

# Properties of Conservative Extensions and Modules

C. Lutz, U. Sattler, and F. Wolter

July 31, 2008

# Today: Modularity for Lightweight Description Logics

Currently, two families of lightweight DLs are investigated and applied:

- DL-Lite: for conceptual modeling, data integration, querying instance data using background theories.
- $\mathcal{EL}$ : many large medical and biological ontologies are given in  $\mathcal{EL}$ .

We consider  $\mathcal{EL}$  (but modularity in DL-Lite has been investigated as well).

$\mathcal{EL}$  is a fragment of  $\mathcal{ALC}$ . Concept language of  $\mathcal{EL}$ :

$$C = A \mid \top \mid C \sqcap D \mid \exists r.C$$

TBox  $T$  is a finite set of concept inclusions  $C \sqsubseteq D$ .

Reasoning services:

- Satisfiability: every  $\mathcal{EL}$ -concept  $C$  is satisfiable w.r.t. any  $\mathcal{EL}$ -TBox.
- Subsumption: given  $C, D, T$ , does  $T \models C \sqsubseteq D$ ? This problem is decidable in polynomial time.

## Modularity reasoning for $\mathcal{EL}$

- Deciding whether two  $\mathcal{EL}$ -TBoxes are  $\Sigma$ -inseparable w.r.t.  $\mathcal{EL}$  is ExpTime-complete.
- For  $\mathcal{EL}$ -TBoxes,  $\Sigma$ -inseparability w.r.t. SO is undecidable.
- For  $\mathcal{EL}$ -TBoxes, even  $T \equiv_{\Sigma}^{SO} \emptyset$ , (equivalently, whether

$$\{M_{|\Sigma} \mid M \models T\} = \text{class of all } \Sigma\text{-models}$$

is undecidable.

- $\mathcal{EL}$  has interpolation, but  $(\mathcal{EL}, \mathcal{EL})$  is not robust under replacement.

Today, we consider  $\mathcal{EL}$ -TBoxes of a particular form.

## Definition

An  $\mathcal{EL}$ -TBox  $T$  is a  $\mathcal{EL}$ -terminology if

- every axiom is of the form  $A \equiv C$ , where  $A$  is a concept name;
- no concept name  $A$  occurs more than once on the left hand side of an axiom.

A  $\mathcal{EL}$ -terminology  $T$  is **acyclic** if no concept name refers to itself along definitions:

- let  $A \prec_T X$  if there exists an axiom  $A \equiv C$  in  $T$  such that  $X$  occurs in  $C$ .

Then  $T$  is acyclic iff  $\prec_T$  is acyclic (equivalently  $\prec_T^+$  is irreflexive).

In a TBox  $T$ , we write  $A \sqsubseteq C$  instead of  $A \equiv X \sqcap C$ , if this is the only occurrence of  $X$  (and  $X$  not in any relevant signature  $\Sigma$ ).

## Plan for $\mathcal{EL}$ -terminologies

- Deciding ' $T \equiv_{\Sigma}^{SO} \emptyset$ ' in polynomial time (Uli: then  $T$  is safe!).
- Extracting modules.
- Logical difference: comparing versions of ontologies.
- Uniform interpolation (forgetting).

## Theorem

*The following problem can be solved in polynomial time: given an acyclic  $\mathcal{EL}$ -terminology  $T$ , decide whether*

$$T \equiv_{\Sigma}^{SO} \emptyset.$$

For the proof, we distinguish two types of dependencies between  $\Sigma$ -symbols:

- in  $T$ , the 'definition' of a  $\Sigma$ -symbol uses another  $\Sigma$ -symbols;
- in  $T$ , two  $\Sigma$ -symbols are 'defined' using common non- $\Sigma$ -symbols.

# Syntactic $\Sigma$ -dependencies

Let  $T$  be an acyclic  $\mathcal{EL}$ -terminology.

- $T$  contains a **syntactic  $\Sigma$ -dependency** if there exist  $A, X \in \Sigma$  such that  $A \prec_T^+ X$ .

## Theorem

*If an acyclic  $\mathcal{EL}$ -terminology  $T$  contains a syntactic  $\Sigma$ -dependency, then  $T \not\equiv_{\Sigma}^{SO} \emptyset$ .*

Proof. Suppose  $T$  contains a syntactic  $\Sigma$ -dependency  $A \prec_{\Sigma}^+ X$ . Take a  $\Sigma$ -model  $M$  with  $A^M = \Delta^M$  and  $X^M = \emptyset$ . Then  $M$  can't be expanded to a model of  $T$ .

## Decomposing an acyclic $\mathcal{EL}$ -terminology

Consider an acyclic  $\mathcal{EL}$ -terminology  $T$ . Take partition

$$T = T_\Sigma \cup T',$$

where

$$T_\Sigma = \{A \equiv C \mid A \in \Sigma \text{ or } \exists B \in \Sigma \ B \prec_T^+ A\}$$

### Theorem

*If  $M \models T_\Sigma$ , then there exists  $M'$  with  $M'_\Sigma = M_\Sigma$  such that  $M' \models T$ .*

Proof. Expand  $M$  inductively by setting  $A^{M'} := C^{M'}$  for  $A \equiv C \in T'$ .

## Checking indirect $\Sigma$ -dependencies

### Theorem

Let  $T$  be an acyclic  $\mathcal{EL}$ -terminology without syntactic  $\Sigma$ -dependencies. Then the following conditions are equivalent:

- $T \equiv_{\Sigma}^{SO} \emptyset$ ;
- Every one point  $\Sigma$ -model can be expanded to a model of  $T_{\Sigma}$ .

Point 2 implies Point 1. Let  $M$  be a  $\Sigma$ -model. As  $T_{\Sigma}$  contains no  $\Sigma$ -roles, we may assume that  $\Sigma$  contains no roles. For each  $d \in M$ , let  $M'_{\{d\}} \models T_{\Sigma}$  be an expansion of  $M_{\{d\}}$ . Then

$$M' = \bigcup_{d \in M} M'_{\{d\}} \models T_{\Sigma}$$

and  $M'$  is an expansion of  $M$ .

## Polytime algorithm for $T \equiv_{\Sigma}^{SO} \emptyset$

To decide whether  $T \equiv_{\Sigma}^{SO} \emptyset$ , check

- $T$  contains no syntactic  $\Sigma$ -dependencies;
- every one point  $\Sigma$ -model can be expanded to a model of  $T_{\Sigma}$ .

Point 2 holds iff

For all  $A \in \Sigma$ ,

$$\{X \mid A \prec_T^+ X\} \not\subseteq \{X \mid \exists B \in \Sigma \setminus \{A\} \ B \prec_T^+ X\}.$$

Observation: For acyclic  $\mathcal{ALC}$ -terminologies without syntactic  $\Sigma$ -dependencies, one can decide  $T \equiv_{\Sigma}^{SO} \emptyset$  by considering one point-models (then  $\Pi_2^P$ -complete).

# Module extraction

Given  $T$  and  $\Sigma$ , we use this decision procedure to extract from  $T$  the smallest  $M \subseteq T$  such that

$$T \setminus M \equiv_{\Sigma \cup \text{sig}(M)}^{SO} \emptyset.$$

Equivalently,

$$M \equiv_{\Sigma \cup \text{sig}(M)}^{SO} T.$$

Uli: then  $T \setminus M$  is safe for  $\Sigma \cup \text{sig}(M)$ .

# Module extraction algorithm

Input: acyclic  $\mathcal{EL}$ -terminology  $T$  and signature  $\Sigma$ .

Output: Smallest  $M \subseteq T$  such that  $T \setminus M \equiv_{\Sigma \cup \text{sig}(M)} \emptyset$ .

Initialise:  $M = \emptyset$ ,  $\Sigma' = \Sigma$ . Apply Rules 1 and 2 exhaustively, preferring Rule 1.

- if  $A \in \Sigma'$ ,  $A \equiv C \in T \setminus M$ , and exists  $X \in \Sigma'$  with  $A \prec_{T \setminus M}^+ X$ ,

$$M := M \cup \{A \equiv C\}, \quad \Sigma' := \Sigma' \cup \text{sig}(C).$$

- if  $A \in \Sigma'$ ,  $A \equiv C \in T \setminus M$ , and

$$\{X \mid A \prec_{T \setminus M}^+ X\} \subseteq \{X \mid \exists B \in \Sigma' \setminus \{A\} \ B \prec_{T \setminus M}^+ X\},$$

then set

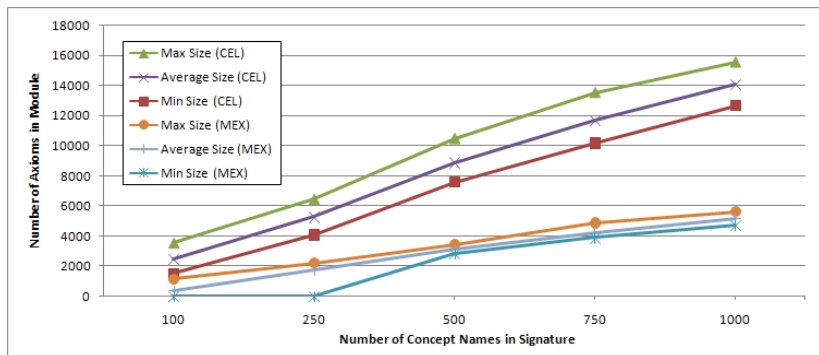
$$M := M \cup \{A \equiv C\}, \quad \Sigma' := \Sigma' \cup \text{sig}(C).$$

## SNOMED CT:

- Systematised Nomenclature of Medicine (Clinical Terms).
- $\sim 400,000$  terms
- used in health care etc. in the US, UK, Australia etc.
- an acyclic  $\mathcal{EL}$ -terminology (+ role box):

# Experiment: Extraction of modules from SNOMED CT

- We use implementation MEX of the algorithm above.
- $\Sigma$  — randomly selected from **SNOMED CT**.
- 1000 samples for each signature size
- **with** role box (under simplifying assumptions)



## Logical Difference: motivation

Problem: given two **versions**  $T_1$  and  $T_2$  of an ontology and a signature  $\Sigma$ , compute “the difference” between  $T_1$  and  $T_2$  observable in  $\Sigma$  in a query language  $QL$ .

Lot's of tools to compute the difference between versions of texts and program code! But this is not of much help for ontologies/logical theories:

- $T_1 = \{A \sqsubseteq B_1 \sqcap B_2\}$
- $T_2 = \{A \sqsubseteq B_1, A \sqsubseteq B_2\}$ .
- $\Sigma = \{A, B_1, B_2\}$ .

Then  $T_1 \neq T_2$ , but  $T_1$  and  $T_2$  are equivalent (so  $\Sigma$ -inseparable w.r.t. SO).

# Logical Difference

$T_1$  and  $T_2$  ontologies,  $\mathcal{QL}$  a query language,  $\Sigma$  a signature.  
The **logical  $\Sigma$ -difference between  $T_1$  and  $T_2$  w.r.t.  $\mathcal{QL}$**  is defined as

$$\text{Diff}_{\Sigma}^{\mathcal{QL}}(T_1, T_2) \cup \text{Diff}_{\Sigma}^{\mathcal{QL}}(T_2, T_1),$$

where

- $\text{Diff}_{\Sigma}^{\mathcal{QL}}(T_1, T_2) = \{\varphi \in \mathcal{QL}_{\Sigma} \mid T_1 \models \varphi, T_2 \not\models \varphi\}$ .
- $\text{Diff}_{\Sigma}^{\mathcal{QL}}(T_2, T_1) = \{\varphi \in \mathcal{QL}_{\Sigma} \mid T_2 \models \varphi, T_1 \not\models \varphi\}$ .

**Observation:**  $\text{Diff}_{\Sigma}^{\mathcal{QL}}(T_1, T_2) \cup \text{Diff}_{\Sigma}^{\mathcal{QL}}(T_2, T_1) = \emptyset$  iff  $T_1$  and  $T_2$  are  $\Sigma$ -inseparable w.r.t.  $\mathcal{QL}$ .

**Problem:** How to present  $\text{Diff}_{\Sigma}^{\mathcal{QL}}(T_1, T_2)$  if it is non-empty?

## $\Sigma$ -difference for $\mathcal{EL}$ -terminologies

Take query language  $\mathcal{EL}$  consisting of  $C \sqsubseteq D$ , where  $C, D$  are  $\mathcal{EL}$ -concepts. We set

$$\text{Diff}_{\Sigma}(T_1, T_2) = \text{Diff}_{\Sigma}^{\mathcal{EL}}(T_1, T_2).$$

Example of 'large' smallest elements in  $\text{Diff}_{\Sigma}(T_1, T_2)$ :

- $T_2 = \emptyset$ ;
- $T_1 = \{A' \sqsubseteq B_0, A \equiv B_n\} \cup \{B_{i+1} \equiv \exists r.B_i \sqcap \exists s.B_i \mid i < n\}$ ;
- $\Sigma = \{A', A, r, s\}$ .

For the minimal  $C \sqsubseteq A \in \text{Diff}_{\Sigma}(T_1, T_2)$  we have  $|C| = 2^n$ .

### Theorem

If  $(C \sqsubseteq D) \in \text{Diff}_\Sigma(T_1, T_2)$  then either

- $(A \sqsubseteq D_0) \in \text{Diff}_\Sigma(T_1, T_2)$  or
- $(C_0 \sqsubseteq A) \in \text{Diff}_\Sigma(T_1, T_2)$ ,

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  $C \sqsubseteq A_1 \sqcap A_2 \in \text{Diff}_\Sigma(T_1, T_2)$ , then

- $C \sqsubseteq A_1 \in \text{Diff}_\Sigma(T_1, T_2)$  or
- $C \sqsubseteq A_2 \in \text{Diff}_\Sigma(T_1, T_2)$ .

# Compact representation of $\text{Diff}_\Sigma(T_1, T_2)$

Let

- $\text{diffL}_\Sigma(T_1, T_2) = \left\{ A \in \Sigma \mid \begin{array}{l} \text{there is a } \Sigma\text{-concept } C \text{ in } \mathcal{EL} \text{ s.t.} \\ T_1 \models A \sqsubseteq C \text{ and } T_2 \not\models A \sqsubseteq C \end{array} \right\}$
- $\text{diffR}_\Sigma(T_1, T_2) = \left\{ A \in \Sigma \mid \begin{array}{l} \text{there is a } \Sigma\text{-concept } C \text{ in } \mathcal{EL} \text{ s.t.} \\ T_1 \models C \sqsubseteq A \text{ and } T_2 \not\models C \sqsubseteq A \end{array} \right\}$

$\text{diffL}_\Sigma(T_1, T_2)$  and  $\text{diffR}_\Sigma(T_1, T_2)$  provide a list of concept names in  $\Sigma$  about which  $T_1$  “says more” than  $T_2$ .

### Theorem

Let  $T_1$  and  $T_2$  be  $\mathcal{EL}$ -terminologies and  $\Sigma$  a signature. Then

- $\text{diffL}_\Sigma(T_1, T_2)$  and
- $\text{diffR}_\Sigma(T_1, T_2)$

can be computed in polynomial time. In particular,  $\Sigma$ -inseparability w.r.t.  $\mathcal{EL}$  is tractable.

## Proof idea for $\text{DiffR}_\Sigma(T_1, T_2)$

Consider  $T_2$  and  $A \in \Sigma$ . Compute set

$$\text{pre}_\Sigma^{T_2}(A)$$

of **most specific  $\Sigma$ -concepts**  $C$  such that  $T_2 \not\models C \sqsubseteq A$ .

More precisely,  $\text{pre}_\Sigma^{T_2}(A)$  is a minimal set of  $\Sigma$ -concepts such that

- $T_2 \not\models C \sqsubseteq A$ , for all  $C \in \text{pre}_\Sigma^{T_2}(A)$ ;
- if  $T_2 \not\models D \sqsubseteq A$  and  $D$  a  $\Sigma$ -concept, then  $\models C \sqsubseteq D$ , for some  $C \in \text{pre}_\Sigma^{T_2}(A)$ .

Suppose  $\text{pre}_\Sigma^{T_2}(A)$  has been computed. Then the following are equivalent:

- $A \in \text{DiffR}_\Sigma(T_1, T_2)$ ;
- $T_1 \models C \sqsubseteq A$ , for some  $C \in \text{pre}_\Sigma^{T_2}(A)$ .

## Computing $\text{pre}_{\Sigma}^{T_2}(A)$ (propositional case)

Observation: In general,  $\text{pre}_{\Sigma}^{T_2}(A)$  is infinite and contains concepts of exponential size. However, polynomial representation possible using a TBox.

Consider acyclic propositional case:

$A$  is **primitive** in  $T_2$  if it has no occurrence of the form  $A \equiv D$  in  $T_2$ .

Then  $\text{pre}_{\Sigma}^{T_2}(A)$  can be defined inductively:

- For  $A$  primitive:  $\text{pre}_{\Sigma}(A) = \{\prod_{B \in \Sigma, T_2 \not\models B \sqsubseteq A} B\}$ ;
- For  $A \equiv B_1 \sqcap \dots \sqcap B_n \in T_2$ :  $\text{pre}_{\Sigma}(A) = \bigcup_{i \leq n} \text{pre}_{\Sigma}(B_i)$ .

- **CEX**: implementation of tractable algorithm computing  $\text{DiffL}_\Sigma(T_1, T_2)$  and  $\text{DiffR}_\Sigma(T_1, T_2)$  for acyclic  $\mathcal{EL}$ -terminologies.
- Two versions of SNOMED CT: SNOMED CT'05 vs SNOMED CT'06

## SNOMED CT'05 vs SNOMED CT'06

- $\Sigma$  — randomly selected from  $\text{sig}(\text{SNOMED CT}'05) \cap \text{sig}(\text{SNOMED CT}'06)$
- 20 samples for every signature size

Size of $\Sigma$	CEX: diff(SNOMED CT'05,SNOMED CT'06)			
	Time (Sec.)	Memory (MByte)	$ \text{diffL}_{\Sigma} $	$ \text{diffR}_{\Sigma} $
100	513.1	1 393.7	0.0	0.0
1 000	512.4	1 394.6	2.5	2.5
10 000	517.7	1 424.3	183.2	122.0
100 000	559.8	1 473.2	11 322.1	4 108.5

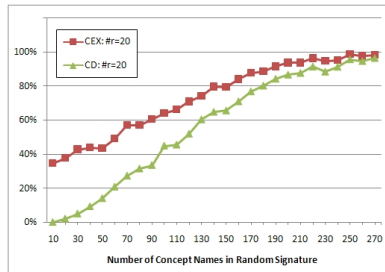
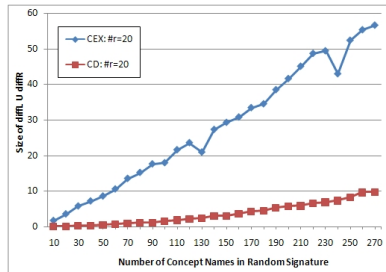
- Rolebox ignored

## Comparison on the Joint Signature

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

# Comparing with $\emptyset$

- Combined  $\text{diffL}_{\Sigma}(\emptyset, S)$  and  $\text{diffR}_{\Sigma}(\emptyset, S)$ 
  - $S$  — a part of SNOMED CT'05 containing  $\sim 140,000$  axioms
  - $\Sigma$  — randomly selected from  $M$
  - 500 samples for each signature size
- Difference in class hierarchy



# CEX on MEX

Instead of computing  $\text{diffL}_\Sigma(T_1, T_2) \cup \text{diffR}_\Sigma(T_1, T_2)$  directly,

- first extract minimal  $\Sigma$ -modules  $T'_1$  and  $T'_2$  from  $T_1$  and  $T_2$ , respectively,
- then compute  $\text{diffL}_\Sigma(T'_1, T'_2) \cup \text{diffR}_\Sigma(T'_1, T'_2)$ .

Size of $\Sigma$	CEX: diff(SNOMED CT'05,SNOMED CT'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

# Uniform Interpolation

Let  $T$  be a  $\mathcal{EL}$  TBox and  $\Sigma$  a signature. A TBox  $T'$  is called a **uniform interpolant** of  $T$  w.r.t.  $\Sigma$  if the following holds:

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

## Theorem

Let  $T'_1, T'_2$  be uniform interpolants of  $T_1$  and  $T_2$  w.r.t.  $\Sigma$ . The following conditions are equivalent:

- $T_1 \equiv_{\Sigma}^{\mathcal{EL}} T_2$ ;
- $T'_1$  and  $T'_2$  are logically equivalent.

## Theorem

*There exist an  $\mathcal{EL}$ -terminology  $T$  and  $\Sigma$  such that there does not exist an uniform interpolant of  $T$  w.r.t.  $\Sigma$ .*

Proof. Let

$$T = \{A_0 \sqsubseteq B, B \sqsubseteq A_1 \sqcap \exists r.B\}, \quad \Sigma = \{A_0, A_1, r\}.$$

A uniform interpolant  $T_\Sigma$  would have to finitely axiomatise the class of models  $M$  satisfying:

- if  $d_0 \in A_0^M$ , then there exists a sequence  $d_0 r^M d_1 r^M d_2 r^M \dots$  with  $d_i \in A_1^M$  for all  $i \geq 0$ .

Such a  $T_\Sigma$  does not exist (even in first-order logic).

## Theorem

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

Proof of second part. Let

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

and

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

Then

$$T_\Sigma = \{A_{1j_1} \sqcap \dots \sqcap A_{n,j_n} \sqsubseteq A \mid 1 \leq j_1, \dots, j_n \leq n\}$$

is a minimal uniform interpolant.