Modularity in Ontologies: Locality-based modules

Thomas Schneider[1] Dirk Walther[2]

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

[2]Faculty of Informatics, Technical University of Madrid, Spain

ESSLLI, 3 August 2011

Plan for today

Yesterday, we discussed inseparability relations and their properties:

- inseparability = same functionality w.r.t. interface
  = same answers to queries
- decidability/complexity
- robustness under vocabulary extensions, joins, replacement

Today, we'll look a bit closer on how to use these insights to help ontology engineers re-use ontologies

- in a controlled way
- without (unwanted) side-effects

Thanks: partly based on slides by Uli Sattler.

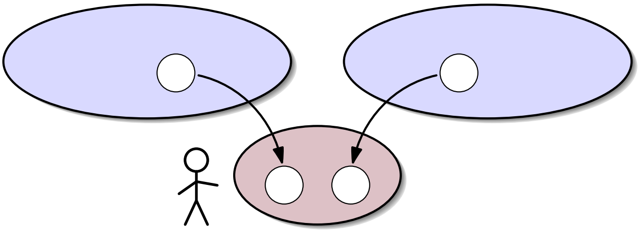
Plan for today

1. Motivation: Modular reuse of ontologies
2. Logical guarantees in detail
3. Efficient safety test and module extraction
4. Summary

And now . . .

Remember: an import/reuse scenario

"Borrow" knowledge from external ontologies



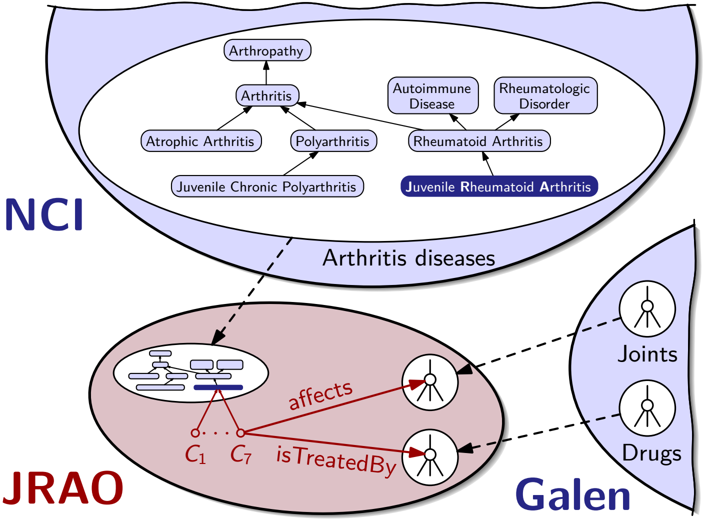
- Provides access to well-established knowledge
- Doesn't require expertise in external disciplines

This scenario is well-understood and implemented.

A *real* example

- Build an ontology *JRAO* that describes `JRA`
  `JRA` = Juvenile Rheumatoid Arthritis
- Describe `JRA` subkinds by
  - Joints affected
  - Occurrence of concomitant symptoms, e.g., fever
  - Treatment with certain drugs
- Re-use information provided by biomedical ontologies
  - *NCI*: diseases, drugs, proteins etc.
  - *Galen*: human anatomy

A real example



Why reuse an ontology?

- Saves time and effort
- Provides access to well-established knowledge and terminology
- Doesn't require expertise in drugs, proteins, anatomy etc.

$\rightsquigarrow$ A tool supporting reuse should **guarantee**:

- reusing imported terms doesn't change their meaning **Safety**
- all relevant parts of external ont.s are imported **Coverage**
  in addition, import *only* relevant parts (Economy)
- the order of imports doesn't matter **Independence**

Does this sound like inseparability?

Guarantees by example

Safety

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

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

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

iff

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

Guarantees by example

Coverage

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

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

iff

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

Guarantees by example

Independence

If *JRAO* is safe for *Galen* and for *NCI*, then
JRAO $\cup$ *NCI-module* is still safe for *Galen* and
JRAO $\cup$ *Galen-module* is still safe for *NCI*.

And now . . .

1. Motivation: Modular reuse of ontologies
2. Logical guarantees in detail
3. Efficient safety test and module extraction
4. Summary

Safety guarantee in detail

Safety for an ontology

$\mathcal{O}_1$ imports $\mathcal{O}_2$ in an $\mathcal{L}$-**safe** way (or $\mathcal{O}_1$ is **safe for** $\mathcal{O}_2$ w.r.t. $\mathcal{L}$)
if $\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}_2)} \mathcal{O}_2$.

Intuition: $\mathcal{O}_1 \cup \mathcal{O}_2$ doesn't change the meaning of $\mathcal{O}_2$-terms.

Safety guarantee in detail

Safety for an ontology

$\mathcal{O}_1$ imports $\mathcal{O}_2$ in an $\mathcal{L}$-**safe** way (or $\mathcal{O}_1$ is **safe for** $\mathcal{O}_2$ w.r.t. $\mathcal{L}$)

if $\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}_2)} \mathcal{O}_2$.

Intuition: $\mathcal{O}_1 \cup \mathcal{O}_2$ doesn't change the meaning of $\mathcal{O}_2$-terms.

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 $\mathcal{O}_2$ and $\mathsf{sig}(\mathcal{O}_2)$

  $\mathcal{O}_2 = \mathit{NCI}$ might change over time, we want latest version

Safety guarantee in detail

Safety for an ontology

$\mathcal{O}_1$ imports $\mathcal{O}_2$ in an $\mathcal{L}$-**safe** way (or $\mathcal{O}_1$ is **safe for** $\mathcal{O}_2$ w.r.t. $\mathcal{L}$)
if $\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}_2)} \mathcal{O}_2$.

Intuition: $\mathcal{O}_1 \cup \mathcal{O}_2$ doesn't change the meaning of $\mathcal{O}_2$-terms.

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 $\mathcal{O}_2$ and $\mathsf{sig}(\mathcal{O}_2)$

  $\mathcal{O}_2 = NCI$ might change over time, we want latest version

Solution: Safety for a signature!

Safety for a signature

Definition

$\mathcal{O}_1$ is **safe for Σ w.r.t. $\mathcal{L}$** if,

for every $\mathcal{L}$-ontology $\mathcal{O}_2$ with $\mathsf{sig}(\mathcal{O}_1) \cap \mathsf{sig}(\mathcal{O}_2) \subseteq \Sigma$,
$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O}_2$.

Safety for a signature

Definition

$\mathcal{O}_1$ is **safe for Σ w.r.t. $\mathcal{L}$** if,

for every $\mathcal{L}$-ontology $\mathcal{O}_2$ with $\mathsf{sig}(\mathcal{O}_1) \cap \mathsf{sig}(\mathcal{O}_2) \subseteq \Sigma$,
$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O}_2$.

Theorem

1. If $\mathcal{O}_1$ is a model Σ-conservative extension of $\emptyset$ $(\mathcal{O}_1 \equiv^{\mathsf{SO}}_{\Sigma} \emptyset)$, then $\mathcal{O}_1$ is safe for Σ w.r.t. any $\mathcal{L} \leqslant \mathsf{SO}$.
2. Let $\mathcal{L}$ be robust under replacements. Then $\mathcal{O}_1$ is safe for Σ w.r.t. $\mathcal{L}$ iff $\mathcal{O}_1 \equiv^{\mathcal{L}}_{\Sigma} \emptyset$.

Safety for a signature

Definition

$\mathcal{O}_1$ is **safe for Σ w.r.t. $\mathcal{L}$** if,

for every $\mathcal{L}$-ontology $\mathcal{O}_2$ with $\mathsf{sig}(\mathcal{O}_1) \cap \mathsf{sig}(\mathcal{O}_2) \subseteq \Sigma$,
$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\Sigma} \mathcal{O}_2$.

Theorem

1. If $\mathcal{O}_1$ is a model Σ-conservative extension of $\emptyset$ $\quad(\mathcal{O}_1 \equiv^{\mathsf{SO}}_{\Sigma} \emptyset)$,
then $\mathcal{O}_1$ is safe for Σ w.r.t. any $\mathcal{L} \leqslant \mathsf{SO}$.
2. Let $\mathcal{L}$ be robust under replacements.
Then $\mathcal{O}_1$ is safe for Σ w.r.t. $\mathcal{L}$ iff $\mathcal{O}_1 \equiv^{\mathcal{L}}_{\Sigma} \emptyset$.

Bad news: robustness under replacements fails easily . . .

When robustness under replacements fails

Take ontology language $\mathcal{ALC}$ and $\mathcal{L}$ = "$\mathcal{ALC}$-concept inclusions".

Consider $\mathcal{O}_1 = \{A \sqsubseteq \exists r.B\}$ and $\Sigma = \{A, B\}$.

When robustness under replacements fails

Take ontology language $\mathcal{ALC}$ and $\mathcal{L}$ = "$\mathcal{ALC}$-concept inclusions".

Consider $\mathcal{O}_1 = \{A \sqsubseteq \exists r.B\}$ and $\Sigma = \{A, B\}$.

- $\mathcal{O}_1 \equiv_\Sigma^{\mathcal{ALC}} \emptyset$,
- but if we take $\mathcal{O}_2 = \{A \equiv \top,\ B \equiv \bot\}$,
  then $\mathcal{O}_1 \cup \mathcal{O}_2 \models \top \sqsubseteq \bot$,
  while $\mathcal{O}_2 \not\models \top \sqsubseteq \bot$.
- Hence, $\mathcal{O}_1 \cup \mathcal{O}_2 \not\equiv_\Sigma^{\mathcal{ALC}} \mathcal{O}_2$.
- Hence, $\mathcal{O}_1$ is not safe for Σ.

When robustness under replacements fails

Take ontology language $\mathcal{ALC}$ and $\mathcal{L}$ = "$\mathcal{ALC}$-concept inclusions".

Consider $\mathcal{O}_1 = \{A \sqsubseteq \exists r.B\}$ and $\Sigma = \{A, B\}$.

- $\mathcal{O}_1 \equiv_{\Sigma}^{\mathcal{ALC}} \emptyset$,
- but if we take $\mathcal{O}_2 = \{A \equiv \top,\ B \equiv \bot\}$,
  then $\mathcal{O}_1 \cup \mathcal{O}_2 \models \top \sqsubseteq \bot$,
  while $\mathcal{O}_2 \not\models \top \sqsubseteq \bot$.
- Hence, $\mathcal{O}_1 \cup \mathcal{O}_2 \not\equiv_{\Sigma}^{\mathcal{ALC}} \mathcal{O}_2$.
- Hence, $\mathcal{O}_1$ is not safe for Σ.

(This problem can be resolved by extending $\mathcal{L}$ to
"Boolean conjunctions of $\mathcal{ALC}$-concept inclusions".)

Coverage guarantee in detail

Module for an ontology

$\mathcal{M} \subseteq \mathcal{O}_2$ is a **module for $\mathcal{O}_1$ in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$** if

$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}_1)} \mathcal{O}_1 \cup \mathcal{M}$.

Coverage guarantee in detail

Module for an ontology

$\mathcal{M} \subseteq \mathcal{O}_2$ is a **module for $\mathcal{O}_1$ in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$** if

$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}_1)} \mathcal{O}_1 \cup \mathcal{M}$.

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.

Coverage guarantee in detail

Module for an ontology

$\mathcal{M} \subseteq \mathcal{O}_2$ is a **module for $\mathcal{O}_1$ in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$** if

$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathsf{sig}(\mathcal{O}_1)} \mathcal{O}_1 \cup \mathcal{M}$.

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!
$\leadsto$ interoperability of $\mathcal{M}$

Module for a signature

Definition

$\mathcal{M} \subseteq \mathcal{O}_2$ is **a module for Σ in $\mathcal{O}_2$** w.r.t. $\mathcal{L}$ if,

for every $\mathcal{L}$-ontology $\mathcal{O}_1$ with $\mathrm{sig}(\mathcal{O}_1) \cap \mathrm{sig}(\mathcal{O}_2) \subseteq \Sigma$,
$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathrm{sig}(\mathcal{O}_1)} \mathcal{O}_1 \cup \mathcal{M}$.

Module for a signature

Definition

$\mathcal{M} \subseteq \mathcal{O}_2$ is **a module for Σ in $\mathcal{O}_2$** w.r.t. $\mathcal{L}$ if,

for every $\mathcal{L}$-ontology $\mathcal{O}_1$ with $\mathrm{sig}(\mathcal{O}_1) \cap \mathrm{sig}(\mathcal{O}_2) \subseteq \Sigma$,
$\mathcal{O}_1 \cup \mathcal{O}_2 \equiv^{\mathcal{L}}_{\mathrm{sig}(\mathcal{O}_1)} \mathcal{O}_1 \cup \mathcal{M}$.

Observation

1. If $\mathcal{M} \subseteq \mathcal{O}_2$ and $\mathcal{O}_2$ is a model Σ-c.e. of $\mathcal{M}$ $(\mathcal{O}_2 \equiv^{\mathsf{SO}}_{\Sigma} \mathcal{M})$,
   then $\mathcal{M}$ is a module for Σ in $\mathcal{O}_2$ w.r.t. any $\mathcal{L} \leqslant \mathsf{SO}$
2. Let $\mathcal{L}$ be robust under replacements.
   Then $\mathcal{M} \subseteq \mathcal{O}_2$ is a module for Σ in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$
   iff $\mathcal{O}_2 \setminus \mathcal{M} \equiv^{\mathcal{L}}_{\Sigma} \emptyset$.

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 $\emptyset$

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 $\Sigma \cup \mathsf{sig}(\mathcal{M})$ 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 $\Sigma \cup \mathsf{sig}(\mathcal{M})$

Independence Guarantee in Detail

Basic requirement for importing ontologies independently.

Independence

Safety is preserved under imports:

If $\mathcal{O}_1$ is safe for Σ_i ($\mathcal{O}_i$), then $\mathcal{O}_1 \cup \mathcal{O}_j$ is still safe for Σ_i ($\mathcal{O}_i$).

Independence Guarantee in Detail

Basic requirement for importing ontologies independently.

Independence

Safety is preserved under imports:

If $\mathcal{O}_1$ is safe for Σ_i $(\mathcal{O}_i)$, then $\mathcal{O}_1 \cup \mathcal{O}_j$ is still safe for Σ_i $(\mathcal{O}_i)$.

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

- when the Σ_i share terms:

  e.g., $\mathcal{O}_1 = \{A \sqsubseteq \top\}$ is safe for $\Sigma = \{A, B\}$,
  but $\mathcal{O}_1 \cup \{A \sqsubseteq B\}$ is *not safe* for Σ

- when the Σ_i don't share terms:

  e.g., $\mathcal{O}_1 = \{A \sqsubseteq B\}$ is safe for $\Sigma_2 = \{A\}$ and $\Sigma_3 = \{B\}$,
  but $\mathcal{O}_1 \cup \{B \equiv \bot\}$ is *not safe* for Σ_2
  and $\mathcal{O}_1 \cup \{A \equiv \top\}$ is *not safe* for Σ_3

Problems to solve for supporting Ontology Engineering

Given “our” ontology $\mathcal{O}_1$
and ontologies $\mathcal{O}_i$ from which we want to reuse terms Σ_i,

1. make sure that $\mathcal{O}_1$ is safe for Σ_i
2. determine modules for Σ_i from $\mathcal{O}_i$ $\rightsquigarrow$ 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 $\mathcal{O}_1$
   (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 $\Sigma = \{\text{Knee}, \text{HingeJoint}\}$. Suppose *Galen* contains:

$$
\begin{aligned}
\text{Knee} &\equiv \text{Joint} \sqcap \exists \text{hasPart}.\text{Patella} \sqcap && (1)\\
&\quad \exists \text{hasFunct}.\text{Hinge} && \\
\text{Patella} &\sqsubseteq \text{Bone} \sqcap \text{Sesamoid} && (2)\\
\text{Ginglymus} &\equiv \text{Joint} \sqcap \exists \text{hasFunct}.\text{Hinge} && (3)\\
\text{Joint} \sqcap \exists \text{hasPart}.(\text{Bone} \sqcap \text{Sesamoid}) &\sqsubseteq \text{Ginglymus} && (4)\\
\text{Ginglymus} &\equiv \text{HingeJoint} && (5)\\
\text{Meniscus} &\equiv \text{FibroCartilage} \sqcap \exists \text{locatedIn}.\text{Knee} && (6)
\end{aligned}
$$

$\subseteq$-Minimal module for Σ ?

Example

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

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

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

Example

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

$$\mathsf{Knee} \equiv \mathsf{Joint} \sqcap \exists \mathsf{hasPart}.\mathsf{Patella} \sqcap \exists \mathsf{hasFunct}.\mathsf{Hinge} \tag{1}$$

$$\mathsf{Ginglymus} \equiv \mathsf{Joint} \sqcap \exists \mathsf{hasFunct}.\mathsf{Hinge} \tag{3}$$

$$\mathsf{Ginglymus} \equiv \mathsf{HingeJoint} \tag{5}$$

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

Example

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

$$
\begin{aligned}
\text{Knee} &\equiv \text{Joint} \sqcap \exists\text{hasPart.Patella} \sqcap && (1)\\
&\qquad \exists\text{hasFunct.Hinge} && \\
\text{Patella} &\sqsubseteq \text{Bone} \sqcap \text{Sesamoid} && (2)\\
\text{Ginglymus} &\equiv \text{Joint} \sqcap \exists\text{hasFunct.Hinge} && (3)\\
\text{Joint} \sqcap \exists\text{hasPart.}(\text{Bone} \sqcap \text{Sesamoid}) &\sqsubseteq \text{Ginglymus} && (4)\\
\text{Ginglymus} &\equiv \text{HingeJoint} && (5)\\
\text{Meniscus} &\equiv \text{FibroCartilage} \sqcap \exists\text{locatedIn.Knee} && (6)
\end{aligned}
$$

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

Example

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

$$
\begin{aligned}
\mathsf{Knee} &\equiv \mathsf{Joint} \sqcap \exists \mathsf{hasPart}.\mathsf{Patella} \sqcap && (1)\\
&\quad\; \exists \mathsf{hasFunct}.\mathsf{Hinge} &&\\
\mathsf{Patella} &\sqsubseteq \mathsf{Bone} \sqcap \mathsf{Sesamoid} && (2)\\
\mathsf{Ginglymus} &\equiv \mathsf{Joint} \sqcap \exists \mathsf{hasFunct}.\mathsf{Hinge} && (3)\\
\mathsf{Joint} \sqcap \exists \mathsf{hasPart}.(\mathsf{Bone} \sqcap \mathsf{Sesamoid}) &\sqsubseteq \mathsf{Ginglymus} && (4)\\
\mathsf{Ginglymus} &\equiv \mathsf{HingeJoint} && (5)\\
\mathsf{Meniscus} &\equiv \mathsf{FibroCartilage} \sqcap \exists \mathsf{locatedIn}.\mathsf{Knee} && (6)
\end{aligned}
$$

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

Note that a module for Σ does not necessarily contain

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

Bad news for expressive ontology languages?

Big, sad theorem ➳ *Tuesday's lecture*

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 $\mathcal{L} = \mathcal{EL}$, ➳ *Tuesday's lecture*
   2ExpTime-compl. for $\mathcal{ALC} \leqslant \mathcal{L} \leqslant \mathcal{ALCQI}$, and
   undecidable for $\mathcal{L} \geqslant \mathcal{ALCQIO}$, including 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 $\mathcal{ALC}$).

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! ➺ *Thursday's lecture*
3. Approximate for expressive logics? Yes – but from the *right* direction!

Next: 2 approximations, i.e., sufficient conditions for safety

1. based on semantic locality
2. based on syntactic locality

And now . . .

Locality

Remember:

$$\mathcal{O} \text{ is } \Sigma\text{-safe w.r.t. any } \mathcal{L}$$

$$\text{if}$$

$$\mathcal{O} \text{ is a model } \Sigma\text{-conserv. extension of } \emptyset$$

$$\text{iff}$$

$$\text{for each } \mathcal{I}, \text{ there is } \mathcal{J} \models \mathcal{O} \text{ with } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$$

Locality

$$\mathcal{O} \text{ is } \Sigma\text{-safe w.r.t. any } \mathcal{L}$$
$$\text{if}$$
$$\mathcal{O} \text{ is a model } \Sigma\text{-conserv. extension of } \emptyset$$
$$\text{iff}$$
$$\text{for each } \mathcal{I}, \text{ there is } \mathcal{J} \models \mathcal{O} \text{ with } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$$
$$\text{if}$$
$$\forall \mathcal{I}\ \exists \mathcal{J} \models \mathcal{O} \text{ with } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma \text{ and } X^{\mathcal{I}} = \emptyset,\ \forall X \notin \Sigma$$

Locality

$\mathcal{O}$ is Σ-safe w.r.t. any $\mathcal{L}$

if

$\mathcal{O}$ is a model Σ-conserv. extension of $\emptyset$

iff

for each $\mathcal{I}$, there is $\mathcal{J} \models \mathcal{O}$ with $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$

if

$\forall \mathcal{I}\ \exists \mathcal{J} \models \mathcal{O}$ with $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ and $X^\mathcal{I} = \emptyset$, $\forall X \notin \Sigma$

iff

$\forall \mathcal{I}\ \exists \mathcal{J}\ \forall \alpha \in \mathcal{O} : \mathcal{J} \models \alpha$ and $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ and $X^\mathcal{I} = \emptyset$, $\forall X \notin \Sigma$

Locality

$$\mathcal{O} \text{ is } \Sigma\text{-safe w.r.t. any } \mathcal{L}$$
if
$$\mathcal{O} \text{ is a model } \Sigma\text{-conserv. extension of } \emptyset$$
iff
$$\text{for each } \mathcal{I}, \text{ there is } \mathcal{J} \models \mathcal{O} \text{ with } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$$
if
$$\forall \mathcal{I}\ \exists \mathcal{J} \models \mathcal{O} \text{ with } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma \text{ and } X^\mathcal{I} = \emptyset,\ \forall X \notin \Sigma$$
iff
$$\forall \mathcal{I}\ \exists \mathcal{J}\ \forall \alpha \in \mathcal{O} : \mathcal{J} \models \alpha \text{ and } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma \text{ and } X^\mathcal{I} = \emptyset,\ \forall X \notin \Sigma$$
iff
$$\forall \mathcal{I}\ \forall \alpha \in \mathcal{O}\ \exists \mathcal{J} : \mathcal{J} \models \alpha \text{ and } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma \text{ and } X^\mathcal{I} = \emptyset,\ \forall X \notin \Sigma$$

Locality

$$\mathcal{O} \text{ is } \Sigma\text{-safe w.r.t. any } \mathcal{L}$$
if
$$\mathcal{O} \text{ is a model } \Sigma\text{-conserv. extension of } \emptyset$$
iff
$$\text{for each } \mathcal{I}, \text{ there is } \mathcal{J} \models \mathcal{O} \text{ with } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$$
if
$$\forall \mathcal{I}\ \exists \mathcal{J} \models \mathcal{O} \text{ with } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma \text{ and } X^\mathcal{I} = \emptyset, \forall X \notin \Sigma$$
iff
$$\forall \mathcal{I}\ \exists \mathcal{J}\ \forall \alpha \in \mathcal{O} : \mathcal{J} \models \alpha \text{ and } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma \text{ and } X^\mathcal{I} = \emptyset, \forall X \notin \Sigma$$
iff
$$\forall \mathcal{I}\ \forall \alpha \in \mathcal{O}\ \exists \mathcal{J} : \mathcal{J} \models \alpha \text{ and } \mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma \text{ and } X^\mathcal{I} = \emptyset, \forall X \notin \Sigma$$
iff
$$\forall \alpha \in \mathcal{O} : \text{“}\alpha \text{ with all } X \notin \Sigma \text{ replaced by } \bot\text{” is a tautology}$$

Testing locality

Ergo: $\mathcal{O}$ is Σ-safe w.r.t. any $\mathcal{L}$ if:
for each $\alpha \in \mathcal{O}$ and each $\mathcal{I}$ where all $r, A \notin \Sigma$ are interpreted as $\emptyset$, we have $\mathcal{I} \models \alpha$.

Algorithm for testing locality

Input: Σ, $\mathcal{O}$ $\mathcal{ALC}$ TBox

`safe` $\leftarrow$ true
For each $C_1 \sqsubseteq C_2 \in \mathcal{O}$ with C_i in NNF, construct C_i' from C_i by
 replacing all $A \notin \Sigma$ with $\bot$
 replacing all $\exists r.C$ with $r \notin \Sigma$ with $\bot$
 replacing all $\forall r.C$ with $r \notin \Sigma$ with $\top$
`safe` $\leftarrow$ false if $C_1' \sqcap \neg C_2'$ is satisfiable % *can find countermodel*
Return `safe`

Answers "true" if $\mathcal{O}$ is Σ-safe w.r.t. $\mathcal{ALC}$; extensible to more expressive DLs

Dual notion of locality

Analogously: $\mathcal{O}$ is Σ-safe w.r.t. any $\mathcal{L}$ if:
for each $\alpha \in \mathcal{O}$ and each $\mathcal{I}$ where all $r, A \notin \Sigma$ are interpreted as Δ,
we have $\mathcal{I} \models \alpha$.

Algorithm for testing locality

Input: Σ, $\mathcal{O}$ $\mathcal{ALC}$ TBox

`safe` $\leftarrow$ true
For each $C_1 \sqsubseteq C_2 \in \mathcal{O}$ with C_i in NNF, construct C_i' from C_i by
- replacing all $A \notin \Sigma$ with $\top$
- replacing all $\exists r.\top$ with $r \notin \Sigma$ with $\top$
- replacing all $\forall r.\bot$ with $r \notin \Sigma$ with $\bot$

`safe` $\leftarrow$ false if $C_1' \sqcap \neg C_2'$ is satisfiable % *can find countermodel*
Return `safe`

Answers "true" if $\mathcal{O}$ is Σ-safe w.r.t. $\mathcal{ALC}$; extensible to more expressive DLs

Testing locality

Both variants of our algorithm decide Σ-safety.

But:

- Both locality notions only *approximate* Σ-safety.
- We still need to perform **reasoning:**
  for each axiom α, test satisfiability of $C_1' \sqcap \neg C_2'$
  - There are highly optimised reasoners available to do so, but ...
  - Testing satisfiability in $\mathcal{ALC}$ is ExpTime-complete!
  - Testing satisfiability in $\mathcal{SROIQ}$ is N2ExpTime-complete!

Q: Isn't there a **cheaper** approximation?

A: We can use **syntactic approximation** of locality!

Syntactic approximation of locality

Axiom α is **syntactically Σ-local**: α of form $C \sqsubseteq C^{\Delta}$ or $C^{\emptyset} \sqsubseteq C$, for $C^{\emptyset}$ and C^{Δ} given by the following grammars.

Start with $A^{\not\Sigma}$, $r^{\not\Sigma}$ terms *not* in Σ, and r, C *any* term

$$C^{\emptyset} ::= A^{\not\Sigma} \mid \neg C^{\Delta} \mid C \sqcap C^{\emptyset} \mid C^{\emptyset} \sqcap C \mid \exists r^{\not\Sigma}.C \mid \exists r.C^{\emptyset}$$
$$C^{\Delta} ::= \top \mid \neg C^{\emptyset} \mid C^{\Delta} \sqcap C^{\Delta}$$

An ontology is **syntactically Σ-local**
if it contains only syntactically Σ-local axioms.

Syntactic approximation of locality

Axiom α is **syntactically Σ-local**: α of form $C \sqsubseteq C^{\Delta}$ or $C^{\emptyset} \sqsubseteq C$, for $C^{\emptyset}$ and C^{Δ} given by the following grammars.

Start with $A^{\not\Sigma}$, $r^{\not\Sigma}$ terms *not* in Σ, and r, C *any* term

$$C^{\emptyset} ::= A^{\not\Sigma} \mid \neg C^{\Delta} \mid C \sqcap C^{\emptyset} \mid C^{\emptyset} \sqcap C \mid \exists r^{\not\Sigma}.C \mid \exists r.C^{\emptyset}$$
$$C^{\Delta} ::= \top \mid \neg C^{\emptyset} \mid C^{\Delta} \sqcap C^{\Delta}$$

An ontology is **syntactically Σ-local**
if it contains only syntactically Σ-local axioms.

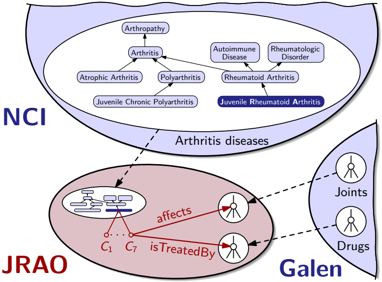
Theorem

Syntactic Σ-locality implies semantic Σ-locality implies Σ-safety

Examples of syntactically (non)-local axioms

$\overline{B} \sqsubseteq A$	form $C \sqsubseteq C^{\emptyset} \rightsquigarrow$ not $\{\overline{B}, \dots\}$-local
$A \sqsubseteq \overline{B} \sqcap \exists r.\overline{C}$	form $C^{\emptyset} \sqsubseteq C \rightsquigarrow \{\overline{B}, \overline{C}\}$-local
$X \sqcap A \sqsubseteq Y$	is Σ-local if, e.g., $A \notin \Sigma$
$\overline{B} \sqcap \exists r.\overline{C} \sqsubseteq A$	is $\{\overline{B}, \overline{C}\}$-local
$\overline{A} \sqsubseteq \overline{A} \sqcup \overline{B}$	is not $\{\overline{A}, \overline{B}\}$-local, yet a tautology!

Back to our real example



In *JRAO*, we can reuse

$$\{\overline{\text{Arthritis}}, \overline{\text{Joint}}, \overline{\text{Knee}}\}$$

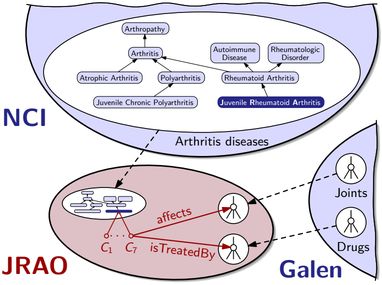
and "syntactically safely" write:

$$\text{JRA} \equiv \overline{\text{Arthritis}} \sqcap \exists\text{affects}.(\overline{\text{Joint}} \sqcap \exists\text{locatedIn}.\text{Juvenile})$$

$$\text{KJRA} \equiv \text{JRA} \sqcap \exists\text{affects}.\overline{\text{Knee}}$$

$\rightsquigarrow$ safely reference and refine existing terms from *NCI* and *Galen*.

Back to our real example



In *JRAO*, we can reuse

$$\{\overline{\text{Arthritis}}, \overline{\text{Joint}}, \overline{\text{Knee}}\}$$

and "syntactically safely" write:

$$\text{JRA} \equiv \overline{\text{Arthritis}} \sqcap \exists \text{affects}.(\overline{\text{Joint}} \sqcap \exists \text{locatedIn}.\text{Juvenile})$$

$$\text{KJRA} \equiv \text{JRA} \sqcap \exists \text{affects}.\overline{\text{Knee}}$$

$\leadsto$ safely reference and refine existing terms from *NCI* and *Galen*.

Generalise terms? – Use different syntactic locality: dual notion

Locality for modules

Remember: If $\mathcal{O}_2 \setminus \mathcal{M}$ is safe for $\Sigma \cup \text{sig}(\mathcal{M})$ w.r.t. $\mathcal{L}$, then $\mathcal{M}$ is a Σ-module in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$.

$\leadsto$ poly-time algorithm to **compute a Σ-module in $\mathcal{O}_2$**:

Locality for modules

Remember: 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. $\mathcal{L}$.

$\leadsto$ poly-time algorithm to **compute a Σ-module in $\mathcal{O}_2$**:

Algorithm

Input: Sig. Σ, TBox $\mathcal{O}$

$\mathcal{M} \leftarrow \emptyset$, $\Sigma_1 \leftarrow \Sigma$, $\Sigma_0 \leftarrow \Sigma$
Repeat $\Sigma_0 \leftarrow \Sigma_1$
    For each $\alpha \in \mathcal{O}_2 \setminus \mathcal{M}$
        If α **not Σ_1-safe**, then add α to $\mathcal{M}$ and $\mathrm{sig}(\alpha)$ to Σ_1
Until $\Sigma_0 = \Sigma_1$
Return $\mathcal{M}$

Locality for modules

Remember: If $\mathcal{O}_2 \setminus \mathcal{M}$ is safe for $\Sigma \cup \mathsf{sig}(\mathcal{M})$ w.r.t. $\mathcal{L}$, then $\mathcal{M}$ is a Σ-module in $\mathcal{O}_2$ w.r.t. $\mathcal{L}$.

$\leadsto$ poly-time algorithm to **compute a Σ-module in $\mathcal{O}_2$**:

Algorithm

Input: Sig. Σ, TBox $\mathcal{O}$

$\mathcal{M} \leftarrow \emptyset, \quad \Sigma_1 \leftarrow \Sigma, \quad \Sigma_0 \leftarrow \Sigma$

Repeat $\Sigma_0 \leftarrow \Sigma_1$

For each $\alpha \in \mathcal{O}_2 \setminus \mathcal{M}$

If α **not Σ_1-safe**, then add α to $\mathcal{M}$ and $\mathsf{sig}(\alpha)$ to Σ_1

Until $\Sigma_0 = \Sigma_1$

Return $\mathcal{M}$

Observation: $\mathcal{M}$ is a Σ_1-module in $\mathcal{O}$ and therefore a Σ-module (since $\Sigma \subseteq \Sigma_1$ and – we need some anti-monotonicity here)

Variations to the module extraction algorithm

- Different safety checks, based on locality, lead to different notions of a **locality-based modules**:
  - semantic locality $\rightsquigarrow$ "$\emptyset$-modules"
  - dual notion $\rightsquigarrow$ "Δ-modules"
  - syntactic locality ($\bot$-locality) $\rightsquigarrow$ $\bot$-modules
  - dual notion ($\top$-locality) $\rightsquigarrow$ $\top$-modules
  - Remember: the first two require reasoning (often intractable), while a syntactic locality check is tractable!
- Smaller modules by nesting $\top$- and $\bot$-module extraction: $\top\bot^*$-modules
- More efficient extraction of (semantic) $\emptyset$- and Δ-modules: start with extracting a $\bot$- or $\top$-module

And now ...

1. Motivation: Modular reuse of ontologies
2. Logical guarantees in detail
3. Efficient safety test and module extraction
4. Summary

Summary

- Safety and economy/coverage are important guarantees (not only) for reuse.
- They can be approximated using locality.
- Modules based on syntactic locality can be extracted efficiently in logics up to OWL.
- There is tool support for extracting modules.
  `http://owl.cs.manchester.ac.uk/modularity`
  `http://owlapi.sourceforge.net/`
- This line of research is rather new for DLs and ontology languages, and many questions are (half)open.

Course overview

4. Versioning and Forgetting
    - Logical difference
    - Forgetting/uniform interpolants
5. Recent Advances/Current Work
    - Atomic decomposition
    - Signature decomposition, relevance of terms