Pinpointing in the Description Logic $\mathcal{EL}^+$

Lorenz Leutgeb

International Center for Computational Logic, TU Dresden
lorenz.leutgeb@mailbox.tu-dresden.de

Abstract. Description Logic (DL) is an area of active research in the field knowledge representation and artificial intelligence. It is concerned with representing concepts in formal logics whose expressiveness lies between propositional logic and first order logic, and (computationally) reasoning about them. A key reasoning problem in general, and thus also for DLs, is the question of whether one concept is subsumed by another. Additionally, given such a subsumption relationship between concepts A and B, we call the problem of finding the set(s) of axioms that allow to conclude A *subsumes* B, *axiom pinpointing.* We are concerned with the description logic $\mathcal{EL}^+$ in particular, for which subsumption can be solved in polynomial time. An algorithm for axiom pinpointing, based on a subsumption algorithm, is discussed in detail. We briefly summarize complexity results and mention weaker variants that still behave well in practice.

1 Introduction

Description Logic (DL) is a successful subfield of research in Knowledge Representation and Reasoning (KRR). At its core lies a family of logic languages, called Description Logics (DLs). Most of these logics lie in decidable fragments of classical first order logic, meaning that their syntax corresponds to a set of formulae for which first order logic is decidable, e.g. the guarded fragment or the two-variable fragment. Analysis of relations between the expressiveness of a logic and tractability of associated reasoning problems is a big effort within DL. DL (field) DLs (logics)

From an application standpoint, DLs are used to structure conceptual knowledge in problem domains like natural language processing, databases and ontologies. The Web Ontology Language (OWL), fundamental to the semantic web, is formally supported by Description Logic. Also, large biomedical knowledge bases like Snomed CT and Galen are expressed in DLs. Applications

A common reasoning problem is *concept subsumption*.[1] Intuitively, this is the same as asking whether all objects that belong to concept C also belong to concept D, under the axioms of a given knowledge base, i.e. whether D *subsumes* C wrt. the axioms. With the advent of large knowledge bases (hundreds of thousands of axioms), new challenges arise: Assume that under the axioms of Motivation

[1] Other reasoning problems include *satisfiability* and *equivalence* of concepts, as well as *consistency* of knowledge bases. Concept subsumption is the most relevant for this work.

a large knowledge base, D subsumes C, but it turns out that this consequence is not consistent with observations in the real world (e.g. an object that is a D, but not a C was newly discovered). Now, the knowledge base should be adjusted to better model reality. But which of the thousands of axioms give rise to the faulty conclusion? We call this problem *axiom pinpointing.*

Naturally, we are interested in **computing minimal axiom sets (MinA)** that explain why D subsumes C.

Contents

In this work we summarize results on axiom pinpointing for the description logic $\mathcal{EL}^+$. First, in Section 2, we revisit syntax and semantics of the logics under consideration. Section 3 is meant to showcase different approaches to axiom pinpointing. We follow the distinction made in [9, Section 1]: A *glass-box* algorithm based on a polynomial-time algorithm for subsumption in $\mathcal{EL}^+$ is discussed in 3.1, while in 3.2 we address two *black-box* algorithms. We defer all complexity results to Section 4 where we address the complexity of finding a single (minimal) set of axioms (Section 4.2) separately from the complexity of finding all (minimal) sets of axioms (Section 4.1). Modifications that improve run time for computing one MinAs towards application in practice are mentioned in Section 5. We conclude in Section 6.

Note that our investigation starts with the work by Baader, Peñaloza and Suntisrivaraporn [2,3]. We take into account subsequent results achieved by Peñaloza and Sertkaya [8,9].

2 Preliminaries

In this section we introduce syntax and semantics of $\mathcal{EL}^+$ as well as the core reasoning problem of concept subsumption.

Syntax

Description Logic considers syntax called *concept descriptions*, inductively defined via *constructors* from a set of concept names and a set of role names. The constructors of the respective languages relevant to this work are listed in the upper section of the column titled *Syntax* in Table 1.

Semantics

Semantics are defined in terms of an interpretation I which talks about a non-empty set of individuals denoted Δ^I by means of an interpretation function $\cdot^I$ as follows: $\cdot^I$ maps each concept name $A \in \mathsf{N_C}$ to a set of individuals $A^I \subseteq \Delta^I$, and each role name $r \in \mathsf{N_R}$ to a binary relation $r^I \subseteq \Delta^I \times \Delta^I$. Concept descriptions are interpreted by extending $\cdot^I$ as defined in the upper section of the column titled *Semantics* in Table 1.

TBox

In our setting, knowledge bases take the form of finite sets of axioms. We call such a set *TBox*, denoted by $\mathcal{T}$. Axioms are relations between concept descriptions, and their form is shown in the lower section of Table 1. We say that an axiom is *satisfied* under I if the corresponding condition (column titled *Semantics*) holds. Further, we say that an interpretation I *satisfies* $\mathcal{T}$ if all axioms in $\mathcal{T}$ are satisfied by I.

Since all $\mathcal{EL}^+$ TBoxes are consistent by definition (there are no constructors that may cause logical inconsistencies [2,3, Sec. 2], in particular there is no form of negation available), satisfiability of concepts and the consistency of TBoxes

Table 1. Overview of syntax and semantics of selected logics. Here, $A, B \in \mathsf{N_C}$ are concept names, C, D are concept descriptions (inductively), $r_1, \dots, r_n, s \in \mathsf{N_R}$ are role names. We denote an interpretation by $I = (\Delta^I, \cdot^I)$. GCI stands for General Concept Inclusion.

Name	Syntax	Semantics	$\mathcal{AL}$	$\mathcal{EL}$	$\mathcal{EL}^+$	$\mathcal{HL}$
Top	$\top$	Δ^I	✓	✓	✓	✓
Bottom	$\perp$	$\emptyset$	✓			
Atomic Negation	$\neg A$	$\Delta^I \setminus A^I$	✓			
Conjunction	$C \sqcap D$	$C^I \cap D^I$	✓	✓	✓	✓
Value Restriction	$\forall r.C$	$\{x \in \Delta^I \mid \forall y \in \Delta^I : (x, y) \in r^I \rightarrow y \in C^I\}$	✓			
Existential Restr.	$\exists r.C$	$\{x \in \Delta^I \mid \exists y \in \Delta^I : (x, y) \in r^I \wedge y \in C^I\}$		✓	✓	
GCI	$C \sqsubseteq D$	$C^I \subseteq D^I$	✓	✓	✓	✓
Concept Definition	$C \equiv D$	$C^I = D^I$	✓	✓	✓	✓
Role Inclusion	$r_1 \circ \dots \circ r_n \sqsubseteq s$	$r_1^I \circ \dots \circ r_n^I \subseteq s^I$			✓	

are trivial. Instead, we are generally interested in deciding the reasoning problem of *concept subsumption.*

Definition 1 (Concept Subsumption, cf. [2, Def. 1]). *Given two $\mathcal{EL}^+$ concept descriptions C, D and an $\mathcal{EL}^+$ TBox $\mathcal{T}$, C is subsumed by D wrt. $\mathcal{T}$ (written $C \sqsubseteq_\mathcal{T} D$) if for every interpretation I that satisfies $\mathcal{T}$ we have that $C^I \subseteq D^I$.* $C \sqsubseteq_\mathcal{T} D$

With this, we proceed to define minimal axiom sets which are crucial to this work.

Definition 2 (MinA, cf. [2, Def. 2 without partitioning $\mathcal{T}$]). *Let $\mathcal{T}$ be an $\mathcal{EL}^+$ TBox and A, B concept names occurring in it such that $A \sqsubseteq_\mathcal{T} B$. Then a* minimal axiom set (MinA) *for $\mathcal{T}$ wr.t. $A \sqsubseteq B$ is a subset $\mathcal{S} \subseteq \mathcal{T}$ such that $A \sqsubseteq_\mathcal{S} B$ but $A \not\sqsubseteq_{\mathcal{S}'} B$ for all strict subsets $\mathcal{S}' \subset S$.* MinA

Example 1 (Humans and Animals (cf. [2,3, Sec. 3])). Consider the following TBox $\mathcal{T}$ containing axioms that model knowledge about species:

$\mathsf{Human} \sqsubseteq \exists \mathsf{parent}.\mathsf{Human}$ — Humans have a human parent. (a_1)

$\mathsf{Human} \sqsubseteq \mathsf{Monkey}$ — Humans are monkeys. (a_2)

$\exists \mathsf{parent}.\mathsf{Monkey} \sqsubseteq \mathsf{Animal}$ — Monkey parent? It's an animal. (a_3)

$\mathsf{Monkey} \sqsubseteq \mathsf{Animal}$ — Monkeys are animals. (a_4)

$\mathsf{Fish} \sqsubseteq \mathsf{Animal}$ — Fish are animals. (a_5)

Observe that $\mathsf{Human} \sqsubseteq_\mathcal{T} \mathsf{Animal}$ holds. Now assume evidence has been found that not all humans are animals, thus we should adjust our TBox such that this relationship will not hold anymore. What are the axioms that give rise to the subsumption? We can see that two minimal sets of axioms explain it: $\mathcal{S}_1 = \{a_2, a_4\}$ and $\mathcal{S}_2 = \{a_1, a_2, a_3\}$. Even though $|\mathcal{S}_1| < |\mathcal{S}_2|$ both are MinAs, since we cannot find strict subsets that explain why all humans are animals.

Also note that the consequence holds when removing either a_1 and a_3 or a_4 alone and that a_2 is contained in both sets. The axiom a_5 is not relevant in our scenario, and we may imagine that there are thousands of other irrelevant axioms in $\mathcal{T}$.

We hypothesize that a_4 is indeed a sensible axiom and a_1, a_3 are reasonable, but a_2 causes trouble since it states that humans are monkeys. If we remove a_2, then the resulting TBox will conform with our observation.

3 Algorithms

In this section we want to show how one MinA and multiple MinAs can be computed. We distinguish between *glass-box* algorithms and *black-box* algorithms. The former captures algorithms that may take advantage of the internals of the reasoning algorithm by inspecting axioms and their structure, generally using more involved data structures. On the contrary, the latter means algorithms that are built on top of basic reasoning services, thus are more abstract but can leverage already existing implementations of these reasoning services.

We will detail one glass-box approach and a black-box approach with some variations.

3.1 Glass-Box Approach

The glass box algorithm we present is designed as an extension of a polynomial-time algorithm for subsumption [5] in $\mathcal{EL}^+$ (cf. Def. 1). Therefore we first consider the algorithm for subsumption (Algorithm 1), and only then extend it to compute MinAs (Algorithm 2).

Normalization

Following two simplifications based on normalization greatly reduce the number of cases that need to be covered in the algorithm [2, Sec. 2]: (1) It is sufficient to address subsumption for concept names, given the following reduction: $C \sqsubseteq_\mathcal{T} D$ iff $A \sqsubseteq_{\mathcal{T} \cup \{A \sqsubseteq C, D \sqsubseteq B\}} B$ where A, B are new concept names not occurring in $C, D, \mathcal{T}$ (for a proof concerning $\mathcal{EL}$ cf. [1, Lemma 6.1,p. 142]). (2) It is possible to translate a TBox $\mathcal{T}$ into a *normalized TBox* (meaning that all GCIs are of the forms as presented in Table 2, cf. [1, Fig. 6.1, p. 143]) in time $\mathcal{O}(|\mathcal{T}|)$ (proven in [1, Lemma 6.2, p. 143]).

Note that lines 3 and 5 in Algorithm 1 refer to Table 2, and that the choice of r (and therefore the rule to apply) in those lines is don't care nondeterministic.

Now we extend Algorithm 1 to compute all MinAs in two steps: First, we modify it so that a monotone Boolean *pinpointing formula* that corresponds to the subsumption relation in question is generated via labeling of assertions. Second, we obtain these MinAs by constructing models for the pinpointing formula, such that the interpretation of propositional variables corresponds to a selection of axioms in the MinAs. Together this means that we can efficiently encode MinAs by means of Boolean formulae.

Algorithm 1: SUBSUMPTION($\mathcal{T}$, A, B)

Input: An $\mathcal{EL}^+$ TBox $\mathcal{T}$ in normal form over $\mathsf{N_C}$ and $\mathsf{N_R}$, and $A, B \in \mathsf{N_C}$.
Output: "yes" if $A \sqsubseteq_{\mathcal{T}} B$ holds, "no" otherwise.

$\mathcal{A} := \{(A, A) \mid A \in \mathsf{N_C}\}$ `// Account for` $A \sqsubseteq_{\mathcal{T}} A$
  $\cup\ \{(A, \top) \mid A \in \mathsf{N_C}\}$ `// Account for` $A \sqsubseteq_{\mathcal{T}} \top$
while *there is an i s.t.* $1 \leq r \leq 5$*,* $a_r \in \mathcal{T}$ *and* $P_r \subseteq \mathcal{A}$ `// see Table 2`
 do
  $\mathcal{A} := \mathcal{A} \cup \{q_r\}$ `// Axiom` a_r `explains` q_r
end
if $(A, B) \in \mathcal{A}$ **then**
  return "yes"
end
return "no"

Table 2. Completion rules for subsumption in $\mathcal{EL}^+$, adapted from [2,3, Fig. 1], see also [11, Fig. 5.2, p. 104]. Rule r is applicable if $a_r \in \mathcal{T}$ and $P_r \subseteq \mathcal{A}$. If rule r is applied, then q_r is added to $\mathcal{A}$.

Rule	**Applicability**		**Result**
r	a_r (GCI)	P_r (set of assertions)	q_r (assertion)
1	$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$	$(X, A_1), \ldots, (X, A_n)$	(X, B)
2	$A \sqsubseteq \exists r.B$	(X, A)	(X, r, B)
3	$\exists r.A \sqsubseteq B$	$(X, r, Y), (Y, A)$	(X, B)
4	$r \sqsubseteq s$	(X, r, Y)	(X, s, Y)
5	$r \circ r' \sqsubseteq s$	$(X, r, Y), (Y, r', Z)$	(X, s, Z)

Labeling

By means of a labeling function $lab : \mathcal{T} \longrightarrow \mathcal{R}$ (where $\mathcal{R}$ is a set of propositional variables) every axiom $a \in \mathcal{T}$ is labeled with a unique propositional variable $lab(a) \in \mathcal{R}$, and we denote by $lab(\mathcal{T}) \subseteq \mathcal{R}$ the set of labels of all axioms in $\mathcal{T}$. A *monotone Boolean formula over* $lab(\mathcal{T})$ is a Boolean formula using (some of) the propositional variables $lab(\mathcal{T})$ and only the binary connectives conjunction and disjunction, as well as the nullary connective `true` for truth.

Valuation

A *valuation* is characterized by the set of propositional variables that it assigns to be true. For a valuation $\mathcal{V} \subseteq lab(\mathcal{T})$ let $\mathcal{T}_\mathcal{V} = \{a \in \mathcal{T} \mid lab(a) \in \mathcal{V}\}$ be the selection of axioms for which their corresponding label evaluates to true under $\mathcal{V}$.

Pinpointing Formula

Definition 3 (Pinpointing Formula [2, Def. 3]). *Given an $\mathcal{EL}^+$ TBox $\mathcal{T}$ and concept names A, B occuring in it, a monotone Boolean formula ψ over $lab(\mathcal{T})$ is a* pinpointing formula *for $\mathcal{T}$ wrt. $A \sqsubseteq B$ if for every valuation $\mathcal{V} \subseteq lab(\mathcal{T})$ it holds that $A \sqsubseteq_{\mathcal{T}_\mathcal{V}} B$ iff $\mathcal{V}$ satisfies ψ.*

For the two MinAs in Example 1, $\mathcal{S}_1 = \{a_2, a_4\}$ and $\mathcal{S}_2 = \{a_1, a_2, a_3\}$, we obtain $\phi = a_2 \wedge (a_4 \vee (a_1 \wedge a_3))$ as a pinpointing formula. An obvious connection can be pointed out by translating ϕ into disjunctive normal form (DNF) which yields $(a_2 \wedge a_4) \vee (a_1 \wedge a_2 \wedge a_3)$. Lastly, to establish a correspondence, we also need to address $\subseteq$-minimality of valuations.

Proposition 1 ([2, Prop. 1]). *Let ψ be a pinpointing formula for $\mathcal{T}$ wrt. $A \sqsubseteq B$. Then*

$$\{\mathcal{T}_\mathcal{V} \mid \mathcal{V} \text{ is a } \subseteq\text{-minimal valuation satisfying } \psi\}$$

is the set of all MinAs for $\mathcal{T}$ wrt. $A \sqsubseteq B$.

The initial assertions (added in lines 1 and 2) of the form (A, A) and $(A, \top)$ where $A \in \mathsf{N_C}$ are labeled with `true` in line 3.

Note that Algorithm 2 terminates in time exponential in $|\mathcal{T}|$, and it allows us to find all MinAs not only for a specific subsumption relation $A \sqsubseteq_\mathcal{T} B$, but we may query the returned set of assertions for (A, B) and consider $lab((A, B))$ for any subsumption relation of interest.

More formally (cf. [2,3, Thm. 2]), the resulting set of assertions $\mathcal{A}$, after termination of the loop in lines 4-15 satisfies the following properties: (1) $A \sqsubseteq_\mathcal{T} B$ if and only if $(A, B) \in \mathcal{A}$, and (2) $lab((A, B))$ is a pinpointing formula for $\mathcal{T}$ wrt. $A \sqsubseteq B$.

Example 2. Reconsider the TBox $\mathcal{T}$ from Example 1. We trace the execution of the loop in AllMinAs($\mathcal{T}$):

1. $r = 1$: Since we have a_2 asserting that $\mathsf{Human} \sqsubseteq \mathsf{Monkey}$ and $(\mathsf{Human}, \mathsf{Human}) \in \mathcal{A}$ labeled `true`, the assertion $(\mathsf{Human}, \mathsf{Monkey})$ with label $a_2 \wedge \mathtt{true} \equiv a_2$ is added.
2. $r = 2$: We may add $(\mathsf{Human}, \mathsf{parent}, \mathsf{Human})$ with label $a_1 \wedge \mathtt{true} \equiv a_1$.
3. $r = 1$: Given a_4 and $(\mathsf{Human}, \mathsf{Monkey}) \in \mathcal{A}$ (from iteration 1) we add $(\mathsf{Human}, \mathsf{Monkey})$ to $\mathcal{A}$ with label $a_2 \wedge a_4$.

Algorithm 2: $\textsc{AllMinAs}(\mathcal{T})$

Input: An $\mathcal{EL}^+$ TBox $\mathcal{T}$ in normal form over $\mathsf{N_C}$.
Output: A set of assertions and a labeling function.

$\mathcal{A} := \{(A, A) \mid A \in \mathsf{N_C}\}$ `// Account for` $A \sqsubseteq_{\mathcal{T}} A$.
  $\cup \{(A, \top) \mid A \in \mathsf{N_C}\}$ `// Account for` $A \sqsubseteq_{\mathcal{T}} \top$.
$lab(a) :=$ `true` for all $a \in \mathcal{A}$ `// Initialize labels.`
while *there is an* r *s.t.* $1 \le r \le 5$, $a_r \in \mathcal{T}$ *and* $P_r \subseteq \mathcal{A}$ `// see Table 2`
do
  $\phi = lab(a_r) \wedge \bigwedge_{p \in P_r} lab(p)$
  if $q_r \notin \mathcal{A}$ **then**
    `//` a_i `is the only known explanation of` a_r.
    $\mathcal{A} := \mathcal{A} \cup \{q_i\}$
    $lab(q_r) := \phi$
  else
    `//` q_r `can already be explained,` a_r `might be a new explanation!`
    $\psi = lab(q_r)$
    if $\psi \vee \phi \not\equiv \psi$ **then**
      `// Override, as` q_i `might be explained by` ψ `or` ϕ.
      $lab(q_r) := \psi \vee \phi$
    end
  end
end
return $(\mathcal{A}, lab)$

4. $r = 3$: Finally, for a_3 and $(\mathsf{Human}, \mathsf{Monkey}), (\mathsf{Human}, \mathsf{parent}, \mathsf{Human}) \in \mathcal{A}$, the label for $(\mathsf{Human}, \mathsf{Animal})$ is overridden to $(a_2 \wedge a_4) \vee (a_1 \wedge a_2 \wedge a_3)$.

Since no more assertions can be added after four iterations, the algorithm terminates. The label for $(\mathsf{Human}, \mathsf{Animal})$, and therefore the computed pinpointing formula corresponding to $\mathsf{Human} \sqsubseteq \mathsf{Animal}$, is $(a_2 \wedge a_4) \vee (a_1 \wedge a_2 \wedge a_3)$ as expected.

Denormalization

Lastly , we need to address normalization assumptions: In case the input TBox $\mathcal{T}$ was obtained from another TBox $\widehat{\mathcal{T}}$ by normalization, the resulting pinpointing formula may be translated to correspond to $\widehat{\mathcal{T}}$. To do this, we assume a unique labeling of axioms in $\widehat{T}$ and note that any axiom in $\mathcal{T}$ has some corresponding *sources* in $\widehat{\mathcal{T}}$, since $\mathcal{T}$ is generated by splitting axioms from $\widehat{\mathcal{T}}$ until they all are in normal form. Now to translate a pinpointing formula ϕ for $\mathcal{T}$ into a pinpointing formula for $\widehat{\mathcal{T}}$, replace each axiom label in ϕ with a disjunction of all the labels of the source axioms of the corresponding axiom (see [2,3, Sec. 3, Sec. 5] and for a formal argument [11, Sec. 5.2, p. 107]).

3.2 Black-Box Approaches

In this section we want to present a naïve algorithm and a slightly more involved variant, to compute *one* minimal set of axioms for a given subsumption consequence, in a *black-box* manner: Both Algorithm 3 and 4, discussed below, only rely on deciding subsumption and counting/choosing elements of the input TBox. In contrast to Algorithm 1 and its extension Algorithm 2 from the previous section, the structure of individual axioms is not considered. Consequently, their high-level approach is exposed in a more direct fashion as we omit statements that deal with peculiarities, such as the structure of an axiom.

Algorithm 3 linearily scans a working copy of the given knowledge base: For every axiom, it is tested whether the subsumption relation under question still holds without it. If this is the case, then it is removed from the working copy. Clearly, $\mathcal{O}(|\mathcal{T}|)$ subsumption tests must be performed. The result is a minimal set of axioms.

Algorithm 3: LinOneMinA($\mathcal{T}$, A, B)
(cf. [2, Alg. 1], [4, Alg. 1], [11, Alg. 7, p. 97])

Input: An $\mathcal{EL}^+$ TBox $\mathcal{T} = \{a_1, \ldots, a_n\}$ over $\mathsf{N_C}$ and $\mathsf{N_R}$, and $A, B \in \mathsf{N_C}$.
Output: If $A \sqsubseteq_\mathcal{T} B$ holds, one MinA for $\mathcal{T}$ wrt. $A \sqsubseteq B$, else $\emptyset$.

if $A \not\sqsubseteq_\mathcal{T} B$ **then**
  return $\emptyset$
end
$\mathcal{S} := \mathcal{T}$ `// Initialize MinA to set of all axioms.`
for $1 \leq i \leq n$ **do**
  if $A \sqsubseteq_{\mathcal{S}\setminus\{a_i\}} B$ **then**
    $\mathcal{S} := \mathcal{S} \setminus \{a_i\}$ `// Remove irrelevant axiom.`
  end
end
return $\mathcal{S}$

Algorithm 4 employs a divide-and-conquer approach reminiscent of binary search. The minimal axiom set $\mathcal{S}$ is constructed recursively. For every recursive call, the input TBox is split into two halves (lines 4 and 5) and in case any of the two halves preserves $A \sqsubseteq B$, search is continued in this part of the input. Otherwise, $\mathcal{S}$ is constructed as the union of the result for both halves respectively. With this approach, only $\mathcal{O}((|\mathcal{S}|-1)+|\mathcal{S}|\cdot\log(|\mathcal{T}|/|\mathcal{S}|))$ subsumption tests need to be performed [11, p. 89].

4 Complexity

In this section we briefly address the computational complexity of computing all MinAs and one MinA respectively.

4.1 All MinAs

Since there may be exponentially many MinAs for a given $\mathcal{EL}^+$ TBox, it is clear that an exponential run-time in the worst-case cannot be avoided.

Algorithm 4: LogOneMinA($\mathcal{T}$, A, B, $\mathcal{S}$)
(cf. [4, Alg. 2], [11, Alg. 8, p. 98])

Input: An $\mathcal{EL}^+$ TBox $\mathcal{T} = \{a_1, \ldots, a_n\}$ over $\mathsf{N_C}$ and $\mathsf{N_R}$, and $A, B \in \mathsf{N_C}$ s.t. $A \sqsubseteq_\mathcal{T} B$ holds, as well as a "support set" (of axioms) for $A \sqsubseteq_\mathcal{T} B$.
Output: minimal subset $\mathcal{S}' \subseteq \mathcal{T}$ s.t. $A \sqsubseteq_{\mathcal{S} \cup \mathcal{S}'} B$ holds.

if $|\mathcal{T}| = 1$ **then**
  return $\mathcal{T}$
end
$L := \{a_i \in \mathcal{T} \mid 1 \leq i \leq \lfloor n/2 \rfloor\}$
$R := \{a_i \in \mathcal{T} \mid \lfloor n/2 \rfloor + 1 \leq i \leq n\}$
if $A \sqsubseteq_{\mathcal{S} \cup L} B$ **then**
  return LogOneMinA*(L, A, B, $\mathcal{S}$)*
end
if $A \sqsubseteq_{\mathcal{S} \cup R} B$ **then**
  return LogOneMinA*(R, A, B, $\mathcal{S}$)*
end
$\mathcal{S}_1$:= LogOneMinA(L, A, B, $\mathcal{S} \cup R$)
$\mathcal{S}_2$:= LogOneMinA(R, A, B, $\mathcal{S} \cup \mathcal{S}_1$)
return $\mathcal{S}_1 \cup \mathcal{S}_2$

Example 3 ([2,3, Ex. 1]). Consider the language $\mathcal{HL}$, which is a fragment of $\mathcal{EL}^+$ (cf. Table 1). Given $n \geq 1$, we construct a $\mathcal{HL}$ TBox of size linear in n, that gives rise to 2^n MinAs:

$$\mathcal{T}_n := \left\{ \quad B_{i-1} \sqsubseteq P_i \sqcap Q_i, \quad P_i \sqsubseteq B_i, \quad Q_i \sqsubseteq B_i \quad \middle| \quad 1 \geq i \geq n \quad \right\}$$

Note that there are 2^n MinAs for $\mathcal{T}_n$ wrt. $B_0 \sqsubseteq B_n$, since for each i with $1 \leq i \leq n$ either the axiom $P_i \sqsubseteq B_i$ or the axiom $Q_i \sqsubseteq B_i$ might be in a MinA.

Further, it can be shown that computing any interesting property of the set of all MinAs is not possible in polynomial time (unless P = NP).

Theorem 1 ([2,3, Thm. 3 without partitioning $\mathcal{T}$]). *Given an $\mathcal{HL}$ TBox $\mathcal{T}$, concept names A, B occurring in $\mathcal{T}$, and a natural number n, it is NP-complete to decide whether or not there is a MinA $\mathcal{S}$ for $\mathcal{T}$ wrt. $A \sqsubseteq B$ such that $|\mathcal{S}| \leq n$.*

With this, [2,3] starts further investigation. In particular, since one might regard the problem of computing all MinAs as an enumeration problem, this raises the question whether there is an *output polynomial* algorithm for it. An algorithm with this property would only take polynomial time in cases where the number of MinAs is polynomial in the size of the input. However, after showing that there is no output polynomial algorithm for computing all minimal satisfying valuations of monotone Boolean formulae (unless P = NP) [2,3, Thm. 4], a negative result is presented:

Theorem 2 ([2,3, Thm. 5]). *There is no output polynomial algorithm that computes for a given $\mathcal{HL}$ TBox $\mathcal{T} = (\mathcal{T}_s \uplus \mathcal{T}_r)$ and concept names A, B occurring in $\mathcal{T}$, all MinAs for $\mathcal{T}$ wrt. $A \sqsubseteq B$, unless $P = NP$.*

Note that the setting considered in [2,3] and in the above theorem the TBox $\mathcal{T}$ is split into "static" part of $\mathcal{T}$, denoted $\mathcal{T}_s$ and $\mathcal{T}_r$, the "refutable" part respectively. These parts are disjoint as $\uplus$ stands for the *disjoint union* of two sets.

The question whether Theorem 2 also holds for the case where $\mathcal{T}_s = \emptyset$, which corresponds to the view in this work, is addressed in [8, Thm. 2]. There, the problem of computing all MinAs is shown to be "at least as hard as recognizing the set of all minimal transversals of a given hypergraph [6], which is a prominent open problem in hypergraph theory" (quoting from [8], importance of the problem surveyed in [7]).

A comprehensive analysis of the complexity of computing all MinAs in different variants can be found in [9], where similar logics are considered.

4.2 One MinA

Algorithm 3 shows how one MinA can be computed in polynomial time. Since the loop in lines 5-8 is executed exactly $n = |\mathcal{T}|$ times, and the subsumption test also runs in time $\mathcal{O}(P(|\mathcal{T}|))$, an overall time complexity polynomial in $|\mathcal{T}|$ is achieved. The result is highlighted in [2,3, Thm. 6].

5 Implementations and Application in Practice

As stated in the previous section, computing one MinA takes time polynomial in the number of axioms in the TBox, i.e. in the size of the knowledge base. While this might sound promising for practical applications and implementation, experimental reports show a different picture: In [11, Sec. 5.2, p. 97], Suntisrivaraporn states that "The naïve minimization algorithm [Algorithm 3] did not terminate on Snomed CT in 48 hours".

To make use of pinpointing in practice, [2,3, Sec. 5] proposes a trade-off. By relaxing the minimality property of computed axiom sets, significant gains in runtime can be realized. To decrease the size of the input for a black-box algorithm, a greedy algorithm based on Algorithm 2, where the branch in lines 10-14 is dropped, makes a pre-selection. Note that minimality breaks, since only the first explanation that the algorithm considers will be reflected by the label. The remaining set is minimized using Algorithm 3.

Example 4. To highlight how minimality is given up when using the modified, greedy version of Algorithm 2, consider the following TBox $\mathcal{T}$:

$$
\begin{array}{llll}
A \sqsubseteq X & (a_1) & X \sqsubseteq Y & (a_2) \\
Y \sqsubseteq X & (a_3) & X \sqsubseteq B & (a_4) \\
Z \sqsubseteq X & (a_5) & &
\end{array}
$$

Note that we might explain $A \sqsubseteq B$ wrt. $\mathcal{T}$ by two sets of axioms $\mathcal{S}_1 = \{a_1, a_4\}$ or $\mathcal{S}_2 = \{a_1, a_2, a_3, a_4\}$. However, since $\mathcal{S}_2 \supset \mathcal{S}_1$, we see that $\mathcal{S}_2$ is not a MinAs, but $\mathcal{S}_1$ is.

Running Algorithm 2 normally would yield the pinpointing formula $(a_1 \wedge a_4) \vee (a_1, a_2, a_3, a_4)$, and by the minimality condition in Proposition **??** we arrive at $\mathcal{S}_1$. However, with the greedy modification, it is possible that we obtain either of two different pinpointing formulas, $\phi_1 = a_1 \wedge a_4$ and $\phi_2 = a_1 \wedge a_2 \wedge a_3 \wedge a_4$. This is because of the fact that rule application is don't care nondeterministic. So, in case the algorithm labels (A, B) by ϕ_2, we arrive at the non-minimal set of explanations $\mathcal{S}_2$. Through further minimization by means of Algorithm 3 the final result $\mathcal{S}_1$ is obtained.

A denormalization translation, similar to the one described in Section 3.1 might be needed, which introduces disjunction to the pinpointing formula [2,3, Sec. 5]. In order to keep polynomial runtime, a greedy strategy to find valuations might be appropriate.

With this approach, run times for the Galen [10] ontology decreased from 7 hours to approximately 10 minutes with an increase in size of 2.59% [2,3, Sec. 5].

Further empirical results are given in [11, Sec. 6.2.5, p. 127].

6 Summary

We have presented a formalization of axiom pinpointing based on [2,3] and covered a glass-box as well as two black-box algorithms to compute MinAs. While implementations for computing one MinAs are practicable, the complexity of enumerating all MinAs is not fully resolved.

References

1. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017), `http://www.cambridge.org/de/academic/subjects/computer-science/knowledge-management-databases-and-data-mining/introduction-description-logic?format=PB#17zVGeWD2TZUeu6s.97`
2. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic $\mathcal{EL}^+$. In: Calvanese, D., Franconi, E., Haarslev, V., Lembo, D., Motik, B., Turhan, A., Tessaris, S. (eds.) Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007. CEUR Workshop Proceedings, vol. 250. CEUR-WS.org (2007), `http://ceur-ws.org/Vol-250/paper_16.pdf`
3. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic $\mathcal{EL}^+$. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, KI 2007, Osnabrück, Germany, September 10-13, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4667, pp. 52–67. Springer (2007), `https://doi.org/10.1007/978-3-540-74565-5_7`

4. Baader, F., Suntisrivaraporn, B.: Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$. In: Cornet, R., Spackman, K.A. (eds.) Proceedings of the Third International Conference on Knowledge Representation in Medicine, Phoenix, Arizona, USA, May 31st - June 2nd, 2008. CEUR Workshop Proceedings, vol. 410. CEUR-WS.org (2008), http://ceur-ws.org/Vol-410/Paper01.pdf
5. Brandt, S.: Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and - what else? In: de Mántaras, R.L., Saitta, L. (eds.) Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004. pp. 298–302. IOS Press (2004)
6. Eiter, T., Gottlob, G.: Identifying the minimal transversals of a hypergraph and related problems. SIAM J. Comput. 24(6), 1278–1304 (1995), https://doi.org/10.1137/S0097539793250299
7. Eiter, T., Gottlob, G.: Hypergraph transversal computation and related problems in logic and AI. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) Logics in Artificial Intelligence, European Conference, JELIA 2002, Cosenza, Italy, September, 23-26, Proceedings. Lecture Notes in Computer Science, vol. 2424, pp. 549–564. Springer (2002), https://doi.org/10.1007/3-540-45757-7_53
8. Peñaloza, R., Sertkaya, B.: Axiom pinpointing is hard. In: Grau, B.C., Horrocks, I., Motik, B., Sattler, U. (eds.) Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009. CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009), http://ceur-ws.org/Vol-477/paper_39.pdf
9. Peñaloza, R., Sertkaya, B.: On the complexity of axiom pinpointing in the $\mathcal{EL}$ family of description logics. In: Lin, F., Sattler, U., Truszczynski, M. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010. AAAI Press (2010), http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1345
10. Rector, A.L., Horrocks, I.R.: Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In: In Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97. AAAI Press (1997)
11. Suntisrivaraporn, B.: Polynomial time reasoning support for design and maintenance of large-scale biomedical ontologies. Ph.D. thesis, Dresden University of Technology, Germany (2009), http://hsss.slub-dresden.de/deds-access/hsss.urlmapping.MappingServlet?id=1233830966436-5928