Modularity in Ontologies: Incremental Classification, Logical Diff and Forgetting

Thomas Schneider[1] *Dirk Walther*[2]

[1]Department of Computer Science, University of Bremen, Germany

[2]Center for Advancing Electronics Dresden, TU Dresden, Germany

ESSLLI, 16 August 2013

Plan for today

1. Incremental Classification
2. Logical Difference for Ontology Versioning
3. Forgetting and Uniform Interpolation
4. Conclusion

And now . . .

Incremental Classification: Motivation

- early detection of modelling errors is important for developing and maintaining ontologies
- frequent classification of ontologies required
- issue:
  - long response times from reasoners
- solutions:
  - classification algorithms and optimisation techniques
  - lightweight logics with tractable classification such as $\mathcal{EL}$
- limitation: similarities between versions of ontologies are not taken into account (reasoning is repeated from scratch)

Example: Incremental Classification

Original ontology $\mathcal{O}_1$

$$\text{Cystic\_Fibrosis} \equiv \text{Fibrosis} \sqcap \exists \text{located\_In.Pancreas} \quad (1)$$

$$\text{Genetic\_Fibrosis} \equiv \text{Fibrosis} \sqcap \text{has\_Origin.Genetic\_Origin} \quad (2)$$

$$\text{Pancreatic\_Fibrosis} \equiv \text{Fibrosis} \sqcap \text{Pancreatic\_Disorder} \quad (3)$$

$$\text{Genetic\_Fibrosis} \sqsubseteq \text{Genetic\_Disorder} \quad (4)$$

$$\text{Pancreatic\_Disorder} \equiv \text{Disorder} \sqcap \exists \text{located\_In.Pancreas} \quad (5)$$

- 9 concept names $+ \top + \bot$
- taxonomy: 121 subsumptions

Example: Incremental Classification

Updated ontology $\mathcal{O}_2 = (\mathcal{O}_1 \setminus \mathcal{O}^-) \cup \mathcal{O}^+$

$$\begin{aligned}
\text{Cystic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \exists\text{located_In.Pancreas} && (1)\\
&\quad \sqcap \exists\text{has_Origin.Genetic_Origin} \\
\text{Genetic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{has_Origin.Genetic_Origin} && (2)\\
\text{Pancreatic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{Pancreatic_Disorder} && (3)\\
\text{Genetic_Fibrosis} &\sqsubseteq \text{Genetic_Disorder} && (4)\\
\text{Pancreatic_Disorder} &\equiv \text{Disorder} \sqcap \exists\text{located_In.Pancreas} && (5)
\end{aligned}$$

- $\mathcal{O}^- = \{(1)\}$
- $\mathcal{O}^+ = \{(1) \sqcap \exists\text{has\_Origin.Genetic\_Origin}\}$

Example: Incremental Classification

Updated ontology $\mathcal{O}_2 = (\mathcal{O}_1 \setminus \mathcal{O}^-) \cup \mathcal{O}^+$

$$
\begin{aligned}
\text{Cystic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \exists \text{located\_In.Pancreas} & (1)\\
&\quad \sqcap \exists \text{has\_Origin.Genetic\_Origin} & \\
\text{Genetic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{has\_Origin.Genetic\_Origin} & (2)\\
\text{Pancreatic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{Pancreatic\_Disorder} & (3)\\
\text{Genetic\_Fibrosis} &\sqsubseteq \text{Genetic\_Disorder} & (4)\\
\text{Pancreatic\_Disorder} &\equiv \text{Disorder} \sqcap \exists \text{located\_In.Pancreas} & (5)
\end{aligned}
$$

- Which subsumptions have changed?

Incremental Classification

Idea

- small changes in ontologies affect relatively few subsumptions
- avoid recomputing unaffected subsumptions
- identify subsumptions affected by change using modules
- [Cuenca Grau et al., JAR 2010]

Incremental Classification using Modules

Definition

$\mathcal{M} \subseteq \mathcal{O}$ is **a module for axiom α in $\mathcal{O}$** if:

$\mathcal{M} \models \alpha$ iff $\mathcal{O} \models \alpha$.

- locality-based modules for sig(α) of $\mathcal{O}$ have this property

Proposition

Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies and α an axiom. Let $\mathcal{M}^i_\alpha$ be a module for α in $\mathcal{O}_i$, for $i = 1, 2$.

- if $\mathcal{O}_1 \models \alpha$ and $\mathcal{M}^1_\alpha \subseteq \mathcal{O}_2$, then $\mathcal{O}_2 \models \alpha$
- if $\mathcal{O}_1 \not\models \alpha$ and $\mathcal{M}^2_\alpha \subseteq \mathcal{O}_1$, then $\mathcal{O}_2 \not\models \alpha$

Algorithm: Incremental Classification using Modules

Algorithm: Step 1

Input: $\mathcal{O}_1$, $\sqsubseteq_1$, $\mathcal{M}^1_A|_{A \in \mathsf{sig}(\mathcal{O}_1)}$, $\mathcal{O}^-$, $\mathcal{O}^+$

Output: $\mathcal{O}_2$, $\sqsubseteq_2$, $\mathcal{M}^2_A|_{A \in \mathsf{sig}(\mathcal{O}_2)}$

$\mathcal{O}_2 := (\mathcal{O}_1 \setminus \mathcal{O}^-) \cup \mathcal{O}^+$

For every $A \in \mathsf{sig}(\mathcal{O}_2) \setminus \mathsf{sig}(\mathcal{O}_1)$

$\quad \mathcal{M}^1_A := \mathcal{M}^1_\top$

$\quad$ for every $(\top, B) \in \sqsubseteq_1$: $\sqsubseteq_1 := \sqsubseteq_1 \cup \{(A, B)\}$

$\quad$ for every $(B, \bot) \in \sqsubseteq_1$: $\sqsubseteq_1 := \sqsubseteq_1 \cup \{(B, A)\}$

Update ontology

Set a module and update classification $\sqsubseteq_1$ for the new concept names

Algorithm: Incremental Classification using Modules

Algorithm: Step 2

$N^- := \emptyset$, $N^+ := \emptyset$

For every $A \in \text{sig}(\mathcal{O}_2)$

For every $\alpha \in \mathcal{O}^-$

if α **not** sig($\mathcal{M}^1_A$)**-local**, then $N^- := N^- \cup \{A\}$

For every $\alpha \in \mathcal{O}^+$

if α **not** sig($\mathcal{M}^1_A$)**-local**, then $N^+ := N^+ \cup \{A\}$

Determine concept names whose modules are affected by the change (using locality check for axioms)

Algorithm: Incremental Classification using Modules

Algorithm: Step 3

For every $A \in \mathrm{sig}(\mathcal{O}_2)$

If $A \in N^- \cup N^+$, then $\mathcal{M}_A^2 := \mathrm{extract\_module}(\{A\}, \mathcal{O}_2)$;

else $\mathcal{M}_A^2 := \mathcal{M}_A^1$

Determine the module for every concept name (recomputing affected modules)

Algorithm: Incremental Classification using Modules

Algorithm: Step 4

For every $A \in \mathsf{sig}(\mathcal{O}_2)$
  For every $B \in \mathsf{sig}(\mathcal{O}_2) \cup \{\bot\}$
    if $(A \in N^-$ and $A \sqsubseteq_1 B)$ or
      $(A \in N^+$ and $A \not\sqsubseteq_1 B)$
    then $\sqsubseteq_2 := \sqsubseteq_2 \cup \{(A, B)\}$ if $\mathcal{M}^2_A \models A \sqsubseteq B$
    else $\sqsubseteq_2 := \sqsubseteq_2 \cup \{(A, B)\}$ if $(A, B) \in \sqsubseteq_1$

Determine new classification $\sqsubseteq_2$ (possibly using a reasoner – any reasoner)

Example: Incremental Classification

Updated ontology $\mathcal{O}_2 = (\mathcal{O}_1 \setminus \mathcal{O}^-) \cup \mathcal{O}^+$

$$\begin{aligned}
\text{Cystic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \exists \text{located_In.Pancreas} \\
&\quad \sqcap \ \exists \text{has_Origin.Genetic_Origin} && (6) \\
\text{Genetic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{has_Origin.Genetic_Origin} && (7) \\
\text{Pancreatic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{Pancreatic_Disorder} && (8) \\
\text{Genetic_Fibrosis} &\sqsubseteq \text{Genetic_Disorder} && (9) \\
\text{Pancreatic_Disorder} &\equiv \text{Disorder} \sqcap \exists \text{located_In.Pancreas} && (10)
\end{aligned}$$

- Algorithm Step 1:
  - skip since no new symbols were added

Example: Incremental Classification

Updated ontology $\mathcal{O}_2 = (\mathcal{O}_1 \setminus \mathcal{O}^-) \cup \mathcal{O}^+$

$$\begin{aligned}
\text{Cystic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \exists\text{located_In.Pancreas} && (6)\\
&\sqcap\ \exists\text{has_Origin.Genetic_Origin} &&\\
\text{Genetic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{has_Origin.Genetic_Origin} && (7)\\
\text{Pancreatic_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{Pancreatic_Disorder} && (8)\\
\text{Genetic_Fibrosis} &\sqsubseteq \text{Genetic_Disorder} && (9)\\
\text{Pancreatic_Disorder} &\equiv \text{Disorder} \sqcap \exists\text{located_In.Pancreas} && (10)
\end{aligned}$$

- Algorithm Step 2:
  - $N^- = \{\text{Cystic\_Fibrosis}, \text{Pancreatic\_Fibrosis}\}$
  - $N^+ = \{\text{Cystic\_Fibrosis}\}$

Example: Incremental Classification

Updated ontology $\mathcal{O}_2 = (\mathcal{O}_1 \setminus \mathcal{O}^-) \cup \mathcal{O}^+$

$$\begin{aligned} \text{Cystic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \exists \text{located\_In.Pancreas} \\ &\sqcap \ \exists \text{has\_Origin.Genetic\_Origin} && (6) \\ \text{Genetic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{has\_Origin.Genetic\_Origin} && (7) \\ \text{Pancreatic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{Pancreatic\_Disorder} && (8) \\ \text{Genetic\_Fibrosis} &\sqsubseteq \text{Genetic\_Disorder} && (9) \\ \text{Pancreatic\_Disorder} &\equiv \text{Disorder} \sqcap \exists \text{located\_In.Pancreas} && (10) \end{aligned}$$

- Algorithm Step 3:
  - compute modules for Cystic_Fibrosis and Pancreatic_Fibrosis

Example: Incremental Classification

Updated ontology $\mathcal{O}_2 = (\mathcal{O}_1 \setminus \mathcal{O}^-) \cup \mathcal{O}^+$

$$
\begin{aligned}
\text{Cystic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \exists \text{located\_In.Pancreas} && (6)\\
&\sqcap \ \exists \text{has\_Origin.Genetic\_Origin} && \\
\text{Genetic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{has\_Origin.Genetic\_Origin} && (7)\\
\text{Pancreatic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \text{Pancreatic\_Disorder} && (8)\\
\text{Genetic\_Fibrosis} &\sqsubseteq \text{Genetic\_Disorder} && (9)\\
\text{Pancreatic\_Disorder} &\equiv \text{Disorder} \sqcap \exists \text{located\_In.Pancreas} && (10)
\end{aligned}
$$

- Algorithm Step 4:
  - compute whether $\mathcal{O}_2 \models \text{Pancreatic\_Fibrosis} \sqsubseteq B$
  - compute whether $\mathcal{O}_2 \models \text{Cystic\_Fibrosis} \sqsubseteq B$
  - perform only about 13 subsumption tests

And now . . .

1. Incremental Classification
2. Logical Difference for Ontology Versioning
3. Forgetting and Uniform Interpolation
4. Conclusion

Logical Difference: Motivation

Task

- given two versions $\mathcal{T}_1$ and $\mathcal{T}_2$ of an ontology and a signature Σ, compute "the difference" between $\mathcal{T}_1$ and $\mathcal{T}_2$ observable in Σ in a query language $\mathcal{QL}$.

Syntactical difference

- Many tools compute the syntactical difference between versions of texts and program code.
- But many syntactic differences do not affect the semantics of ontologies!
- Example:
  - $\mathcal{T}_1 = \{A \sqsubseteq B_1 \sqcap B_2\}$, $\mathcal{T}_2 = \{A \sqsubseteq B_1, A \sqsubseteq B_2\}$
    $\Sigma = \{A, B_1, B_2\}$
  - Then $\mathcal{T}_1 \neq \mathcal{T}_2$, but $\mathcal{T}_1 \equiv_\Sigma^{SO} \mathcal{T}_2$.

Logical Difference: Motivation

Structural difference

- extends syntactic diff by taking into account structural meta-information of distinct versions of ontologies
- regards ontologies as structured objects (e.g., taxonomy, set of RDF triplets, set of axioms)
- changes are structural operations (e.g., adding/deleting/extending/renaming classes)
- but:
  - syntax dependent and no formal semantics
  - tailored to applications of ontologies based on taxonomy
  - ontology based data access not captured

Logical Difference

$\mathcal{T}_1$ and $\mathcal{T}_2$ ontologies, $\mathcal{QL}$ a query language, Σ a signature.
The logical difference between $\mathcal{T}_1$ and $\mathcal{T}_2$ wrt. $(\mathcal{QL},\Sigma)$ is defined as

$$\mathsf{Diff}^{\mathcal{QL}}_{\Sigma}(\mathcal{T}_1,\mathcal{T}_2) \cup \mathsf{Diff}^{\mathcal{QL}}_{\Sigma}(\mathcal{T}_2,\mathcal{T}_1),$$

where

- $\mathsf{Diff}^{\mathcal{QL}}_{\Sigma}(\mathcal{T}_1,\mathcal{T}_2) = \{\varphi \in \mathcal{QL} \mid \mathcal{T}_1 \models \varphi, \mathcal{T}_2 \not\models \varphi, \mathsf{sig}(\varphi) \in \Sigma\}$.
- $\mathsf{Diff}^{\mathcal{QL}}_{\Sigma}(\mathcal{T}_2,\mathcal{T}_1) = \{\varphi \in \mathcal{QL} \mid \mathcal{T}_2 \models \varphi, \mathcal{T}_1 \not\models \varphi, \mathsf{sig}(\varphi) \in \Sigma\}$.

Observation: $\mathsf{Diff}^{\mathcal{QL}}_{\Sigma}(\mathcal{T}_1,\mathcal{T}_2) \cup \mathsf{Diff}^{\mathcal{QL}}_{\Sigma}(\mathcal{T}_2,\mathcal{T}_1) = \emptyset$ iff $\mathcal{T}_1 \equiv^{\mathcal{QL}}_{\Sigma} \mathcal{T}_2$.

Problem: How to present $\mathsf{Diff}^{\mathcal{QL}}_{\Sigma}(\mathcal{T}_1,\mathcal{T}_2)$ if it is non-empty?

Σ-difference for $\mathcal{EL}$-terminologies

Take query language $\mathcal{QL}_{\mathcal{EL}}$ consisting of $C \sqsubseteq D$, where C, D are $\mathcal{EL}$-concepts. We also denote $\mathcal{QL}_{\mathcal{EL}}$ simply as $\mathcal{EL}$.
Set

$$\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \mathsf{Diff}_\Sigma^{\mathcal{EL}}(\mathcal{T}_1, \mathcal{T}_2).$$

Example of 'large' smallest elements in $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$:

- $\mathcal{T}_2 = \emptyset$;
- $\mathcal{T}_1 = \{A' \sqsubseteq B_0, A \equiv B_n\} \cup \{B_{i+1} \equiv \exists r.B_i \sqcap \exists s.B_i \mid i < n\}$;
- $\Sigma = \{A', A, r, s\}$.

For the minimal $C \sqsubseteq A \in \mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ we have $|C| = 2^n$.

Σ-difference for $\mathcal{EL}$-terminologies

Theorem ("Primitive Witnesses Theorem")

If $(C \sqsubseteq D) \in \mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ then either

- $(A \sqsubseteq D_0) \in \mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ or
- $(C_0 \sqsubseteq A) \in \mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$,

where A is a concept name and
A, C_0 — subconcepts of C;
D_0, A — subconcepts of D, resp.

In propositional $\mathcal{EL}$: if $C \sqsubseteq A_1 \sqcap A_2 \in \mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, then

- $C \sqsubseteq A_1 \in \mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ or
- $C \sqsubseteq A_2 \in \mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$.

Compact representation of $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$

Let

- $\mathsf{diffL}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) =$
$$\left\{ A \in \Sigma \;\middle|\; \begin{array}{l} \text{there is a } \Sigma\text{-concept } C \text{ in } \mathcal{EL} \text{ s.t.} \\ \mathcal{T}_1 \models A \sqsubseteq C \text{ and } \mathcal{T}_2 \not\models A \sqsubseteq C \end{array} \right\}$$
- $\mathsf{diffR}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) =$
$$\left\{ A \in \Sigma \;\middle|\; \begin{array}{l} \text{there is a } \Sigma\text{-concept } C \text{ in } \mathcal{EL} \text{ s.t.} \\ \mathcal{T}_1 \models C \sqsubseteq A \text{ and } \mathcal{T}_2 \not\models C \sqsubseteq A \end{array} \right\}$$

$\mathsf{diffL}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ and $\mathsf{diffR}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ provide a list of concept names in Σ about which $\mathcal{T}_1$ "says more" than $\mathcal{T}_2$.

Σ-difference between $\mathcal{EL}$-terminologies

Theorem

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{EL}$-terminologies and Σ a signature. Then

- $\mathsf{diffL}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ and
- $\mathsf{diffR}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$

can be computed in polynomial time. In particular, Σ-inseparability wrt. $\mathcal{EL}$ is tractable.

Tools

CEX

- implementation of tractable algorithm computing $\mathrm{DiffL}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ and $\mathrm{DiffR}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ for acyclic $\mathcal{EL}$-terminologies [Konev, Walther, Wolter, 2008]
- `http://www.csc.liv.ac.uk/~konev/software/`

OWLDiff

- CEX-diff for $\mathcal{EL}$-terminologies [Kremen, Smid, Kouba, 2011, to appear]
- plugins for Protégé and NeON toolkit
- `http://krizik.felk.cvut.cz/km/owldiff`

Tools

CEX2

- extends CEX to $\mathcal{ELH}^r$ (i.e. $\mathcal{EL}$ with role inclusion axioms and domain and range restrictions) without loosing tractability [Konev, Ludwig, Walther, Wolter, 2012]
- `http://www.csc.liv.ac.uk/~michel/software/cex2/`

LogDiffViz

- Protégé plugin that calls CEX2 and visualises ontology versions and the differences as a hierarchical structure
- `http://www.csc.liv.ac.uk/~cs8wg/LogDiffViz/`

CEX applied to SNOMED CT

Task: Compute the logical difference of two versions of SNOMED CT

- two versions:
  - SNOMED CT 2005 (SM-05):
    - 379 691 axioms
    - 09 February 2005
  - SNOMED CT 2006 (SM-06):
    - 389 472 axioms
    - 30 December 2006
- $\Sigma \subseteq \mathrm{sig}(\text{SM-05}) \cap \mathrm{sig}(\text{SM-06})$ randomly selected
- compute average (of time/memory/diff-size) over 20 samples for every signature size
- hardware: Intel Core 2 CPU at 2.13 GHz and 3 GB of RAM

SM-05 vs SM-06

Size of Σ	CEX: diff(SM-05,SM-06)			
	Time (Sec.)	Memory (MByte)	$\lvert diffL_\Sigma \rvert$	$\lvert diffR_\Sigma \rvert$
100	513.1	1 393.7	0.10	0.10
1 000	512.4	1 394.6	2.35	2.15
10 000	517.7	1 424.3	155.35	125.35
100 000	559.8	1 473.2	11 795.90	4 108.6

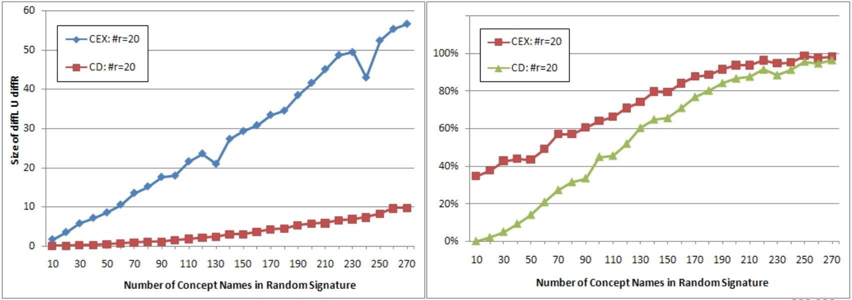
- Note: role box ignored

Comparison on the Joint Signature

- diff(SM-05,SM-06) on
  $\Sigma = \text{sig}(\text{SM-05}) \cap \text{sig}(\text{SM-06})$
  - 689 seconds
  - $|\text{diffL}_\Sigma| + |\text{diffR}_\Sigma| = 162010$
  - Class hierarchy comparison misses 32475 of them

Comparing with classification

- Combined $\mathrm{diffL}_\Sigma(\emptyset, M)$ and $\mathrm{diffR}_\Sigma(\emptyset, M)$
  - M is a subset of SM-05 containing $\sim 140,000$ axioms
  - Σ — randomly selected from M (incl. 20 role names)
  - avg. over 500 samples for each signature size
- Difference in class hierarchy



CEX on MEX

Instead of computing $\text{diffL}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) \cup \text{diffR}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ directly,

- first extract minimal Σ-modules $\mathcal{T}_1'$ and $\mathcal{T}_2'$ from $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively,
- then compute $\text{diffL}_\Sigma(\mathcal{T}_1', \mathcal{T}_2') \cup \text{diffR}_\Sigma(\mathcal{T}_1', \mathcal{T}_2')$.

Size of Σ	CEX: diff(SM-05,SM-06)				CEX: diff(Mod'05,Mod'06)	
	Time (Sec.)	Memory (MByte)	$\|\text{diffL}_\Sigma\|$	$\|\text{diffR}_\Sigma\|$	Time (Sec.)	Memory (MByte)
100	513.1	1 393.7	0.0	0.0	3.66	116.5
1 000	512.4	1 394.6	2.5	2.5	4.46	122.5
10 000	517.7	1 424.3	183.2	122.0	22.29	126.5
100 000	559.8	1 473.2	11 322.1	4 108.5	189.98	615.8
379741	790.0	1999.3	191714	684.1	1850.7	237044

And now . . .

Forgetting Vocabulary: Motivation

Forgetting vocabulary is eliminating that vocabulary from the ontology (involving a reformulation of the ontology).

Use-cases

- re-use: instead of whole ontology, use a potentially much smaller ontology resulting from forgetting
- predicate hiding: concealing confidential information in ontologies
- ontology summary: succinct presentation of what ontology states about non-forgotten vocabulary

The dual notion of forgetting is uniform interpolation.

Uniform Interpolation

Let $\mathcal{T}$ be a $\mathcal{EL}$-TBox and Σ a signature. A TBox $\mathcal{T}'$ is called a uniform interpolant of $\mathcal{T}$ wrt. Σ if the following holds:

- $\mathsf{sig}(\mathcal{T}') \subseteq \Sigma$;
- $\mathcal{T} \equiv_{\Sigma}^{\mathcal{EL}} \mathcal{T}'$.

Theorem

Let $\mathcal{T}_1', \mathcal{T}_2'$ be uniform interpolants of $\mathcal{T}_1$ and $\mathcal{T}_2$ wrt. Σ. The following conditions are equivalent:

- $\mathcal{T}_1 \equiv_{\Sigma}^{\mathcal{EL}} \mathcal{T}_2$;
- $\mathcal{T}_1'$ and $\mathcal{T}_2'$ are logically equivalent.

$\mathcal{EL}$-terminologies

Theorem

There exist an $\mathcal{EL}$-terminology $\mathcal{T}$ and Σ such that there does not exist an uniform interpolant of $\mathcal{T}$ wrt. Σ.

Proof. Let

$$\mathcal{T} = \{A \sqsubseteq B, B \sqsubseteq \exists r.B\}, \quad \Sigma = \{A, r\}.$$

An infinite axiomatisation of the uniform interpolant is given by

$$\{A \sqsubseteq \underbrace{\exists r. \cdots \exists r}_{n}.\top \mid n \geq 1\}.$$

A finite $\mathcal{T}_\Sigma$ does not exist (even in first-order logic).

Acyclic $\mathcal{EL}$-terminologies

Theorem

For acyclic $\mathcal{EL}$-terminologies, uniform interpolants always exist. In the worst case, exponentially many axioms are required.

Proof of second part. Let

$$\mathcal{T} = \{A \equiv B_1 \sqcap \cdots \sqcap B_n\} \cup \{A_{ij} \sqsubseteq B_i \mid 1 \leq i, j \leq n\}.$$

and

$$\Sigma = \{A\} \cup \{A_{ij} \mid 1 \leq i, j \leq n\}.$$

Then

$$\mathcal{T}_\Sigma = \{A_{1j_1} \sqcap \cdots \sqcap A_{n,j_n} \sqsubseteq A \mid 1 \leq j_1, \ldots, j_n \leq n\}$$

is a minimal uniform interpolant. Note that $|\mathcal{T}_\Sigma| = n^n$.

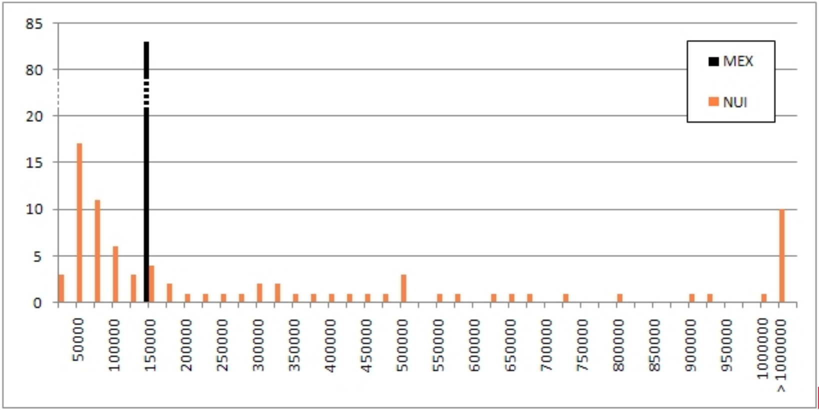
Computing uniform interpolants for SNOMED CT and NCI

- NUI: prototype implementation computing uniform interpolants for acyclic $\mathcal{EL}$-terminologies.
- Σ — randomly selected from sig(SNOMED CT) and sig(NCI), respectively.
- table shows success rate of NUI

$\lvert\Sigma\rvert$	SNOMED CT	$\lvert\Sigma\rvert$	NCI
2 000	100.0%	5 000	97.0%
3 000	92.2%	10 000	81.1%
4 000	67.0%	15 000	72.0%
5 000	60.0%	20 000	59.2%

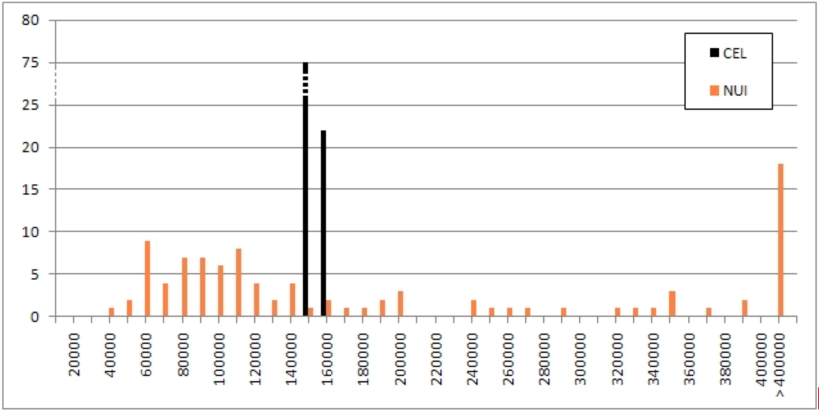
Comparing the size of MEX-modules and Σ-interpolants

- Size distribution of MEX-modules and instance Σ-interpolants of SNOMED CT wrt. signatures containing 3 000 concept names and 20 role names



Comparing the size of ⊤-local modules and Σ-interpolants

- Size distribution of CEL-modules and instance Σ-interpolants of NCI wrt. signatures containing 7 000 concept names and 20 role names



Uniform interpolants beyond $\mathcal{EL}$

Theorem

For $\mathcal{ALC}$-TBoxes, uniform interpolants expressed in FOL do not always exist. [Ghilardi, Lutz, Wolter, 2006]

Theorem

For $\mathcal{ALC}$-TBoxes, deciding the existence of uniform interpolants in ALC is 2ExpTime-complete. If they exist, uniform interpolants are most triple exponential in the size of the original TBox. [Lutz, Wolter, 2011]

And now . . .

Conclusion

What are modules good for (so far)?

- import/reuse of ontologies (locality-based/MEX modules)
- towards understanding the structure of an ontology (atomic decomposition)
- incremental reasoning

Related notions

- Σ-inseparability (foundation of modules)
- logical difference (ontology versioning)
- forgetting (hiding of symbols)

Outlook

Some open problems

- finding appropriate signature for a module (shopping for symbols)
- methodology for collaborative ontology development using modules
- ontology comprehension/visualisation (e.g. using atomic decomposition)
- modular reasoning (improve performance using modules)

So.. that's it.

Thank you for coming!

tschneider(at)informatik.uni-bremen.de
dirk.walther(at)tu-dresden.de

Reminder: Workshop on Modular Ontologies (WoMO)
7th International Workshop on Modular Ontologies
September 15, Corunna, Spain
http://www.iaoa.org/womo/2013.html