Modularity in Ontologies: Module extraction and its logical foundations (Part B)

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, 13 August 2013

And now . . .

1. Logical guarantees in detail

2. Overview of the remainder of this course

Reminder

Safety

Concerns the **usage of (imported) terms** in the importing ontology:

Let $\texttt{JRA}, \texttt{GeneticDisorder} \in \mathsf{sig}(\mathit{NCI})$.

$$\mathit{JRAO} \cup \mathit{NCI} \models \texttt{JRA} \sqsubseteq \texttt{GeneticDisorder}$$
$$\text{iff}$$
$$\mathit{NCI} \models \texttt{JRA} \sqsubseteq \texttt{GeneticDisorder}$$

Does this sound like inseparability?

We want: $\mathit{JRAO} \cup \mathit{NCI} \ \equiv_{\text{“the imported terms”}} \ \mathit{NCI}$

Reminder

Independence

Concerns **preservation of safety**:

If *JRAO* is safe for *Galen* and for *NCI*, then

- *JRAO* ∪ *NCI-module* is still safe for *Galen* and
- *JRAO* ∪ *Galen-module* is still safe for *NCI*.

Reminder

Coverage

Concerns what we would consider a **module**:

Let $\mathtt{JRA}, \mathtt{GeneticDisorder} \in \mathsf{sig}(\mathit{NCI})$.

$$\mathit{JRAO} \cup \mathit{NCI} \models \mathtt{JRA} \sqsubseteq \mathtt{GeneticDisorder}$$

iff

$$\mathit{JRAO} \cup \textbf{\textit{NCI-module}} \models \mathtt{JRA} \sqsubseteq \mathtt{GeneticDisorder}$$

Does this sound like inseparability?

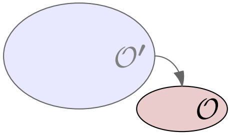
We want: $\mathit{JRAO} \cup \mathit{NCI} \equiv_{\text{“the imported terms”}} \mathit{JRAO} \cup \textbf{\textit{NCI-module}}$

Safety guarantee in detail

Safety for an ontology

$\mathcal{O}$ imports $\mathcal{O}'$ in an $\mathcal{L}$-**safe** way
(or $\mathcal{O}$ is **safe for** $\mathcal{O}'$ w.r.t. $\mathcal{L}$)

if $\mathcal{O} \cup \mathcal{O}' \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}')} \mathcal{O}'$.



Intuition: $\mathcal{O} \cup \mathcal{O}'$ doesn't change the *meaning* of $\mathcal{O}'$-terms *observable in* $\mathcal{L}$.

Problems

- Which $\mathcal{L}$ to choose?
  - for ontology design: subsumptions betw. (complex?) concepts
  - for ontology usage: my favourite query language
- We might not have control over $\mathcal{O}'$ and $\mathsf{sig}(\mathcal{O}')$

  $\mathcal{O}' = \mathit{NCI}$ might change over time, we want latest version

Solution: Safety for a signature!

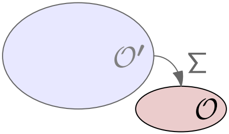
Safety for a signature

Definition

$\mathcal{O}$ is **safe for Σ w.r.t. $\mathcal{L}$** if,

for every $\mathcal{L}$-ontology $\mathcal{O}'$ with $\mathrm{sig}(\mathcal{O}) \cap \mathrm{sig}(\mathcal{O}') \subseteq \Sigma$,

$\mathcal{O} \cup \mathcal{O}' \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O}'$.



Theorem

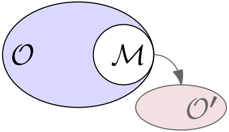
1. If $\mathcal{O}$ is a model Σ-conservative extension of $\emptyset$ $(\mathcal{O} \equiv^{\mathsf{SO}}_{\Sigma} \emptyset)$, then $\mathcal{O}$ is safe for Σ w.r.t. any $\mathcal{L} \leqslant \mathsf{SO}$.
2. Under certain assumptions: $\mathcal{O}$ is safe for Σ w.r.t. $\mathcal{L}$ iff $\mathcal{O} \equiv^{\mathcal{L}}_{\Sigma} \emptyset$.

Coverage guarantee in detail

Module for an ontology

$\mathcal{M} \subseteq \mathcal{O}$ is a **module for $\mathcal{O}'$ in $\mathcal{O}$ w.r.t. $\mathcal{L}$** if

$\mathcal{O}' \cup \mathcal{O} \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}')} \mathcal{O}' \cup \mathcal{M}$.



Intuition: $\mathcal{O}' \cup \mathcal{M}$ says as much about the $\mathcal{O}'$-terms as $\mathcal{O}' \cup \mathcal{O}$ (*observable in $\mathcal{L}$*)

Problems

- Which $\mathcal{L}$ to choose?
  - for ontology design: subsumptions betw. (complex?) concepts
  - for ontology usage: my favourite query language
- The module shouldn't depend on the importing ontology, but only on the signature we want to use.

Solution: Module for a signature!
$\rightsquigarrow$ interoperability of $\mathcal{M}$

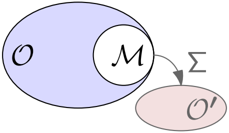
Module for a signature

Definition

$\mathcal{M} \subseteq \mathcal{O}$ is **a module for Σ in $\mathcal{O}$** w.r.t. $\mathcal{L}$ if,

for every $\mathcal{L}$-ontology $\mathcal{O}'$ with
$\mathsf{sig}(\mathcal{O}') \cap \mathsf{sig}(\mathcal{O}) \subseteq \Sigma$,

$\mathcal{O}' \cup \mathcal{O} \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}')} \mathcal{O}' \cup \mathcal{M}$.



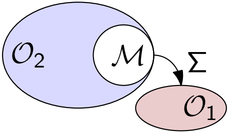
Observation

1. If $\mathcal{M} \subseteq \mathcal{O}$ and $\mathcal{O}$ is a model Σ-c.e. of $\mathcal{M}$ $(\mathcal{O} \equiv^{\mathsf{SO}}_{\Sigma} \mathcal{M})$, then $\mathcal{M}$ is a module for Σ in $\mathcal{O}$ w.r.t. any $\mathcal{L} \leqslant \mathsf{SO}$
2. Under certain assumptions:
   $\mathcal{M} \subseteq \mathcal{O}$ is a module for Σ in $\mathcal{O}$ w.r.t. $\mathcal{L}$
   iff $\mathcal{O} \setminus \mathcal{M} \equiv^{\mathcal{L}}_{\Sigma} \emptyset$.

Modules and Safety are closely related

The following is immediate from the previous definitions.

Homework: Prove.



Let $\mathcal{O}_1,\ \mathcal{M} \subseteq \mathcal{O}_2$ be ontologies in $\mathcal{L}$ and Σ a signature. Then

1. $\mathcal{O}_1$ is safe for Σ w.r.t. $\mathcal{L}$ iff $\emptyset$ is a Σ-module in $\mathcal{O}_1$ w.r.t. $\mathcal{L}$
   $\mathcal{O}_1$ constrains interpretation of terms in Σ as much as $\emptyset$
2. If $\mathcal{O}_2 \setminus \mathcal{M}$ is safe for $\Sigma \cup \mathrm{sig}(\mathcal{M})$ w.r.t. $\mathcal{L}$,
   then $\mathcal{M}$ is a Σ-module in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$
   $\mathcal{O}_2 \setminus \mathcal{M}$ doesn't constrain interpretation of terms from $\Sigma \cup \mathrm{sig}(\mathcal{M})$

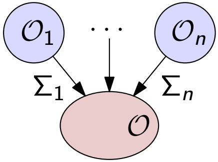
Independence Guarantee in Detail

Basic requirement for importing ontologies independently.

Independence

Safety is preserved under imports:

If $\mathcal{O}$ is safe for Σ_i ($\mathcal{O}_i$),
then $\mathcal{O} \cup \mathcal{O}_j$ is still safe for Σ_i ($\mathcal{O}_i$).



Independence is **difficult to guarantee** ...

- when the Σ_i share terms:

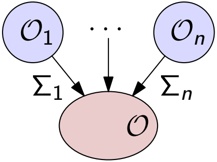
  e.g., $\mathcal{O} = \{A \sqsubseteq \top\}$ is safe for $\Sigma = \{A, B\}$,
  but $\mathcal{O} \cup \{A \sqsubseteq B\}$ is *not safe* for Σ

- when the Σ_i don't share terms:

  e.g., $\mathcal{O} = \{A \sqsubseteq B\}$ is safe for $\Sigma_2 = \{A\}$ and $\Sigma_3 = \{B\}$,
  but $\mathcal{O} \cup \{B \equiv \bot\}$ is *not safe* for Σ_2
  and $\mathcal{O} \cup \{A \equiv \top\}$ is *not safe* for Σ_3

Problems to solve for supporting Ontology Engineering

Given "our" ontology $\mathcal{O}$
and ontologies $\mathcal{O}_i$ from which we want to reuse terms Σ_i,



1. make sure that $\mathcal{O}$ is safe for Σ_i
2. determine modules for Σ_i from $\mathcal{O}$ $\rightsquigarrow$ but which?
   (a) Did engineer "forget something" when specifying Σ_i?
   (b) Should modules be as small as possible?
   (c) Even minimal modules are not unique (see next slide)
   $\rightsquigarrow$ which one to use?
3. add modules $\mathcal{M}_i$ to $\mathcal{O}$
   (a) static/call-by-value: determine and add $\mathcal{M}_i$
   (b) dynamic/call-by-name: always use "freshest" $\mathcal{M}_i$ $\rightsquigarrow$ how?
   (We need to provide mechanisms/syntax for this.)

Example

Let $\Sigma = \{\mathsf{Knee}, \mathsf{HingeJoint}\}$. Suppose *Galen* contains:

$$
\begin{aligned}
\mathsf{Knee} &\equiv \mathsf{Joint} \sqcap \exists \mathsf{hasPart}.\mathsf{Patella} \sqcap & (1)\\
&\quad \exists \mathsf{hasFunct}.\mathsf{Hinge} &\\
\mathsf{Patella} &\sqsubseteq \mathsf{Bone} \sqcap \mathsf{Sesamoid} & (2)\\
\mathsf{Ginglymus} &\equiv \mathsf{Joint} \sqcap \exists \mathsf{hasFunct}.\mathsf{Hinge} & (3)\\
\mathsf{Joint} \sqcap \exists \mathsf{hasPart}.(\mathsf{Bone} \sqcap \mathsf{Sesamoid}) &\sqsubseteq \mathsf{Ginglymus} & (4)\\
\mathsf{Ginglymus} &\equiv \mathsf{HingeJoint} & (5)\\
\mathsf{Meniscus} &\equiv \mathsf{FibroCartilage} \sqcap \exists \mathsf{locatedIn}.\mathsf{Knee} &
\end{aligned}
$$

$\subseteq$-Minimal module for Σ? $\{(1), (2), (4), (5)\}$ and $\{(1), (3), (5)\}$

Note that a module for Σ does not necessarily contain

- only axioms that use terms from Σ
- all axioms that use terms from Σ

Bad news for expressive ontology languages?

Big, sad theorem

Let $\mathcal{O}_1,\ \mathcal{M} \subseteq \mathcal{O}_2$ be ontologies in $\mathcal{L}$ and Σ a signature.

1. Determining whether $\mathcal{O}_1$ is safe for $\mathcal{O}_2$ w.r.t. $\mathcal{L}$ or whether $\mathcal{M}$ is a module for $\mathcal{O}_1$ in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$ is

ExpTime-complete	for	$\mathcal{L} = \mathcal{EL}$,
2ExpTime-complete	for	$\mathcal{L} = \mathcal{ALC}, \mathcal{ALCQI}$, and
undecidable	for	$\mathcal{L} = \mathcal{ALCQIO}$ (almost OWL)

2. Determining whether $\mathcal{O}_1$ is safe for a signature Σ or whether $\mathcal{M}$ is a Σ-module in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$ is

   undecidable w.r.t. $\mathcal{L} = \mathcal{ALCO}$ (even if $\mathcal{O}_1$ is in $\mathcal{ALC}$).

[Konev, Lutz, Walther, Wolter 2009]
[Lutz and Wolter 2010]

Consequences for safety/modules of expressive DLs

Deciding safety/modules is highly complex or even undecidable for expressive DLs.

What to do?

1. Give up? No: modules/safety clearly too important
2. Reduce expressivity of logic? Yes!
3. Approximate for expressive logics? Yes – but from the *right* direction!

Tomorrow:

2. MEX modules for a fragment of $\mathcal{EL}$
3. 2 approximations, i.e., sufficient conditions for safety based on semantic and syntactic locality

And now . . .

1. Logical guarantees in detail

2. Overview of the remainder of this course

Course overview

3. Module extraction
   - Efficient module notions (locality, MEX)
   - Module extraction algorithms and tools
4. Decomposing ontologies
   - Atomic decomposition
5. Related notions and recent advances
   - Forgetting and interpolation
   - Logical difference
   - Reachability-based modules
   - Incremental/modular reasoning