\mathcal{E}} P_F \sqsubseteq A\},$$

and using this construct the set

$$T := \{\mathcal{V}_E \mid E \in \mathcal{E}\}.$$

It is easy to see that the TBox $\mathcal{T}$, the GCI above and the set T can be constructed in time polynomial in the size of $\mathcal{H}$. We claim that $\mathcal{H}$ is 2-colorable if and only if there is a MinA $\mathcal{T}'$ for $\mathcal{T}$ w.r.t. $\bigsqcap_{v\in V} P_v \sqsubseteq A$ such that $\mathcal{T}'$ satisfies $\mathcal{S} \not\subseteq \mathcal{T}'$ for every $\mathcal{S} \in T$.

($\Rightarrow$) Assume $\mathcal{H}$ is 2-colorable. Then there is a $W \subseteq V$ such that $W \cap E \neq \emptyset$ and $(V \setminus W) \cap E \neq \emptyset$ for every $E \in \mathcal{E}$, i.e., both W and its complement are transversals of $\mathcal{H}$. Assume w.l.o.g. that W is minimal. Using W we define the set $\mathcal{T}_W := \{P_v \sqsubseteq \bigsqcap_{v\in E, E\in\mathcal{E}} P_E \mid v \in W\} \cup \{\bigsqcap_{E\in\mathcal{E}} P_E \sqsubseteq A\}$ and claim that this is the MinA we are looking for. Since W is a transversal every P_E, for $E \in \mathcal{E}$, appears on the righthand side of at least one axiom in $\mathcal{T}_W$. That is $\bigsqcap_{v\in V} P_v \sqsubseteq_{\mathcal{T}_W} \bigsqcap_{E\in\mathcal{E}} P_E$, thus $\bigsqcap_{v\in V} P_v \sqsubseteq_{\mathcal{T}_W} A$, i.e., $\mathcal{T}_W$ explains this axiom. $\mathcal{T}_W$ is minimal since W is minimal. Moreover, since $V \setminus W$ is also a transversal, every edge $E \in \mathcal{E}$ contains at least one vertex v that is *not* in W. Thus, every $\mathcal{V}_E \in T$ contains at least one axiom $P_v \sqsubseteq \bigsqcap_{v\in F, F\in\mathcal{E}} P_F$ that is *not* in $\mathcal{T}_W$. That is $\mathcal{T}_W$ is a MinA that is not a superset of any $\mathcal{V}_E \in T$.

($\Leftarrow$) Assume there is a MinA $\mathcal{T}'$ that is not a superset of any $\mathcal{V}_E \in T$. Define the set $W_{\mathcal{T}'} := \{v \mid P_v \sqsubseteq \bigsqcap_{v\in E, E\in\mathcal{E}} P_E \in \mathcal{T}'\}$. Since $\mathcal{T}'$ explains the constructed GCI, for every $E \in \mathcal{E}$ it contains at least one axiom in whose righthandside P_E occurs. That is, $W_{\mathcal{T}'}$ intersects every $E \in \mathcal{E}$. Since $\mathcal{T}'$ is not a superset of any $\mathcal{V}_E \in T$, every such $\mathcal{V}_E$ contains at least one axiom that is *not* in $\mathcal{T}'$. This means that every $E \in \mathcal{E}$ contains at least one vertex that is *not* in $W_{\mathcal{T}'}$. That is, $V \setminus W_{\mathcal{T}'}$ intersects every $E \in \mathcal{E}$. Thus we have shown that $W_{\mathcal{T}'}$ is a 2-coloring of $\mathcal{H}$. □

Next we consider the dual problem, which is checking the existence of a MinA that contains a certain axiom. This problem can be useful in applications where one is interested in MinAs that contain a specific axiom, and is known as the relevance problem:

Problem: MINA-RELEVANCE
Input: An $\mathcal{HL}$ TBox $\mathcal{T}$, a GCI $C \sqsubseteq D$ such that $\mathcal{T} \models C \sqsubseteq D$, and an axiom $t \in \mathcal{T}$.
Question: Is there a MinA $\mathcal{S}$ of $\mathcal{T}$ w.r.t. $C \sqsubseteq D$ such that $t \in \mathcal{S}$?

If one could efficiently solve the relevance problem, then black-box techniques for computing the set of all MinAs (see, e.g. [3, 20]) based on Reiter's Hitting Set Tree algorithm [16] could benefit from Rymon's further optimizations [17]. As we will show next, this problem is also NP-hard.

Theorem 4. MINA-RELEVANCE *is* NP-*complete.*

Proof. The problem is clearly in NP. A nondeterministic algorithm for solving it first guesses a subset of $\mathcal{T}$, then tests in polynomial time whether it is a MinA containing t. For showing hardness we are going to give a reduction from the following NP-complete problem in [11, 7]:

Problem: HORN-RELEVANCE
Input: Two sets of propositional variables H and M, a set T of definite Horn clauses over $H \cup M$, and a propositional variable $p \in H$.
Question: Is there a minimal $H' \subseteq H$ such that $H' \cup T \models M$ and $p \in H'$?

Recall that a definite Horn clause is a Horn clause with exactly one positive literal, i.e., the elements of T are formulae of the form $\bigwedge_{g \in G} g \to f$, where $G \subseteq H \cup M$ and $f \in H \cup M$. Given an instance of HORN RELEVANCE we construct an instance of MINA-RELEVANCE as follows: for every propositional variable $h \in H$ we introduce a concept name P_h, for every propositional variable $m \in M$ we introduce a concept name P_m, and finally two more fresh concept names A and B. Using these we construct the TBox

$$\mathcal{T} := \{A \sqsubseteq P_h \mid h \in H\} \cup \{\bigsqcap_{g \in G} P_g \sqsubseteq P_f \mid \bigwedge_{g \in G} g \to f \in T\} \cup \{\bigsqcap_{m \in M} P_m \sqsubseteq B\}$$

where $G \subseteq H \cup M$ and $f \in H \cup M$. Using A and B we construct the GCI $A \sqsubseteq B$. This construction indeed creates an instance of MINA-RELEVANCE and it can be done in polynomial time. We claim that there is a minimal $H' \subseteq H$ such that $H' \cup T \models M$ and $p \in H'$ if and only if there is a MinA $\mathcal{S}$ of $\mathcal{T}$ w.r.t. $A \sqsubseteq B$ such that $A \sqsubseteq P_p \in \mathcal{S}$.

($\Rightarrow$) Assume there is a minimal $H' \subseteq H$ such that $H' \cup T \models M$ and $p \in H'$. Using H' define the set

$$\mathcal{T}_{H'} := \{A \sqsubseteq P_h \mid h \in H'\} \cup \{\bigsqcap_{g \in G} P_g \sqsubseteq P_f \mid \bigwedge_{g \in G} g \to f \in T\} \cup \{\bigsqcap_{m \in M} P_m \sqsubseteq B\}.$$

$\mathcal{T}_{H'}$ explains $A \sqsubseteq B$ since $H' \cup T \models M$. It is minimal since H' is minimal, and $A \sqsubseteq P_p \in \mathcal{T}_{H'}$ since $p \in H'$.

($\Leftarrow$) Assume there is a MinA $\mathcal{S}$ such that $A \sqsubseteq P_p \in \mathcal{S}$. Since it explains $A \sqsubseteq B$ it contains the axiom $\bigsqcap_{m \in M} P_m \sqsubseteq B$, contains axioms of the form $\bigsqcap_{g \in G} P_g \sqsubseteq P_f$ such that for every $m \in M$ the concept name P_m occurs on the righthand side of at least one such axiom. Additionally $\mathcal{S}$ contains axioms of the form $A \sqsubseteq P_h$ such that $A \sqsubseteq_{\mathcal{S}} \bigsqcap_{m \in M} P_m$. Then the set $S = \{h \mid A \sqsubseteq P_h \in \mathcal{S}\}$ satisfies $S \cup T \models M$. Moreover $p \in S$ since $A \sqsubseteq P_p \in \mathcal{S}$, and S is minimal since $\mathcal{S}$ is minimal. □

4 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. Next we consider this counting problem. It turns out that this is hard for the counting complexity class #P [21], which is defined as the class of functions counting the number of accepting paths of nondeterministic Turing machines.

Problem: #MINA
Input: An $\mathcal{HL}$ TBox $\mathcal{T}$ and a GCI $C \sqsubseteq D$ such that $\mathcal{T} \models C \sqsubseteq D$.
Output: The number of all MinAs of $\mathcal{T}$ w.r.t. $C \sqsubseteq D$.

Theorem 5. #MINA *is* #P*-complete.*

Proof. The problem is clearly in #P. Given an $\mathcal{HL}$ TBox $\mathcal{T}$, a GCI $C \sqsubseteq D$ that follows from $\mathcal{T}$ and a candidate solution $\mathcal{T}' \subseteq T$, we can in polynomial time verify that $\mathcal{T}'$ is a MinA.

For showing #P-hardness, we are going to give a parsimonious reduction from #HYPERGRAPH TRANSVERSAL, which is the problem of counting the minimal transversals of a given hypergraph $\mathcal{H}$. It has been shown to be #P-complete in [6]. In the reduction, from a given hypergraph $\mathcal{H}$ we construct the same TBox and the GCI constructed in the proof of Theorem 2. Note that this construction establishes a bijection between the minimal transversals of $\mathcal{H}$ and MinAs of $\mathcal{T}$ w.r.t. $\bigsqcap_{v \in V} P_v \sqsubseteq A$. That is, a $W \subseteq V$ is a minimal transversal of $\mathcal{H}$ if and only if the set $\mathcal{T} := \{P_v \sqsubseteq \bigsqcap_{v \in E, E \in \mathcal{E}_\mathcal{H}} P_E \mid v \in W\} \cup \{\bigsqcap_{E \in \mathcal{E}_\mathcal{H}} P_E \sqsubseteq A\}$ defined using W is a MinA. That is, this construction preserves the number of solutions, i.e., it is parsimonious. □

Instead of counting all MinAs, one can also try to count the MinAs that contain a specific axiom. 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 [19] as a heuristic for correcting an error while minimizing the changes in the set of axioms.

Problem: #MINA-RELEVANCE
Input: An $\mathcal{HL}$ TBox $\mathcal{T}$, a GCI $C \sqsubseteq D$ such that $\mathcal{T} \models C \sqsubseteq D$ and an axiom $t \in \mathcal{T}$.
Output: The number of all MinAs of $\mathcal{T}$ w.r.t. $C \sqsubseteq D$ that contain t.

Theorem 6. #MINA-RELEVANCE *is* #P*-complete.*

Proof. The problem is in #P since given an $\mathcal{HL}$ TBox $\mathcal{T}$, a GCI $C \sqsubseteq D$ that follows from $\mathcal{T}$ an axiom $t \in \mathcal{T}$ and a candidate solution $\mathcal{T}' \subseteq T$, we can in polynomial time verify that $\mathcal{T}'$ is a MinA and it contains t.

We show #P-hardness by a parsimonious reduction from #MINA, which has been shown to be #P-complete above. Given an instance of #MINA with the TBox $\mathcal{T}$ and the GCI $A \sqsubseteq B$ we construct the TBox $\mathcal{T}' := \mathcal{T} \cup \mathcal{S}_0$, where $\mathcal{S}_0 = \{A \sqsubseteq C, B \sqcap C \sqsubseteq D\}$, and C and D are two concept names not occurring in $\mathcal{T}$. It is not difficult to see that a set $\mathcal{S} \subseteq \mathcal{T}$ is a MinA for $\mathcal{T}$ w.r.t. $A \sqsubseteq B$ if and only if $\mathcal{S} \cup \mathcal{S}_0$ is a MinA for $\mathcal{T}'$ w.r.t. $A \sqsubseteq D$. Moreover, every MinA for $\mathcal{T}'$ w.r.t. $A \sqsubseteq D$ contains the axioms in $\mathcal{S}_0$. Thus, there are exactly as many MinAs for $\mathcal{T}$ w.r.t. $A \sqsubseteq B$ as there are for $\mathcal{T}'$ w.r.t. $A \sqsubseteq D$ containing the axiom $A \sqsubseteq C$. □

5 Concluding remarks and future work

We have analyzed the complexity of finding all the MinAs for a given $\mathcal{HL}$ TBox, and some other related problems in axiom pinpointing. The bottom line of this analysis is that axiom pinpointing is hard, even in this restricted setting, where we can only express conjunctions and implications. Obviously, the hardness results transfer to any logic more expressive than $\mathcal{HL}$ with GCIs. It is however unclear whether better bounds can be obtained if we restrict our attention to acyclic TBoxes, or to logics like DL-Lite [5] in which conjunction is not allowed on the lefthand side of the axioms.

One problem that remains open is the exact complexity of computing all MinAs. In [2] it was shown that if we allow for an irrefutable portion of the TBox, then there cannot be an algorithm that outputs all MinAs in output polynomial time. Here we show that if all axioms are refutable then this problem is at least as hard as generating the minimal transversals of a given hypergraph. However, whether it is intractable remains open. As future work, we are going to work on determining whether this problem is intractable or it is computationally equivalent to generating minimal transversals.

Another branch for future work refers to efficiently finding approximate solutions for the hard problems presented here. For instance finding an approximation to the total number of MinAs allows us at least to get an idea on how problematic is a given (unwanted) consequence. Alternatively, a set of axioms that explains a given consequence but is not necessarily minimal, can be helpful for a better understanding of the given consequence.

References

1. F. Baader and B. Hollunder. Embedding defaults into terminological representation systems. *Journal of Automated Reasoning*, 14:149–180, 1995.
2. F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In J. Hertzberg, M. Beetz, and R. Englert, editors, *Proceedings of the 30th German Conference on Artificial Intelligence (KI2007)*, volume 4667 of *Lecture Notes in Artificial Intelligence*, pages 52–67. Springer-Verlag, 2007.
3. F. Baader and 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.
4. C. Berge. *Hypergraphs.* Elsevier Science Publishers B.V. (North Holland), 1989.
5. D. Calvanese, G. D. Giacomo, M. Lenzerini, R. Rosati, and G. Vetere. DL-Lite: Practical reasoning for rich DLs. In *Proceedings of the 2004 Description Logic Workshop (DL 2004)*, volume 104 of *CEUR Electronic Workshop Proceedings, http://ceur-ws.org/*, 2004.
6. A. Durand and M. Hermann. On the counting complexity of propositional circumscription. *Information Processing Letters*, 106(4):164–170, 2008.
7. T. Eiter and G. Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42(1):3–42, 1995.

8. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.
9. T. Eiter and G. Gottlob. Hypergraph transversal computation and related problems in logic and AI. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proceedings of the European Conference on Logics in Artificial Intelligence (JELIA 2002)*, volume 2424 of *Lecture Notes in Computer Science*, pages 549–564. Springer-Verlag, 2002.
10. M. L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21(3):618–628, 1996.
11. G. Friedrich, G. Gottlob, and W. Nejdl. Hypothesis classification, abductive diagnosis and therapy. In G. Gottlob and W. Nejdl, editors, *Proceedings of the International Workshop on Expert Systems in Engineering, Principles and Applications*, volume 462 of *Lecture Notes in Computer Science*, pages 69–78, Viena, Austria, 1990. Springer-Verlag.
12. M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Company, New York, NY, USA, 1990.
13. D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.
14. K. Lee, T. Meyer, and J. Z. Pan. Computing maximally satisfiable terminologies for the description logic ALC with GCIs. In B. Parsia, U. Sattler, and D. Toman, editors, *Proceedings of the International Workshop on Description Logics (DL'06)*, Lake District, UK, 2006.
15. B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In A. Ellis and T. Hagino, editors, *Proceedings of the 14th international conference on World Wide Web (WWW 2005)*, pages 633–640. ACM, 2005.
16. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
17. R. Rymon. Search through systematic set enumeration. In B. Nebel, C. Rich, and W. Swartout, editors, *Proceedings of the International Conference on the Principles of Knowledge Representation and Reasoning (KR'92)*, pages 539–550, Cambridge, MA, USA, 1992.
18. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In G. Gottlob and T. Walsh, editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 355–362. Morgan Kaufmann, 2003.
19. S. Schlobach, Z. Huang, R. Cornet, and F. Harmelen. Debugging incoherent terminologies. *Journal of Automated Reasoning*, 39(3):317–349, 2007.
20. B. Suntisrivaraporn, G. Qi, Q. Ji, and P. Haase. A modularization-based approach to finding all justifications for OWL DL entailments. In J. Domingue and C. Anutariya, editors, *Proceedings of the 3rd Asian Semantic Web Conference (ASWC'08)*, volume 5367 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2008.
21. L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.