Knowledge Graphs, Description Logics and Reasoning on Data : Reading

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

Théo Delemazure

Ecole normale supérieure, PSL University

April 9, 2020

The pinpointing problem

Why a certain consequence holds ?

SNOMED CT

$\texttt{Amputation-of-finger} \sqsubseteq \texttt{Amputation-of-arm}$

The pinpointing problem

Goal of pinpointing : Find the minimal subset of a Knowledge base that have a given consequence.

Some papers explain an algorithm for pinpointing in $\mathcal{ALC}$ using an adaptation of the Tableau algorithm.

$\Rightarrow$ What about the inexpressive description logic $\mathcal{EL}$?

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

Authors — *Franz Baader Rafael Penaloza, and Boontawee Suntisrivaraporn*

Year — *2007*

Workshop — *Description Logics*

Citations — 172 to this day

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

Franz Baader[1], Rafael Peñaloza[2*], and Boontawee Suntisrivaraporn[1]

[1] Theoretical Computer Science, TU Dresden, Germany
{baader,meng}@tcs.inf.tu-dresden.de
[2] Intelligent Systems, University of Leipzig, Germany
penaloza@informatik.uni-leipzig.de

1. Introduction

For a developer or user of a DL-based ontology, it is often quite hard to understand why a certain consequence holds, and even harder to decide how to change the ontology in case the consequence is unwanted. For example, in the current version of the medical ontology SNOMED [16], the concept *Amputation-of-Finger* is classified as a subconcept of *Amputation-of-Arm*. Finding the axioms that are responsible for this among the more than 350,000 terminological axioms of SNOMED without support by an automated reasoning tool is not easy.

As a first step towards providing such support, Schlobach and Cornet [14] describe an algorithm for computing all the *minimal subsets* of a given knowledge base that *have* a given *consequence*. In the following, we call such a set a *minimal axiom set (MinA)*. It helps the user to comprehend why a certain consequence holds. The knowledge bases considered in [14] are so-called unfoldable $\mathcal{ALC}$-terminologies, and the unwanted consequences are the unsatisfiability of concepts. The algorithm is an extension of the known tableau-based satisfiability algorithm for $\mathcal{ALC}$ [15], where labels keep track of which axioms are responsible for an assertion to be generated during the run of the algorithm. The authors also coin the name "axiom pinpointing" for the task of computing these minimal subsets.

The problem of computing MinAs of a DL knowledge base was actually considered earlier in the context of extending DLs by default rules. In [2], Baader and Hollunder solve this problem by introducing a labeled extension of the tableau-based consistency algorithm for $\mathcal{ALC}$-ABoxes [9], which is very similar to the one described later in [14]. The main difference is that the algorithm described in [2] does not directly compute minimal subsets that have a consequence, but rather a monotone Boolean formula whose variables correspond to the axioms of the knowledge bases and whose minimal satisfying valuations correspond to the MinAs.

The approach of Schlobach and Cornet [14] was extended by Parsia et al. [12] to more expressive DLs, and the one of Baader and Hollunder [2] was extended by Meyer et al. [11] to the case of $\mathcal{ALC}$-terminologies with general concept inclusions (GCIs), which are no longer unfoldable. Axiom pinpointing has also been considered in other research areas, though usually not under this name.

* Funded by the German Research Foundation (DFG) under grant GRK 446.

Summary

Summary

Reminder : $\mathcal{EL}$

$C_1 \sqcap C_2$ $\exists r.C$ $\top$

Reminder : $\mathcal{EL}$ normal form

$$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$$

$$A \sqsubseteq \exists r.B$$

$$\exists r.A \sqsubseteq B$$

$\Rightarrow$ The normalization can be done in linear time.

Example

The following TBox is in normal form :

$$\{ax_1 : A \sqsubseteq \exists r.A,$$
$$ax_2 : A \sqsubseteq Y,$$
$$ax_3 : \exists r.Y \sqsubseteq B,$$
$$ax_4 : Y \sqsubseteq B\}$$

Reminder : Subsumption in $\mathcal{EL}$

$\mathcal{A}$ is a set of assertions :

$$(A, B) \in \mathcal{A} \Leftrightarrow A \sqsubseteq_{\mathcal{T}} B$$
$$(A, r, B) \in \mathcal{A} \Leftrightarrow A \sqsubseteq_{\mathcal{T}} \exists r.B$$

Initialization : $\forall A$,

$$(A, \top) \in \mathcal{A}$$
$$(A, A) \in \mathcal{A}$$

Reminder : Subsumption in $\mathcal{EL}$

The rules :

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$	**and** $\forall i, (X, A_i) \in \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$	**and** $(X, A) \in \mathcal{A}$	**then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$	**and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$

Stop when no new assertion can be added into $\mathcal{A}$.

Example

$$\{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$$

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$	**and** $\forall i, (X, A_i) \in \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$	**and** $(X, A) \in \mathcal{A}$	**then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$	**and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$

Init : $\mathcal{A} = \{(A, \top), (B, \top), (Y, \top), (A, A), (B, B), (Y, Y)\}$

Example

$$\{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$$

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$	**and** $\forall i, (X, A_i) \in \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$	**and** $(X, A) \in \mathcal{A}$	**then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$	**and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$

$$\mathcal{A} = \{(A, \top), (B, \top), (Y, \top), (A, A), (B, B), (Y, Y), (A, r, A)\}$$

Example

$$\{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$$

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$	**and** $\forall i, (X, A_i) \in \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$	**and** $(X, A) \in \mathcal{A}$	**then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$	**and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$

$$\mathcal{A} = \{(A, \top), (B, \top), (Y, \top), (A, A), (B, B), (Y, Y), (A, r, A), (A, B)\}$$

And so on...

Example

$$\{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$$

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$ **and** $\forall i, (X, A_i) \in \mathcal{A}$ **then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$ **and** $(X, A) \in \mathcal{A}$ **then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$ **and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$ **then** add (X, B) to $\mathcal{A}$

$$\mathcal{A} = \{(A, \top), (B, \top), (Y, \top), (A, A), (B, B), (Y, Y), (A, r, A), (A, B)\}$$

And so on...

The pinpointing Problem : minAs

Definition

A minA for a TBox $\mathcal{T}$ with respect to $A \sqsubseteq B$ is a subset $\mathcal{S}$ of $\mathcal{T}$ s.t. $A \sqsubseteq_{\mathcal{S}} B$ but $A \not\sqsubseteq_{\mathcal{S}'} B$ for all $\mathcal{S}' \subset \mathcal{S}$.

Problem : How to find one minA ? How to find all minAs ?

The pinpointing Problem : minAs

Definition

A minA for a TBox $\mathcal{T}$ with respect to $A \sqsubseteq B$ is a subset $\mathcal{S}$ of $\mathcal{T}$ s.t. $A \sqsubseteq_{\mathcal{S}} B$ but $A \not\sqsubseteq_{\mathcal{S}'} B$ for all $\mathcal{S}' \subset \mathcal{S}$.

Problem : How to find one minA ? How to find all minAs ?

The pinpointing Problem : Pinpointing formula

Let's assume each CGI $t \in \mathcal{T}$ is labeled $lab(t) \in lab(\mathcal{T})$. A monotone Boolean formula over $lab(\mathcal{T})$ is a Boolean formula using variables in $lab(\mathcal{T})$, conjunctions, disjunctions and `True`.

Definition

A pinpointing formula for a TBox $\mathcal{T}$ with respect to $A \sqsubseteq B$ is a monotone Boolean ϕ formula over $lab(\mathcal{T})$ such that for each valuation $\mathcal{V} \in lab(\mathcal{T})$

$$A \sqsubseteq_{\mathcal{T}_\mathcal{V}} B \Leftrightarrow \mathcal{V} \text{ satisfies } \phi$$

The pinpointing Problem : Pinpointing formula

Using the pinpointing formula, we can find the minAs. Indeed, if ϕ is a pinpointing formula for $\mathcal{T}$ with respect to $A \sqsubseteq B$ then the set of all MinAs for $\mathcal{T}$ with respect to $A \sqsubseteq B$ is :

$$\{\mathcal{T}_{\mathcal{V}} \mid \mathcal{V} \text{ is a minimal valuation satisfying } \phi\ \}$$

$\Rightarrow$ To compute MinAs, it is enough to find a pinpointing forumla. However, this may cause an exponential blowup in the number of variables.

Example

$$\{ax_1 : \ A \sqsubseteq \exists r.A, ax_2 : \ A \sqsubseteq Y, ax_3 : \ \exists r.Y \sqsubseteq B, ax_4 : \ Y \sqsubseteq B\}$$

In this example, the minAs of $\mathcal{T}$ with respect to $A \sqsubseteq B$ are $\{ax_2, ax_4\}$ and $\{ax_1, ax_2, ax_3\}$.

Therefore, a pinpointing formula is $ax_2 \wedge (ax_4 \vee (ax_1 \wedge ax_3))$.

Summary

Computing the pinpointing formula

Init : assertions $(A, \top)$ and (A, A) receive label t.

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$	**and** $\forall i, (X, A_i) \in \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$	**and** $(X, A) \in \mathcal{A}$	**then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$	**and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$

1. Let ϕ be the conjunction of labels of the CGIs from $\mathcal{T}$ and the assertions from $\mathcal{A}$ used in the rule.
2. If the assertion in the consequence does not yet belong to $\mathcal{A}$, add it with label ϕ.
3. Otherwise, if ψ is the label of the assertion in the consequence and $\psi \not\Rightarrow \phi$, then replace the label of this assertion by $\psi \vee \phi$.

Example

$$\{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$$

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$ **and** $\forall i, (X, A_i) \in \mathcal{A}$ **then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$ **and** $(X, A) \in \mathcal{A}$ **then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$ **and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$ **then** add (X, B) to $\mathcal{A}$

Init :

$$\mathcal{A} = \{(A, \top, \texttt{t}), (B, \top, \texttt{t}), (Y, \top, \texttt{t}), (A, A, \texttt{t}), (B, B, \texttt{t}), (Y, Y, \texttt{t})\}$$

Example

$$\{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$$

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$ **and** $\forall i, (X, A_i) \in \mathcal{A}$ **then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$ **and** $(X, A) \in \mathcal{A}$ **then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$ **and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$ **then** add (X, B) to $\mathcal{A}$

$$\mathcal{A} = \{(A, \top, \mathtt{t}), (B, \top, \mathtt{t}), (Y, \top, \mathtt{t}), (A, A, \mathtt{t}), (B, B, \mathtt{t}), (Y, Y, \mathtt{t}), (A, r, A, ax_1)\}$$

Example

$$\{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$$

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$	**and** $\forall i, (X, A_i) \in \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$	**and** $(X, A) \in \mathcal{A}$	**then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$	**and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$

$$\mathcal{A} = \{(A, \top, \mathtt{t}), (B, \top, \mathtt{t}), (Y, \top, \mathtt{t}), (A, A, \mathtt{t}), (B, B, \mathtt{t}), (Y, Y, \mathtt{t}), (A, r, A, ax_1), (A, B, ax_1 \wedge ax_2 \wedge ax_3)\}$$

And so on...

Example

$$\{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$$

If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$	**and** $\forall i, (X, A_i) \in \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$
If $A \sqsubseteq \exists r.B \in \mathcal{T}$	**and** $(X, A) \in \mathcal{A}$	**then** add (X, r, B) to $\mathcal{A}$
If $\exists r.A \sqsubseteq B \in \mathcal{T}$	**and** $\{(X, r, Y), (Y, A)\} \subseteq \mathcal{A}$	**then** add (X, B) to $\mathcal{A}$

$$\mathcal{A} = \{(A, \top, \texttt{t}), (B, \top, \texttt{t}), (Y, \top, \texttt{t}), (A, A, \texttt{t}), (B, B, \texttt{t}), (Y, Y, \texttt{t}), (A, r, A, ax_1), (A, B, ax_1 \wedge ax_2 \wedge ax_3)\}$$

And so on...

Computing the pinpointing formula : Termination

This algorithm always terminate

- The number of assertion is finite.
- The number of accepting valuation for the label of each assertion only increase (since we always do a disjunction) and there is a finite number of monotone Boolean formula over $lab(\mathcal{T})$.

Computing the pinpointing formula : Correctness

This algorithm return every valid assertion in $\mathcal{T}$, with a valid pinpointing formula

- The first part is true, since the final set $\mathcal{A}$ is the same than the one in the original algorithm.
- The second point is not proven in the paper, but we can see that it is possible to do it by induction.

General case

Note that the algorithm we described only works for normalized TBoxes.

However, we can replace every label of the normalized TBox with the disjunction of its possible sources.

General case

Note that the algorithm we described only works for normalized TBoxes.

However, we can replace every label of the normalized TBox with the disjunction of its possible sources.

Computing one MinA : Algorithm

1. If $A \not\sqsubseteq_{\mathcal{T}} B$, then return no MinA.
2. Initialize $\mathcal{S} := \mathcal{T}$.
3. For $i \in [1, n]$, if $A \sqsubseteq_{\mathcal{S}\backslash\{t_i\}} B$, then t_i is not necessary and we can set

$$\mathcal{S} \leftarrow \mathcal{S}\backslash\{t_i\}$$

4. The resulting set $\mathcal{S}$ is a MinA.

$\Rightarrow$ Take polynomial time

Computing all MinAs : Complexity

$$\mathcal{T}_n := \{B_{i-1} \sqsubseteq P_i \sqcap Q_i, P_i \sqsubseteq B_i, Q_i \sqsubseteq B_i | 1 \leq i \leq n\}$$

Each MinA for $\mathcal{T}_n$ with respects to $B_0 \sqsubseteq B_n$ contains either $P_i \sqsubseteq B_i$ or $Q_i \sqsubseteq B_i$. Therefore, there is 2^n MinAs.

$\Rightarrow$ More generally, the problem of finding if there is a MinA of cardinality $\leq n$ is NP-complete (by reduction of the *hitting set Problem*).

In practice...

SNOMED CT

contains more than 350,000 axioms

$\Rightarrow$ Algorithm for one MinA is impractical....

...Hopefully, we can improve it by first computing a greedy subset of $\mathcal{T}$ from which the subsumption follows, and initialize $\mathcal{S}$ to it.

In practice...

SNOMED CT

contains more than 350,000 axioms

$\Rightarrow$ Algorithm for one MinA is impractical....

...Hopefully, we can improve it by first computing a greedy subset of $\mathcal{T}$ from which the subsumption follows, and initialize $\mathcal{S}$ to it.

Improved algorithm

1. Let ϕ be the conjunction of labels of the CGIs from $\mathcal{T}$ and the assertions from $\mathcal{A}$ used in the rule.
2. If the assertion in the consequence does not yet belong to $\mathcal{A}$, add it with label ϕ.
3. ~~Otherwise, if ψ is the label of the assertion in the consequence and $\psi \not\Rightarrow \phi$, then replace the label of this assertion by $\psi \vee \phi$.~~

$\Rightarrow$ Back to polynomial time, but the final subset for each assertion is not minimal.

\+ Use of a greedy strategy to go from a non normalized TBox to a normalized one.

Improved algorithm

1. Let ϕ be the conjunction of labels of the CGIs from $\mathcal{T}$ and the assertions from $\mathcal{A}$ used in the rule.
2. If the assertion in the consequence does not yet belong to $\mathcal{A}$, add it with label ϕ.
3. ~~Otherwise, if ψ is the label of the assertion in the consequence and $\psi \not\Rightarrow \phi$, then replace the label of this assertion by $\psi \vee \phi$.~~

$\Rightarrow$ Back to polynomial time, but the final subset for each assertion is not minimal.

$+$ Use of a greedy strategy to go from a non normalized TBox to a normalized one.

Experiments

GALEN MEDICAL GROUP

$4,000$ axioms

Experiments

Subsumption algorithm to find the $27,000$ subsumption relationships	14 s
Modified pinpointing algorithm	23 s
Going from normalized TBox to non normalized one	0.27 s
Finding one MinA (optimized)	$9:45$ min
Finding one MinA	7 hours

Experiments

- The average size of the set of axioms before minimizing it with algorithm 1 is 5, which is enough to give an understandable explanation to the user.
- The final set where almost minimal, and the non-minimal sets were in average 2.59% larger than the minimal ones.

Summary

Conclusion

1. The authors propose a PTIME algorithm to compute one MinA in the description logic $\mathcal{EL}$.
2. They show that the problem of finding all MinAs need exponential time.
3. They propose a practical algorithm to find an almost minimal explanation and show that this algorithm gives nice results.

Limits and comments

- In the Algorithm to compute one MinA, it is not clear where the computation of the pinpointing formula is used and where it is used ?
- The practical section is very interesting, maybe a few more experiments (other base), a nice table to show the results, and a better description of the framework would have been great.
- Due to length constraints, some proofs (of validity or termination for instance) are overflown.

Further work

What can be done next ?

- Find pinpointing procedures on other DL and maybe try to generalize it. *Axiom pinpointing in general tableaux* (2010)
- Instead of computing minimal sets which have a given consequence, why not compute maximal sets that do not have a given consequence. We have theoretical results, but not a practical algorithm like here.
- Do more practical experiments and try to optimize the algorithm presented. *Debugging SNOMED CT using axiom pinpointing in the description logic EL* (2008)
- Apply this to OWL for instance. *Explaining inconsistencies in OWL ontologies* (2009)

Thanks for your attention !