Axiom Pinpointing in Description Logics

Franz Baader
TU Dresden

Description Logics

- Family of logic-based knowledge representation languages.
- Many DLs are decidable fragments of first-order logic.
- Close relationship to propositional modal logics.
- Design goal: good compromise between expressiveness and complexity
- Decidability and complexity results for a great variety of DLs and various inference problems, but also implementation of practical systems.
  - very expressive DLs of high worst-case complexity, but with highly optimized "practical" reasoning procedures — *FacT, Racer Pellet, . . .*
  - inexpressive DLs with tractable inference problems, which are expressive enough for certain applications — *CEL, Snorocket QuOnto, . . .*
- Applications: natural language processing, configuration, databases, modelling in engineering domains, ontologies (Web ontology language OWL, biomedical ontologies).

Description logics

Constructors of the expressive DL $\mathcal{ALCN}$:

$C \sqcap D, C \sqcup D, \neg C, \forall r.C, \exists r.C, (\geq n\, r), (\leq n\, r)$

A man	$Human \sqcap \neg Female \sqcap$
that has a rich or beautiful wife	$\exists married\_to.(Rich \sqcup Beautiful) \sqcap$
and at least 3 children,	$(\geq 3\ child) \sqcap$
all of whom are happy	$\forall child.Happy$

Axioms

concept definitions

$Happy\_man \equiv Human \sqcap \ldots$

General concept inclusions (GCIs)

$Human \sqsubseteq \forall child.Human$

$\exists child.Human \sqsubseteq Tax\_Break$

Inferences

Subsumption

$Happy\_man \sqsubseteq Tax\_break$

Satisfiability of concepts

Consistency of knowledge bases

Dresden

The inexpressive Description Logic $\mathcal{EL}$

conjunction $C \sqcap D$,

existential restriction $\exists r.C$,

top concept $\top$

$$Frog \sqsubseteq Animal \sqcap \exists color.Green$$

DL with restricted expressive power

- no value restrictions $\forall r.C$
- can represent large biomedical ontologies: SNOMED CT, Gene Ontology, ...
- $\mathcal{EL}$ has better algorithmic properties than DLs with value restrictions

Dresden

Formal semantics

An interpretation $\mathcal{I}$ has a domain $\Delta^{\mathcal{I}}$ and associates

- concepts C with sets $C^{\mathcal{I}}$, and
- roles r with binary relations $r^{\mathcal{I}}$.

The semantics of the constructors is defined through identities:

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$,
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\exists r.C)^{\mathcal{I}} = \{d \mid \exists e.(d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$.

The interpretation $\mathcal{I}$ is a model of

- the general concept inclusion (GCI) $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- the general TBox $\mathcal{T}$ iff it satisfies all GCIs in $\mathcal{T}$.

Dresden

Subsumption

is concept C a subconcept of concept D?

$$\mathcal{T} \models C \sqsubseteq D \quad \text{iff} \quad C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$$

for all models $\mathcal{I}$ of $\mathcal{T}$

Subsumption in $\mathcal{EL}$ w.r.t. general TBoxes is polynomial.

- This is in strong contrast to the case of DLs with value restrictions, where subsumption w.r.t. general TBoxes is ExpTime-complete.
- Subsumption in $\mathcal{EL}$ w.r.t. general TBoxes remains polynomial if we add the bottom concept, nominals, restricted role-value-maps, and restricted concrete domains.

Error management and explanation

- Large ontologies often contain errors, and thus have unintended consequences.
- Even some of the intended consequences may appear to be unintuitive to users.

Understanding the reasons for unintuitive or unintended consequences can be difficult:

- In the DL version of the medical ontology SNOMED CT, the concept *AmputationOfFinger* is subsumed by *AmputationOfHand*.
- Finding the axioms that are responsible for this among the $> 350\,000$ concept definitions in SNOMED by hand is not easy.
- Pinpointing: compute minimal subsets of the ontology that already have the consequence.

Dresden

Error management and explanation

comes in three different flavours

- Pinpointing: identify the source of the consequence

  minimal subsets of the TBox from which a consequence follows

  MinAs

- Explanation: provide a convincing argument for the consequence

- Correction: provide suggestions for error resolution

  maximal subsets of the TBox from which a consequence does not follow

  ManAs

Dresden

Pinpointing in DLs

example

$\mathcal{T}$

a_1 :	$A \sqsubseteq \exists r.A$
a_2 :	$A \sqsubseteq Y$
a_3 :	$\exists r.Y \sqsubseteq B$
a_4 :	$Y \sqsubseteq B$

$\mathcal{T} \models A \sqsubseteq B$

minimal axiom sets with consequence $A \sqsubseteq B$ (MinAs):

$\{a_2, a_4\}, \ \{a_1, a_2, a_3\}$

pinpointing formula for consequence $A \sqsubseteq B$:

$a_2 \wedge (a_4 \vee (a_1 \wedge a_3))$

monotone Boolean formula whose satisfying valuations correspond to subsets that have the consequence

maximal non-axiom sets, i.e., without consequence $A \sqsubseteq B$ (ManAs):

$\{a_1, a_3, a_4\}, \ \{a_2, a_3\}, \ \{a_1, a_2\}$

Dresden

Pinpointing in DLs

All three possible outputs (MinAs, ManAs, pinpointing formula) contain

- enough information to obtain all subsets that have the consequence
- without requiring additional DL reasoning.

$\Rightarrow$ can be transformed into each other without additional DL reasoning

transformation may be exponential / require the solution of an NP-complete problem

Pinpointing formula to MinAs:

$$a_2 \wedge (a_4 \vee (a_1 \wedge a_3)) \longrightarrow \{a_2, a_4\},\ \{a_1, a_2, a_3\}$$

- minimal satisfying valuations
- disjunctive normal form $(a_2 \wedge a_4) \vee (a_1 \wedge a_2 \wedge a_3)$

Pinpointing in DLs

equivalence of outputs

All three possible outputs (MinAs, ManAs, pinpointing formula) contain

- enough information to obtain all subsets that have the consequence
- without requiring additional DL reasoning.

$\Rightarrow$ can be transformed into each other without additional DL reasoning

transformation may be exponential / require the solution of an NP-complete problem

MinAs to ManAs:

$\{a_2, a_4\}, \ \{a_1, a_2, a_3\} \longrightarrow \{a_1, a_3, a_4\}, \ \{a_2, a_3\}, \ \{a_1, a_2\}$

minimal Hitting sets: $\{a_2\}, \ \{a_1, a_4\}, \ \{a_3, a_4\}$

complement

Approaches to pinpointing

Black Box

- employ existing inference procedure without modification:
  - + highly-optimized implementations can be reused
  - – in the worst-case, the procedure needs to be invoked exponentially often
- naive approach: check for all subsets whether they have the consequence
- more sophisticated approaches work well in practice

Dresden

Approaches to pinpointing

Glass Box

- modify existing inference procedure into one that directly computes minimal subsets or pinpointing formula:
  - + modified procedure is invoked only once
  - – requires new implementation and optimization
- specialized approach: do this for a specific DL and a specific inference procedure
- generic approach: show how a certain class of inference procedures can be generalized to pinpointing procedures

Glass Box Approaches

first developed by modifying tableau-based algorithms

- First introduced in [B. & Hollunder, KR'92] in the context of default reasoning in Description Logic.

  Labeled version of tableau-based algorithm for the DL $\mathcal{ALC}$ (without GCIs) to compute MinAs and ManAs:

  produces pinpointing formula from which both can be derived

- Re-invented in [Schlobach & Cornet, IJCAI'03] to compute minimal unsatisfiable subsets of $\mathcal{ALC}$ TBoxes.

  Labeled tableau-based algorithm similar to the one of B. & Hollunder:

  directly produces all MinAs

- Schlobach's approach extended in [Parsia et al., WWW'05] to more expressive DLs.

- [Lee et al., DL'06] extend approach in [B. & Hollunder, KR'92] to $\mathcal{ALC}$ with GCIs.

- [B., et al., KI'07] introduce labeled variant of the subsumption algorithm for $\mathcal{EL}$ with GCIs.

Glass Box Approach

by modifying a specific
tableau-based algorithm

already there
with label θ

$\forall$-rule: $(\forall r.C)(a) \in \mathcal{A}$ and $r(a,b) \in \mathcal{A}$ $\rightsquigarrow$ add $C(b)$ to $\mathcal{A}$

label φ label ψ label $\theta \vee (\varphi \wedge \psi)$

already there
with label θ

GCI-rule: $C \sqsubseteq D \in \mathcal{T}$ and a occurs in $\mathcal{A}$ $\rightsquigarrow$ add $(\neg C \sqcup D)(a)$ to $\mathcal{A}$

label a_i label $\theta \vee a_i$

Pinpointing in general tableaux

[B. & Penaloza, Tableaux'07]
[B. & Penaloza, JLC'10]

- define a general notion of a tableau system that captures
  - most of the known tableau procedures for DLs
  - also other decision procedures, like the polytime subsumption algorithm for $\mathcal{EL}$, congruence closure, ...
- define the pinpointing extension of a tableau system:
  - show correctness: terminating runs of the pinpointing extension compute a pinpointing formula
  - in general, termination does not transfer to the pinpointing extension
    - ∗ there are terminating tableau systems whose pinpointing extension does not terminate
    - ∗ for a given terminating tableau system, it is undecidable whether its pinpointing extension terminates
- define the notion of ordered forest tableaux:
  - always terminate and so do their pinpointing extensions

Dresden

Automata-based pinpointing

[B. & Penaloza, IJCAR'08]
[B. & Penaloza, JAR'10]

Given set of axioms $\mathcal{T}$ and possible consequence $\mathcal{C}$, automata-based decision procedures

- construct an automaton $\mathcal{A} = \mathcal{A}(\mathcal{T}, \mathcal{C})$.
- perform emptiness test for $\mathcal{A}$.
- $\mathcal{T} \models \mathcal{C}$ iff $L(\mathcal{A}) = \emptyset$.

- Define the notion of an axiomatic automaton $\mathcal{A}(\mathcal{T}, \mathcal{C})$ that "contains" all the automata $\mathcal{A}(\mathcal{S}, \mathcal{C})$ for $\mathcal{S} \subseteq \mathcal{T}$
- Transform a given axiomatic automaton into a weighted automaton whose behaviour is a pinpoining formula
- Show how to compute the behaviour.

Dresden

Black Box Approaches

[Kalyanpur et al., ISWC'07]
[B. & Suntisrivaraporn, KR-MED'08]
[Suntisrivaraporn, 2009]
[Horridge et al., SUM'09]

- naive approach that considers all subsets of $\mathcal{T}$ and tests which of them has the consequence is not practical for large ontologies like SNOMED CT ($> 360\,000$ axioms)

- more practical approaches are all based on the following idea:

  (a) Design an efficient procedure for extracting one MinA.

  (b) Use this procedure within Reiter's Hitting Set Tree algorithm to compute all MinAs.

- useful optimization: first compute a subset of the ontology that is

  - easy to compute
  - rather small
  - contains all MinAs

  Then apply the HST approach to this subset.

Extracting one MinA

extracting a MinA $\mathcal{S}$ from a set of axioms $\mathcal{T}$

Naive linear algorithm:

- Go through the axioms according to some fixed order.
- For each axiom, check whether the consequence still holds if it is removed from the current axiom set.
- If yes, then remove it; otherwise keep it.

– Number of calls to inference procedure linear in $|\mathcal{T}|$

+ Very simple, no overhead.

Logarithmic algorithm:

- Partition $\mathcal{T}$ into two halves
- For each half, check whether the consequence still holds if it is removed from the current ontology.
- If yes for one of them, then recurse on this half.
- Otherwise, do "something smart."

+ Number of calls to inference procedure logarithmic in $|\mathcal{T}|$, but still linear in $|\mathcal{S}|$

– Higher overhead, which may not pay off if $|\mathcal{T}|/|\mathcal{S}|$ is small.

Extracting one MinA

extracting a MinA $\mathcal{S}$ from a set of axioms $\mathcal{T}$

Naive linear algorithm:

- Go through the axioms according to some fixed order.
- For each a… er the c… it is … on- t…
- If yes, then remove it; otherwise keep it.

– Number of calls to inference procedure linear in $|\mathcal{T}|$

+ Very simple, no overhead.

not feasible
for SNOMED CT

Logarithmic algorithm:

- Partition $\mathcal{T}$ into two halves
- For each half, check whether the consequence st… it is removed f… gy.
- I… en re- c…
- Otherwise, do "something smart."

+ Number of calls to inference procedure logarithmic in $|\mathcal{T}|$, but still linear in $|\mathcal{S}|$

– Higher overhead, which may not pay off if $|\mathcal{T}|/|\mathcal{S}|$ is small.

feasible
for SNOMED CT

Dresden

Extracting one MinA

extracting a MinA $\mathcal{S}$ from a set of axioms $\mathcal{T}$

Naive linear algorithm:

- Go through the axioms according to some fixed order.
- For each a… …er the c… …it is … …on- t…
- If yes, then remove it; otherwise keep it.

– Number of calls to inference procedure linear in $|\mathcal{T}|$

\+ Very simple, no overhead.

not feasible

for SNOMED CT

Logarithmic algorithm:

- Partition $\mathcal{T}$ into two halves
- For each half, check whether the consequence still… …is removed f… …y.
- I… …en re- c…
- Otherwise, do "something smart."

\+ Number of calls to inference procedure logarithmic in $|\mathcal{T}|$, but still linear in $|\mathcal{S}|$

– Higher overhead, which may not pay off if $|\mathcal{T}|/|\mathcal{S}|$ is small.

BUT

it takes quite long

Dresden

Extracting one MinA

experimental results for SNOMED CT

- The amputation example has exactly one MinA, which has cardinality 6.
  - The logarithmic algorithm can extract this MinA, but take 26 min.
  - First computing reachability based module and then applying linear algorithm performs much better: 0.54 sec

$$
\begin{array}{rcl}
\mathsf{direct\text{-}procedure\text{-}site} & \sqsubseteq & \mathsf{procedure\text{-}site} \\
\mathsf{AmputationOfFinger} & \sqsubseteq & \mathsf{AmputationOfFingerWithoutThumb} \\
\mathsf{AmputationOfFingerWithoutThumb} & \equiv & \mathsf{HandExcision} \sqcap \\
 & & \exists \mathsf{roleGroup}.(\exists \mathsf{direct\text{-}procedure\text{-}site}.\mathsf{Finger}_S \sqcap \exists \mathsf{method}.\mathsf{Amputation}) \\
\mathsf{AmputationOfHand} & \equiv & \mathsf{HandExcision} \sqcap \\
 & & \exists \mathsf{roleGroup}.(\exists \mathsf{procedure\text{-}site}.\mathsf{Hand}_S \sqcap \exists \mathsf{method}.\mathsf{Amputation}) \\
\mathsf{Finger}_S & \sqsubseteq & \mathsf{DigitOfHand}_S \sqcap \mathsf{Hand}_P \\
\mathsf{Hand}_P & \sqsubseteq & \mathsf{Hand}_S \sqcap \mathsf{UpperExtremity}_P
\end{array}
$$

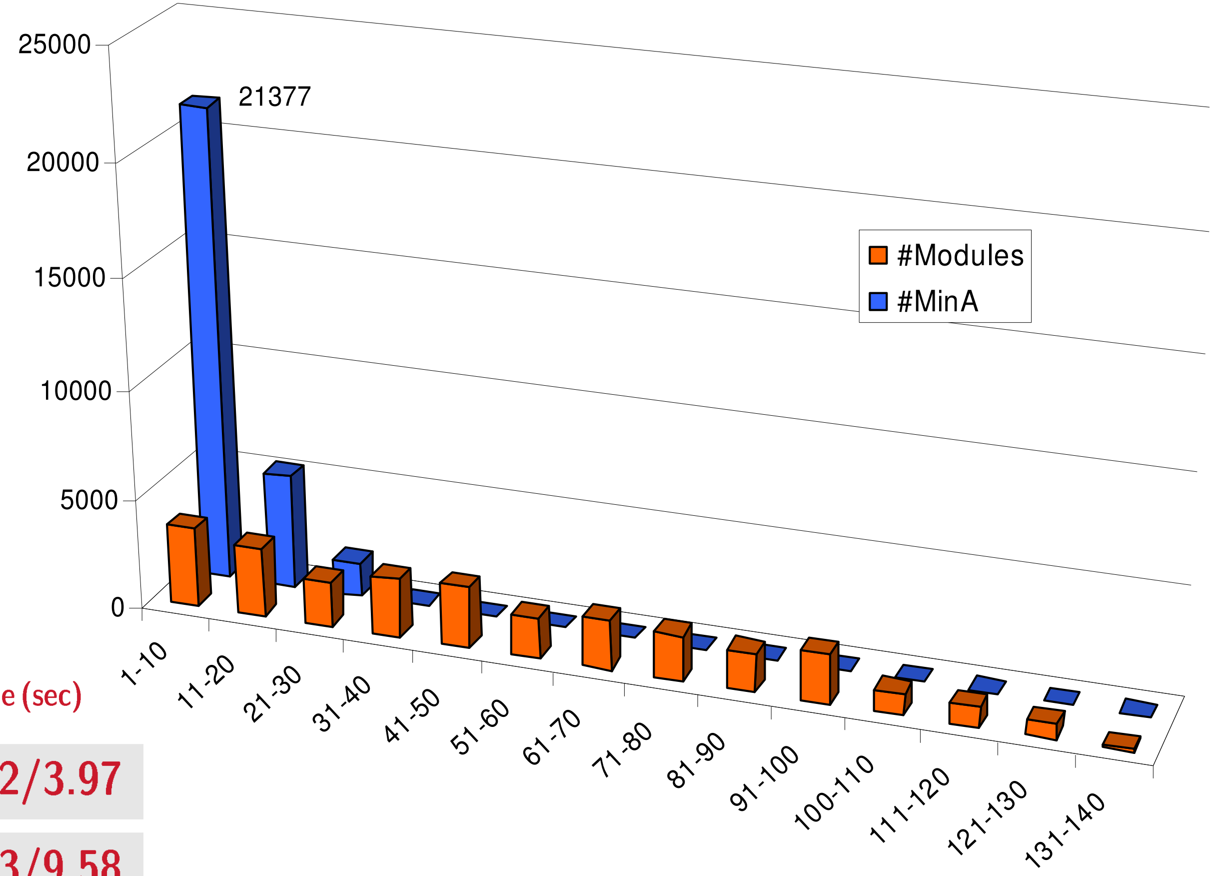
Dresden

Extracting one MinA

experimental results for SNOMED CT

[B. & Suntisrivaraporn, KR-MED'08]

considered 27 477 subsumptions



	time (sec)
extract module	0.02/3.97
logarithmic alg.	1.03/9.58
linear alg.	0.67/5.04

Dresden

Extracting ALL MinAs

experimental results for SNOMED CT
on 27 477 subsumptions

[Suntisrivaraporn, 2009]

Number of MinAs:

- 60% have only one MinA
- 25% have 2–9 MinAs — Easy Samples
- 15% have ≥ 10 MinAs — Hard Samples — computed only the first 10

Samples	Time to extract module $\mathcal{O}_A^{\text{SNOMED}}$ (avg/max)	HST search time excl. subs. calls (avg/max)	♯Subs. calls (avg/max)	Total subs. testing time (avg/max)
easy-samples	0.01 / 2.06	0.07 / 44.08	177.60 / 4 732	8.80 / 131.97
hard-samples	0.02 / 3.96	0.09 / 39.90	769.98 / 4 308	37.77 / 375.68

Table 6.10: Time results (second) of the modularization-based HST pinpointing algorithm on $\mathcal{O}^{\text{SNOMED}}$.

Extracting ALL MinAs

experimental results for SNOMED CT
on 27 477 subsumptions

[Suntisrivaraporn, 2009]

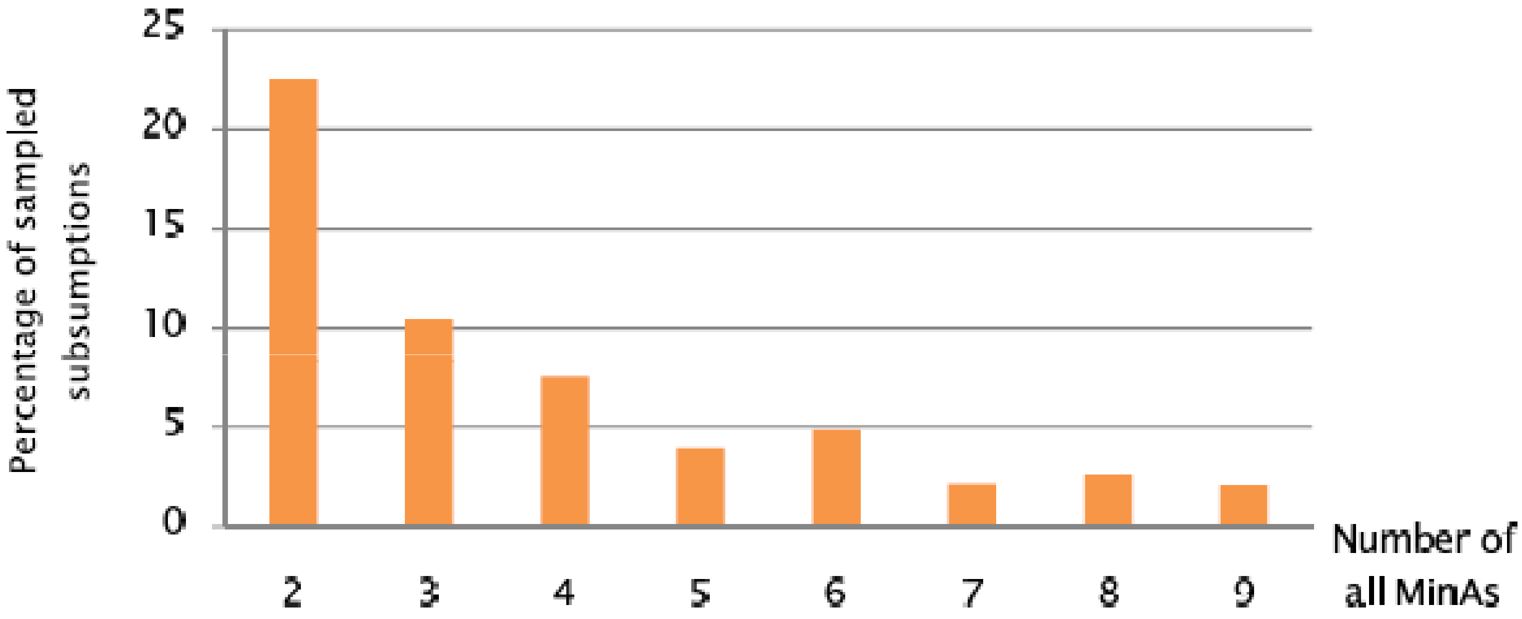


Figure 6.10: Relative frequency of the numbers of *all* MinAs for easy-samples in $\mathcal{O}^{\text{SNOMED}}$.

Dresden

Complexity

of pinpointing in $\mathcal{EL}$

[B. et al., KI'07]

The number of MinAs can become exponential in the cardinality of $\mathcal{T}$:

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

$$\mathcal{T}_n \models B_0 \sqsubseteq B_n$$

- $\mathcal{T}_n$ consists of $3n$ GCIs.
- The consequence $B_0 \sqsubseteq B_n$ has 2^n MinAs.

Dresden

Complexity

of pinpointing in $\mathcal{EL}$

[B. et al., KI'07]

Determining the least cardinality of a MinA is intractable:

The following problem is NP-complete:

> Given: general $\mathcal{EL}$ TBox $\mathcal{T}$, concept names A, B, natural number n
>
> Question: is there a subset $\mathcal{T}'$ of $\mathcal{T}$ of cardinality $\leq n$ with $\mathcal{T}' \models A \sqsubseteq B$?

Reduction from the NP-complete Hitting Set Problem:

> Given: finite sets $S_1, \ldots, S_k$, natural number n
>
> Question: is there a set S of cardinality $\leq n$ with $S \cap S_i \neq \emptyset$ for $i = 1, \ldots, k$?

Dresden

Complexity

of pinpointing in $\mathcal{EL}$

[B. et al., KI'07]

$$S_1 = \{p_{11}, \ldots, p_{1\ell_1}\}, \quad \ldots, \quad S_k = \{p_{k1}, \ldots, p_{k\ell_k}\}$$

$$\downarrow$$

$$\begin{aligned} \mathcal{T} \quad := \quad & \{P_{ij} \sqsubseteq Q_i \mid 1 \leq i \leq k, 1 \leq j \leq \ell_i\} \cup \\ & \{Q_1 \sqcap \ldots \sqcap Q_k \sqsubseteq B\} \cup \\ & \{A \sqsubseteq P_{ij} \mid 1 \leq i \leq k, 1 \leq j \leq \ell_i\} \end{aligned}$$

$S_1, \ldots, S_k$ has a Hitting Set of cardinality $\leq n$.

iff

There is $\mathcal{T}' \subseteq \mathcal{T}$ of cardinality $\leq n + k + 1$ with $\mathcal{T}' \models A \sqsubseteq B$.

Dresden

Complexity

of pinpointing in $\mathcal{EL}$

[Penaloza & Sertkaya, KR'10]

Another intractable problem for MinAs: axiom relevance

Is a given axiom a possible culprit for an erroneous consequence?

Given: general $\mathcal{EL}$ TBox $\mathcal{T}$, concept names A, B, GCI $C \sqsubseteq D \in \mathcal{T}$

Question: is there a MinA $\mathcal{S}$ for $A \sqsubseteq B$ in $\mathcal{T}$ such that $C \sqsubseteq D \in \mathcal{S}$?

This problem is also NP-complete!

Dresden

Enumeration Complexity

of pinpointing in $\mathcal{EL}$

[Penaloza & Sertkaya, KR'10]

- We have seen: the number of MinAs may be exponential.
- Thus, it may take exponential time to enumerate all MinAs.
- What if the number of MinAs is actually polynomial?
  May it still take exponential time to compute them?

Output polynomiality

An algorithm for enumerating all MinAs is output polynomial iff it runs in time polynomial in the size of the TBox and the size of all MinAs.

Dresden

Enumeration Complexity

of pinpointing in $\mathcal{EL}$

[Penaloza & Sertkaya, KR'10]

Unless P=NP, there is no output polynomial algorithm for enumerating all MinAs in $\mathcal{EL}$.

This is an easy consequence of the fact that the following problem is coNP-complete:

Given: general $\mathcal{EL}$ TBox $\mathcal{T}$, concept names A, B, set $\mathcal{M}$ of subsets of $\mathcal{T}$.

Question: is $\mathcal{M}$ the set of all MinAs of $A \sqsubseteq B$ w.r.t. $\mathcal{T}$?

Dresden

Questions?