Description Logics: a Nice Family of Logics — Modularity —

Uli Sattler\textsuperscript{1}  \hspace{1em} Thomas Schneider\textsuperscript{2}

\textsuperscript{1}School of Computer Science, University of Manchester, UK
\textsuperscript{2}Department of Computer Science, University of Bremen, Germany

ESSLLI, 19 August 2016

Plan for today

1. What is modularity good for?
2. Modules for reuse
3. Summary and Outlook

What can I do with my ontology?

Ontology users and engineers want to use ontologies to

- represent and archive knowledge (M)
  in a structured way
- compute inferences from archived knowledge (M)
  e.g., classification, query answering
- explain inferences (M)
  justifications = pinpointing, abduction
- reuse (parts of) other ontologies to build their ontology (M)
  import
- expose the logical structure of the represented knowledge (M)
  comprehension

(M) = modularity helps

And now . . .

1. What is modularity good for?
2. Modules for reuse
3. Summary and Outlook
What can I do with my ontology?

Building and using an ontology often requires

- fast reasoning (M)
  - expressivity ↔ complexity; optimisations, incremental reasoning
- collaborative development (M)
- version control (M)
- efficient reuse (M)
- an understanding of the ontology’s content and structure (M)

\( (M) = \text{modularity helps} \)

A priori vs. a posteriori modularisation

A priori (not covered today)

- At first, a modular structure is decided on.
- Then, the ontology is developed and used according to that structure.

A posteriori

- The ontology is regarded as a monolithic entity.
- At some point, a module is extracted or the ontology is decomposed into several modules.

Comparing two ontologies

Assume that . . .

- you want to buy a medical ontology from me
- I offer two medical ontologies \( O_1 \) and \( O_2 \)

Q: which one do you choose?

Possible A: the one that contains more knowledge.

Q: how do you measure the amount of knowledge in \( O_i \)?

Possible A: Number of axioms?

- Well, compare \( \{ A \sqsubseteq B, \ B \sqsubseteq A \} \) vs. \( \{ A \equiv B \} \)
- or \( \{ A \sqsubseteq B, \ B \sqsubseteq A \Rightarrow A, \ A \sqsubseteq \neg A \sqcap B \} \) vs. \( \{ A \equiv B \} \)

Possible A: Number of entailments? Number of models?
Ontologies and their entailments

Think of axioms as generating entailments – e.g.:

\[ A \sqsubseteq \exists r.B \quad \exists r.T \sqsubseteq C \sqcap D \]

\[ \models A \sqsubseteq D \]

Q: How many entailments can a TBox have?

A: \( \infty \)

\[ A \sqsubseteq D \quad A \sqsubseteq D \sqcup A \quad A \sqsubseteq D \sqcup (A \sqcap D), \ldots \]

Ontologies and their models

Think of axioms as restricting possible models – e.g.:

- Hand \( \sqsubseteq \exists \) hasPart.Finger
  \( \models \) models cannot have instances of Hand with no hasPart-edge to an instance of Finger

- Hand \( \sqsubseteq = 5 \) hasPart.Finger
  \( \models \) models cannot have instances of Hand with \( \neq 5 \) hasPart-edges to instances of Finger

Q: How many models can a TBox have?

A: \( \infty \)

Next attempt at “more” entailments/models

We cannot compare numbers of entailments or models

But we can use set inclusion:
“\( O \) knows at most as much as \( O' \)” if

- every entailment of \( O \) is one of \( O' \):
  \[ \{ \eta \mid O \models \eta \} \subseteq \{ \eta \mid O' \models \eta \} \] or

- every model of \( O' \) is one of \( O' \):
  \[ \{ I \mid I \models O' \} \subseteq \{ I \mid I \models O \} \]

Problem:

How do we test these conditions?

Knowledge w.r.t. a signature

Let’s reformulate the initial dialogue.
Assume that …

- you want to buy a subset of a medical ontology \( O \) from me that covers the subdomain of, say, diseases
- I offer two subsets \( M_1 \) and \( M_2 \)

Q: Which one do you choose?

Possible A: The one that “knows more” about diseases!

Q: Which is the best subset I can offer?

Possible A: A module for diseases

- \( M \sqsubseteq O \) that knows as much as \( O \) about diseases:
  \( M \) indistinguishable from \( O \) w.r.t. all terms relevant for diseases
- \( M \) as small as possible
Inseparability w.r.t. a signature

**Definition**
- **Signature** $\Sigma$ = a set of concept/role names
- The signature of axiom (ontology) $X$ = all concept/role names in $X$
- $O_1$ and $O_2$ are $\Sigma$-inseparable w.r.t. a logic $L$, written $O_1 \equiv_{\Sigma} L O_2$, if:
  - for all $\eta \in L$ with $\text{sig}(\eta) \subseteq \Sigma$, $O_1 \models \eta$ iff $O_2 \models \eta$

**O1 is a $\Sigma$-module of M w.r.t. L**

The signature $\Sigma$ ...
- can be seen as a “topic”
- that the module is required to cover
- is difficult to formulate:
  - $Q$: how many interesting entailments in $\Sigma = \{\text{Disease}\}$ can $O$ possibly have?

Choosing the logic $L$

**Definition (repeated from previous slide)**
- $O$ is a $\Sigma$-module of $M$ w.r.t. $L$
  - if $M \subseteq O$ and $M \equiv_{\Sigma}^L O$

Choice of $L$ depends on your usage of the module:
- for ontology design: subsumptions between (complex?) concepts
- for ontology usage: your favourite query language

Modules for reuse

If we want to reuse module $M$, we need a stronger guarantee:
- $M \cup O' \equiv_{\Sigma}^L O \cup O'$ for all $O'$
  - i.e., we can safely import $M$ into any $O'$

Ensured by two additional requirements:

**Lemma** [Konev et al. 2009]
- If $M \subseteq O$ and $M \equiv_{\Sigma}^L O$, then $M \cup O' \equiv_{\Sigma}^L O \cup O'$, for
  - every $O'$ with $\text{sig}(O) \cap \text{sig}(O') \subseteq \Sigma$,
  - expressive enough $L$, e.g. $\text{SROIQ}$ (OWL).

(1) means that $O'$ may reuse only terms from $\Sigma$
### How is a minimal $\Sigma$-module extracted?

**Simple module extraction algorithm:**

1. $M \leftarrow \mathcal{O}$
2. While $M \setminus \{\alpha\} \equiv_{\Sigma} \mathcal{O}$, for some $\alpha \in M$, do $M \leftarrow M \setminus \{\alpha\}$
3. Output $M$

**Observation:**

Different orders of choosing $\alpha$ can lead to different minimal modules

### Example

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

1. Knee $\equiv$ Joint $\sqcap \exists$hasPart.Patella $\sqcap \exists$hasFunct.Hinge
2. Patella $\subseteq$ Bone $\sqcap$ Sesamoid
3. Ginglymus $\equiv$ Joint $\sqcap \exists$hasFunct.Hinge
4. Joint $\sqcap \exists$hasPart.(Bone $\sqcap$Sesamoid) $\subseteq$ Ginglymus
5. Ginglymus $\equiv$ HingeJoint
6. Meniscus $\equiv$ FibroCartilage $\sqcap \exists$locatedIn.Knee

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

Note that a module for $\Sigma$ does not necessarily contain
- all axioms that use terms from $\Sigma$
- only axioms that only use terms from $\Sigma$

### Bad news for expressive ontology languages?

**Big, sad theorem [Ghilardi et al. 2006]**

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

Determining whether $\mathcal{O}_1 \equiv_{\Sigma} \mathcal{O}_2$ is

<table>
<thead>
<tr>
<th>EXP-TIME-complete</th>
<th>2EXP-TIME-complete</th>
<th>undecidable</th>
</tr>
</thead>
<tbody>
<tr>
<td>for $\mathcal{L} = \mathcal{EL}$</td>
<td>for $\mathcal{ALC} \leq \mathcal{L} \leq \mathcal{ALCQI}$, and</td>
<td>(even if $\mathcal{O}_1, \mathcal{O}_2$ are in $\mathcal{ALC}$).</td>
</tr>
</tbody>
</table>

### Consequences for modules of expressive DLs

Extracting modules is highly complex for expressive DLs.

**What to do?**

- Give up? No: modules clearly too important
- Reduce expressivity of logic? Yes! (Not covered here.)
- Approximate for expressive logics? Yes – but from the right direction!

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

- based on semantic locality
- based on syntactic locality

[Cuenca Grau et al. 2009]
Model-theoretic inseparability

Remember: \( O_1 \equiv^\Sigma O_2 \) if:
for all \( \eta \in \mathcal{L} \) with \( \text{sig}(\eta) \subseteq \Sigma \),
\[ O_1 \models \eta \text{ iff } O_2 \models \eta \]

Good news:
\[ \{ I \models \eta \models O_1 \} = \{ I \models \eta \models O_2 \} \]

i.e., \( O_1 \) and \( O_2 \) have the same models modulo \( \Sigma \)
\( (I|\Sigma \text{ is the restriction of } I \text{ to } \Sigma) \)

shorthand: \( O_1 \equiv^\Sigma_{mod} O_2 \) (model-inseparable)

this notion does not depend on \( \mathcal{L} \)

Bad news: \( O_1 \equiv^\Sigma_{mod} O_2 \) is undecidable already for \( \mathcal{ALC} \! \)!

Semantic locality

We can approximate model-inseparability, exploiting that \( \mathcal{M} \) is a subset of \( \mathcal{O} \)
\[ \mathcal{M} \equiv^\Sigma_{mod} \mathcal{O} \]
\[ \uparrow \]

every \( I \models \mathcal{M} \) can be extended to \( J \models \mathcal{O} \) with \( I|\Sigma = J|\Sigma \)
\[ \uparrow \]

every \( I \models \mathcal{M} \) can be extended to \( J \models \mathcal{O} \) with \( I|\Sigma = J|\Sigma \)
and \( \forall X \not\in \Sigma : X^J = \emptyset \)
\[ \uparrow \]

every \( \alpha \in \mathcal{O} \setminus \mathcal{M} \) is semantically local w.r.t. \( \Sigma \cup \text{sig}(\mathcal{M}) \):
\( \alpha \), with all terms not in \( \Sigma \cup \text{sig}(\mathcal{M}) \) replaced by \( \bot \), is a tautology

From semantic to syntactic locality

Semantic locality involves tautology check
\( \sim \) can be tested using a reasoner
\( \sim \) has the same complexity as standard reasoning

A syntactic approximation that can be tested in poly-time:

syntactic locality

(describes “obviously” sem. local axioms via a grammar)

Both notions lead to modules that are
- \( (\Sigma \cup \text{sig}(\mathcal{M})) \)-inseparable from \( \mathcal{O} \)
- not necessarily minimal

Module extraction with locality

Module extraction algorithm:

- \( \mathcal{M} \leftarrow \emptyset \)

- While \( \alpha \text{ not local w.r.t. } \Sigma \cup \text{sig}(\mathcal{M}) \), for some \( \alpha \in \mathcal{O} \setminus \mathcal{M} \),
  do \( \mathcal{M} \leftarrow \mathcal{M} \cup \{ \alpha \} \)

- Output \( \mathcal{M} \)

Variations:
- this notion: (semantic/syntactic) \( \bot \)-module
- dual notion: (semantic/syntactic) \( T \)-module
- smaller modules by nesting \( T \)- and \( \bot \)-module extraction:
  \( T\bot^* \)-modules
Summary locality-based modules

Locality-based modules . . .
- are “good approximations” of minimal modules because they guarantee \( \mathcal{M} \equiv_{\sum} \mathcal{O} \)
- are not necessarily minimal (but in practice often small enough)
- can be extracted in polynomial time (syntactic locality)
- are even self-contained:
  \[ \mathcal{M} \equiv_{\sum} \Sigma \cup \text{sig}(\mathcal{M}) \]
  and depleting:
  \[ \mathcal{O} \setminus \mathcal{M} \equiv_{\sum} \text{sig}(\mathcal{M}) \]
  and thus unique
- contain all justifications for any \( \alpha \) with \( \text{sig}(\alpha) \subseteq \Sigma \)

\( \sim \) Cheap is cheerful! :)
Appendix and References

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 collaboration scenario

Collaborative ontology development

- Developers work (edit, classify) locally
- Extra care at re-combination
- Prescriptive/analytic behaviour

This approach is mostly understood, but not implemented yet.

Understanding and/or structuring an ontology

Compute the modular structure of an ontology

1,000,000 axioms

This is work in progress.
Understanding and/or structuring an ontology

Compute the modular structure of an ontology

This is work in progress.

References

