Minimal module extraction from *DL-Lite* ontologies

Roman Kontchakov

School of Computer Science and Inf. Systems, Birkbeck College, London

http://www.dcs.bbk.ac.uk/~roman

joint work with

Luca Pulina, Frank Wolter and Michael Zakharyaschev

Large-scale ontologies

- Life-sciences, healthcare, and other knowledge intensive areas depend on having a **common language** for gathering and sharing knowledge
- Such a common language is provided by **reference terminologies**
- Examples:
  - SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms)
  - NCI (National Cancer Institute Ontology)
  - FMA (Foundational Model of Anatomy)
  - GALEN
  - ...
- Typical size: at least **50,000** terms and axioms
- Trend towards axiomatising reference terminologies in **('lightweight') description logics**

Description logic $\mathcal{ALCQI}$

Vocabulary: $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ an **interpretation**

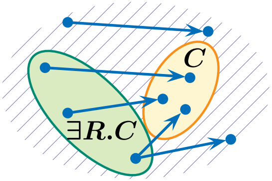
- individuals $a_0, a_1, \ldots$ $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
  (e.g., john, mary)
- concept names $A_0, A_1, \ldots$ $A_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
  (e.g., Person, Female)
- role names $R_0, R_1, \ldots$ $R_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
  (e.g., hasChild, loves)
- roles

$$R \quad ::= \quad R_i \quad | \quad R_i^-$$

$$(R_i^-)^{\mathcal{I}} = \{(y, x) \mid (x, y) \in R_i^{\mathcal{I}}\}$$

- concepts

$$C \quad ::= \quad A_i \mid \neg C \mid C_1 \sqcap C_2 \mid \exists R.C \mid \forall R.C \mid \geq q R.C$$



$$\forall R.C \equiv \neg(\exists R.\neg C)$$

'there are at least q distinct R-successors that are in C'

Description logic $\mathcal{ALCQI}$ (cont.)

knowledge base $\mathcal{K}$ = TBox $\mathcal{T}$ + ABox $\mathcal{A}$

- $\mathcal{T}$ is a set of **terminological axioms** of the form $C \sqsubseteq D$
- $\mathcal{A}$ is a set of **assertional axioms** of the form $C(a)$ and $R(a, b)$

Reasoning: – satisfiability $\mathcal{K}$

is there a *model* $\mathcal{I}$ for $\mathcal{K}$ ($\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$)

– subsumption $\mathcal{K} \models C \sqsubseteq D$

$\mathcal{I} \models C \sqsubseteq D$, for each $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$

– instance checking $\mathcal{K} \models C(a)$

$a^{\mathcal{I}} \in C^{\mathcal{I}}$, for each $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$

– query answering $\mathcal{K} \models q(\vec{a})$, $q(\vec{a})$ a positive existential formula

$\mathcal{I} \models q(a)$ (as a first-order structure), for each $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$

OWL 1.0 DL is based on $\mathcal{SHOIQ}(D)$,

$\mathcal{ALCQI}$ + role inclusions + nominals + transitive roles + concrete domains

OWL 2.0 on $\mathcal{SROIQ}(D)$

$\mathcal{SHOIQ}(D)$ + role chains + disjoint roles + self (diagonal)

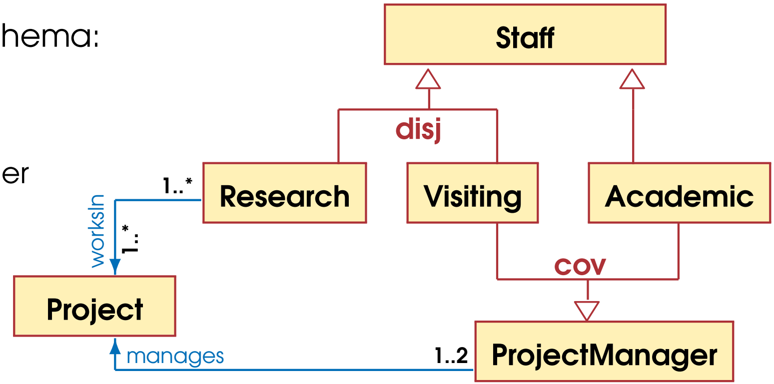
Developing and maintaining ontologies

- **versions:**
  comparing **logical consequences** over some common vocabulary Σ
  not the syntactic form of the axioms (as in `diff`)
- **refinement:**
  adding new axioms but **preserving** the relationships
  between terms of a certain part Σ of the vocabulary
- **reuse:**
  importing an ontology and using its vocabulary Σ as originally defined
  (relationships between terms of Σ should not change)
- **module extraction:**
  computing a subset $\mathcal{M}$ (ideally as small as possible) of an ontology $\mathcal{T}$ that
  'says' the same about Σ as $\mathcal{T}$

new types of reasoning problems

DL-Lite: Description Logic for Databases

A fragment of a conceptual schema:



Translating into DL:

$\exists$ manages.$\top$ $\sqsubseteq$ ProjectManager

$\exists$ manages$^-$.$\top$ $\sqsubseteq$ Project

Project $\sqsubseteq$ $\exists$ manages$^-$.$\top$

≥ 3 manages$^-$.$\top$ $\sqsubseteq$ $\bot$

Research $\sqcap$ Visiting $\sqsubseteq$ $\bot$

Academic $\sqsubseteq$ ProjectManager

ProjectManager $\sqsubseteq$ Academic $\sqcup$ Visiting

...

***DL-Lite*$_{bool}$**

$$R \ ::= \ P \ | \ P^-$$
$$B \ ::= \ \bot \ | \ A_i \ | \ \exists R \ | \ \geq q\, R$$
$$C \ ::= \ B \ | \ \neg C \ | \ C_1 \sqcap C_2 \ | \ C_1 \sqcup C_2$$

TBox axioms: $C_1 \sqsubseteq C_2$ **ABox assertions:** $C(a)$, $R(b, c)$

Essentially positive existential queries: $\exists \vec{y} \varphi(\vec{x}, \vec{y})$, built from $C(t)$, $R(t, t')$, $\wedge$, $\vee$

Σ-entailment and Σ-inseparability

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes and Σ a **signature** (concept and role names)

When do $\mathcal{T}_1$ and $\mathcal{T}_2$ 'say' the same about Σ?

- $\mathcal{T}_1$ **Σ-concept entails** $\mathcal{T}_2$ if, for all Σ-concept inclusions $C \sqsubseteq D$,

  $\mathcal{T}_1 \preceq^c_\Sigma \mathcal{T}_2$ $\qquad$ $\mathcal{T}_1 \models C \sqsubseteq D$ implies $\mathcal{T}_2 \models C \sqsubseteq D$

- $\mathcal{T}_1$ **Σ-query entails** $\mathcal{T}_2$ if, for all Σ-queries $q(\vec{x})$ and ABoxes $\mathcal{A}$,

  $\mathcal{T}_1 \preceq^q_\Sigma \mathcal{T}_2$ $\qquad$ $(\mathcal{T}_1, \mathcal{A}) \models q(\vec{a})$ implies $(\mathcal{T}_2, \mathcal{A}) \models q(\vec{a})$, for all $\vec{a}$

- ...

- $\mathcal{T}_1$ **Σ-model entails** $\mathcal{T}_2$ if, for all Σ-interpretations $\mathcal{I}$,

  $\mathcal{T}_1 \preceq^m_\Sigma \mathcal{T}_2$ $\qquad$ $\exists\, \mathcal{I}_1 \supseteq \mathcal{I}\ \mathcal{I}_1 \models \mathcal{T}_1$ implies $\exists\, \mathcal{I}_2 \supseteq \mathcal{I}\ \mathcal{I}_2 \models \mathcal{T}_2$

- $\mathcal{T}_1$ and $\mathcal{T}_2$ are S_Σ **(concept/query/model) inseparable** if

  $\mathcal{T}_1 \equiv^S_\Sigma \mathcal{T}_2$ $\qquad$ $\mathcal{T}_1 \preceq^S_\Sigma \mathcal{T}_2$ and $\mathcal{T}_2 \preceq^S_\Sigma \mathcal{T}_1$

Σ-inseparability: Examples

Example 1. $\Sigma = \{\mathsf{Lecturer}, \mathsf{Course}\}$

$$\mathcal{T}_1 = \emptyset, \qquad \mathcal{T}_2 = \{\mathsf{Lecturer} \sqsubseteq \exists \mathsf{teaches}, \ \exists \mathsf{teaches}^- \sqsubseteq \mathsf{Course}\}$$

- Is $\mathcal{T}_1 \equiv^c_\Sigma \mathcal{T}_2$?
- Is $\mathcal{T}_1 \equiv^q_\Sigma \mathcal{T}_2$?

Take $\mathcal{A} = \{\mathsf{Lecturer}(a)\}$, $q = \exists y\, \mathsf{Course}(y)$. Then $(\mathcal{T}_1, \mathcal{A}) \not\models q$ but $(\mathcal{T}_2, \mathcal{A}) \models q$

Example 2. $\Sigma = \{\mathsf{Lecturer}\}$

$$\mathcal{T}_1 = \emptyset, \quad \mathcal{T}_2 = \{\mathsf{Lecturer} \sqsubseteq \exists \mathsf{teaches}, \ \mathsf{Lecturer} \sqcap \exists \mathsf{teaches}^- \sqsubseteq \bot\}$$

- Is $\mathcal{T}_1 \equiv^c_\Sigma \mathcal{T}_2$?
- Is $\mathcal{T}_1 \equiv^q_\Sigma \mathcal{T}_2$?

Take $\mathcal{A} = \{\mathsf{Lecturer}(a)\}$, $q = \exists y\, \neg \mathsf{Lecturer}(y)$. Then $(\mathcal{T}_1, \mathcal{A}) \not\models q$ and $(\mathcal{T}_2, \mathcal{A}) \models q$

Σ-inseparability: Examples (cont.)

Example 3. Let $\mathcal{T}_1$ contain the axioms

$$\begin{aligned} \text{Research} &\sqsubseteq \exists\text{worksIn}, & \exists\text{worksIn}^- &\sqsubseteq \text{Project}, \\ \text{Project} &\sqsubseteq \exists\text{manages}^-, & \exists\text{manages} &\sqsubseteq \text{Academic} \sqcup \text{Visiting}, \\ \exists\text{teaches} &\sqsubseteq \text{Academic} \sqcup \text{Research}, & \text{Academic} &\sqsubseteq \exists\text{teaches} \sqcap \leq 1\,\text{teaches}, \\ \text{Research} \sqcap \text{Visiting} &\sqsubseteq \bot, & \exists\text{writes} &\sqsubseteq \text{Academic} \sqcup \text{Research}, \end{aligned}$$

$\mathcal{T}_2 = \mathcal{T}_1 \cup \{\text{Visiting} \sqsubseteq\ \geq 2\,\text{writes}\}$ and $\Sigma = \{\text{teaches}\}$

- $\mathcal{T}_1 \equiv^c_\Sigma \mathcal{T}_2$

  $\mathcal{T}_2 \models \text{Visiting} \sqsubseteq \text{Academic}$, but nothing new in the signature Σ

- $\mathcal{T}_1 \not\equiv^q_\Sigma \mathcal{T}_2$:

  $\mathcal{A} = \{\text{teaches}(a, b), \text{teaches}(a, c)\}$

  $q = \exists x\ ((\exists\text{teaches})(x) \land (\leq 1\,\text{teaches})(x))$

  'is there anybody who teaches precisely one module?'

$(\mathcal{T}_1, \mathcal{A}) \not\models q$

$(\mathcal{T}_2, \mathcal{A}) \models q$

Σ-entailment: semantic criteria

Let Q be a set of numerical parameters and Σ a signature

ΣQ-concepts B: $A_i \in \Sigma$ and $(\geq q\, R)$ with $q \in Q$ and $R \in \Sigma$

ΣQ-type t is a set of ΣQ-concepts containing B or $\neg B$ (but not both), for all B

For a TBox $\mathcal{T}$,

a ΣQ-type t is **$\mathcal{T}$-realisable** if t is satisfied in a model of $\mathcal{T}$

(i.e., there is a $\mathcal{I}$ of $\mathcal{T}$ and a point w in it such that $w \in B^{\mathcal{I}}$ iff $B \in t$)

a set Ξ of ΣQ-types is **precisely $\mathcal{T}$-realisable** if there is a model of $\mathcal{T}$ realising precisely the types from Ξ

Theorem. Let Q denote the set of parameters occurring in $\mathcal{T}_1 \cup \mathcal{T}_2$

$\mathcal{T}_1$ **Σ-concept entails** $\mathcal{T}_2$ iff every $\mathcal{T}_1$-realisable ΣQ-type is **$\mathcal{T}_2$-realisable**

$\mathcal{T}_1$ **Σ-query entails** $\mathcal{T}_2$ iff every precisely $\mathcal{T}_1$-realisable set Ξ of ΣQ-types is **precisely $\mathcal{T}_2$-realisable**

Σ-inseparability: complexity

Theorem.

- Deciding Σ-concept and Σ-query inseparability is **Π_2^p-complete**
- Deciding Σ-model inseparability is NExpTime**-complete**
- Can be simpler for various fragments of *DL-Lite*$_{bool}$

  E.g. deciding Σ-concept and Σ-query inseparability for *DL-Lite*$_{horn}$ is coNP**-complete**

NB. Π_2^p-completeness means that the problem can be encoded as satisfiability of $\forall\exists$ quantified Boolean formulas

Various QBF solvers can be used to check Σ-concept and Σ-query inseparability

NB. Inseparability is much harder for $\mathcal{ALC}$ and other non-'Lite' DLs (2ExpTime-complete for $\mathcal{ALC}$, undecidable for $\mathcal{ALCQIO}$)

Encoding Σ-concept entailment in QBF

Let $\mathcal{T}$ be a TBox, $\boldsymbol{Q}$ a set of numerical parameters and $\boldsymbol{t}$ a **sig**$(\mathcal{T})\boldsymbol{Q}$-type

'$\boldsymbol{t}_0$ is $\mathcal{T}$-realisable with $\boldsymbol{t}_1, \dots, \boldsymbol{t}_n$ being witnesses' $=$ $\Phi_{\mathcal{T}}(\boldsymbol{b}_0, \boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$

propositional formula

$\boldsymbol{b}_j$ is the vector of all propositional variables $\boldsymbol{B}^*$ of the type $\boldsymbol{t}_j$

Then the condition

'every $\mathcal{T}_1$-realisable ΣQ-type $\boldsymbol{t}$ is $\mathcal{T}_2$-realisable'

is described by the following QBF

$$\forall \boldsymbol{b}_0^{\Sigma Q}\Big[\exists \boldsymbol{b}_0^{\mathcal{T}_1\setminus\Sigma Q}\exists \boldsymbol{b}_1^{\mathcal{T}_1}\dots\exists \boldsymbol{b}_{n_1}^{\mathcal{T}_1}\;\Phi_{\mathcal{T}_1}(\boldsymbol{b}_0^{\Sigma Q}\cdot \boldsymbol{b}_0^{\mathcal{T}_1\setminus\Sigma Q}, \boldsymbol{b}_1^{\mathcal{T}_1},\dots,\boldsymbol{b}_{n_1}^{\mathcal{T}_1}) \quad\to\quad \exists \boldsymbol{b}_0^{\mathcal{T}_2\setminus\Sigma Q}\exists \boldsymbol{b}_1^{\mathcal{T}_2}\dots\exists \boldsymbol{b}_{n_2}^{\mathcal{T}_2}\;\Phi_{\mathcal{T}_2}(\boldsymbol{b}_0^{\Sigma Q}\cdot \boldsymbol{b}_0^{\mathcal{T}_2\setminus\Sigma Q}, \boldsymbol{b}_1^{\mathcal{T}_2},\dots,\boldsymbol{b}_{n_2}^{\mathcal{T}_2})\Big]$$

($\boldsymbol{b}_0^{\Sigma Q}$ is the $\boldsymbol{\Sigma Q}$-part of $\boldsymbol{b}_0$ and $\boldsymbol{b}_0^{\mathcal{T}_i\setminus\Sigma Q}$ contains the rest of the variables)

Experiments

TBox instances (standard Department Ontology + ICNARC)

series	description	no. of instances	axioms		basic concepts		
			$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_1$	$\mathcal{T}_2$	Σ
NN	$\mathcal{T}_1$ does not Σ-concept entail $\mathcal{T}_2$	840	59–308	74–396	47–250	49–300	5–103
YN	$\mathcal{T}_1$ Σ-concept but not Σ-query entails $\mathcal{T}_2$	504	56–302	77–382	44–246	58–298	6–89
YY	$\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$	624	43–178	43–222	40–158	40–188	5–64

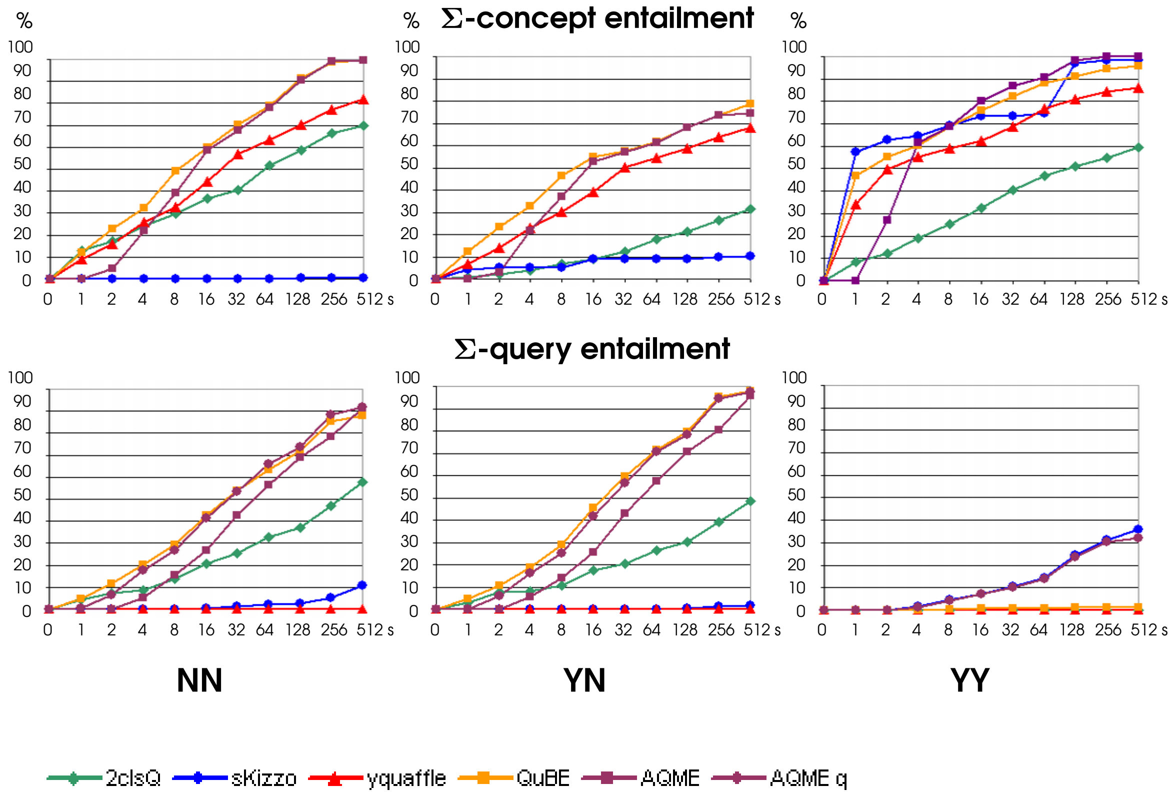
QBF solvers

- sKizzo 0.8.2
- 2clsQ
- yQuaffle
- QuBE 6.4
- **AQME**

series	Σ-concept entailment QBF		Σ-query entailment QBF	
	variables	clauses	variables	clauses
NN	1,469–48,631	2,391–74,621	1,715–60,499	5,763–1,217,151
YN	1,460–46,873	2,352–71,177	1,755–59,397	7,006–1,122,361
YY	1,006–16,033	1,420–23,363	1,202–20,513	2,963–204,889

number of clauses is **linear** **quadratic**
(in the number of roles)

Experimental results: percentage of solved instances



What is a module?

Let $\mathcal{S}$ be an inseparability relation, $\mathcal{T}$ a TBox and Σ a signature.

$\mathcal{M} \subseteq \mathcal{T}$ is

- an **S_Σ-module of $\mathcal{T}$** if $\mathcal{M} \equiv_\Sigma^S \mathcal{T}$
- a **self-contained S_Σ-module of $\mathcal{T}$** if $\mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})}^S \mathcal{T}$
- a **depleting S_Σ-module of $\mathcal{T}$** if $\emptyset \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})}^S \mathcal{T} \setminus \mathcal{M}$

$\mathcal{M}$ is a **minimal module of $\mathcal{T}$** if it can't be made smaller

Facts:

- depleting $\equiv_\Sigma^q$-module $\Rightarrow$ self-contained $\equiv_\Sigma^q$-module $\Rightarrow$ $\equiv_\Sigma^q$-module
- self-contained $\equiv_\Sigma^c$-module $\Rightarrow$ $\equiv_\Sigma^c$-module
- There is precisely **one** minimal depleting $\equiv_\Sigma^q$-module
- There may be (exponentially) many minimal modules of other types

Modules for $\Sigma = \{\mathsf{Publisher}\}$

(1) Publisher $\sqsubseteq$ ∃pubHasDistrib

(2) ∃pubHasDistrib^{-} $\sqsubseteq$ Distributor

(3) Publisher $\sqsubseteq$ ¬Distributor

(4) ∃pubHasDistrib $\sqsubseteq$ Publisher

(5) Publisher $\sqsubseteq$ ≤ 1 pubHasDistrib

(6) Role $\sqsubseteq$ ¬Distributor

(7) User $\sqsubseteq$ ¬Distributor

(8) Publisher $\sqsubseteq$ ∃pubAdmedBy

(9) ∃pubAdmedBy^{-} $\sqsubseteq$ AdmUser $\sqcup$ BookUser

(10) AdmUser $\sqsubseteq$ User

(11) BookUser $\sqsubseteq$ User

(12) User $\sqsubseteq$ ∃hasRole

(13) ∃hasRole^{-} $\sqsubseteq$ Role

(14) Role $\sqsubseteq$ ¬Publisher

(15) User $\sqsubseteq$ ¬Publisher

(16) Role $\sqsubseteq$ ¬User

(17) User $\sqsubseteq$ ∃userAdmedBy

(18) ∃userAdmedBy^{-} $\sqsubseteq$ AdmUser

(19) ∃userAdmedBy $\sqsubseteq$ User

(20) ∃pubAdmedBy $\sqsubseteq$ Publisher

the minimal S^{c}_{Σ}-module is $\emptyset$

minimal S^{q}_{Σ}-modules of $\mathcal{T}$: $\mathcal{M}_D$, $\mathcal{M}_R$ and $\mathcal{M}_U$

the minimal depleting S^{q}_{Σ}-module is $\mathcal{T}$

Module extraction algorithms

- minimal S_Σ-module

```
input T, Σ
M := T
for each α ∈ M do
  if M \ {α} ≡^S_Σ M then M := M \ {α}
end for
output M
```

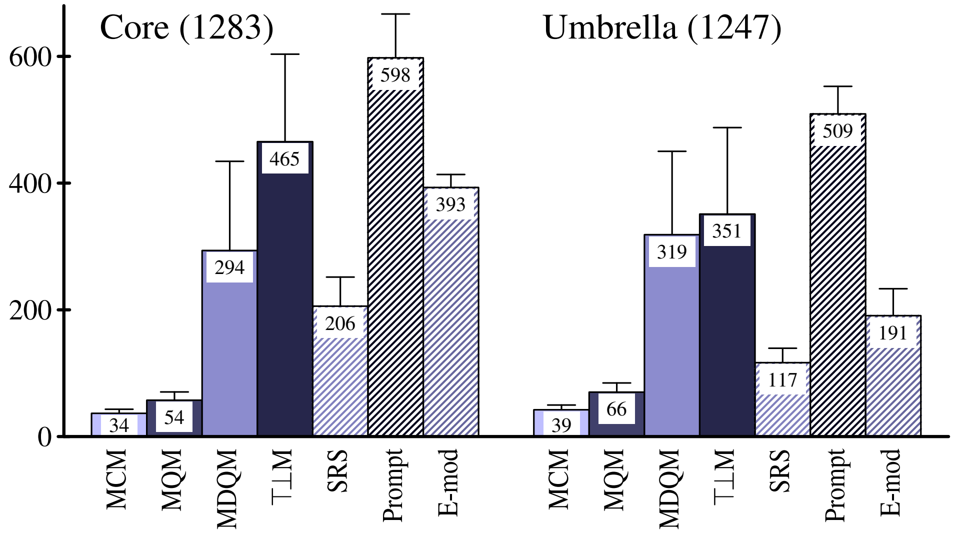
$$
\begin{array}{l}
\texttt{input } \mathcal{T}, \Sigma \\
\mathcal{M} := \mathcal{T} \\
\texttt{for each } \alpha \in \mathcal{M} \texttt{ do} \\
\quad \texttt{if } \mathcal{M} \setminus \{\alpha\} \equiv^{S}_{\Sigma} \mathcal{M} \texttt{ then } \mathcal{M} := \mathcal{M} \setminus \{\alpha\} \\
\texttt{end for} \\
\texttt{output } \mathcal{M}
\end{array}
$$

NB: depends on the order of axioms in $\mathcal{T}$

- minimal depleting S_Σ-module

$$
\begin{array}{l}
\texttt{input } \mathcal{T}, \Sigma \\
\mathcal{T}' := \mathcal{T};\ \Gamma := \Sigma;\ \mathcal{W} := \emptyset \\
\texttt{while } \mathcal{T}' \setminus \mathcal{W} \neq \emptyset \texttt{ do} \\
\quad \texttt{choose } \alpha \in \mathcal{T}' \setminus \mathcal{W} \\
\quad \mathcal{W} := \mathcal{W} \cup \{\alpha\} \\
\quad \texttt{if } \mathcal{W} \not\equiv^{S}_{\Gamma} \emptyset \texttt{ then} \\
\qquad \mathcal{T}' := \mathcal{T}' \setminus \{\alpha\};\ \mathcal{W} := \emptyset;\ \Gamma := \Gamma \cup \mathsf{sig}(\alpha) \\
\quad \texttt{endif} \\
\texttt{end while} \\
\texttt{output } \mathcal{T} \setminus \mathcal{T}'
\end{array}
$$

Practical minimal module extraction



Module sizes and standard deviation for $|\Sigma| = 10$