Modularity in Ontologies: Module extraction: approaches, tools

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, 14 August 2013

Reminder of yesterday's lecture

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!

Today, we will discuss

- 2 approximations, i.e., sufficient conditions for safety based on semantic and syntactic locality
- MEX modules for a fragment of $\mathcal{EL}$
- tool support for module extraction
- the relation between these module notions

Plan for today

1. Locality and locality-based modules
2. Tool support
3. Summary and outlook

Thanks: Parts 1+2 based on slides by **Uli Sattler** and **Frank Wolter**.

And now . . .

1. Locality and locality-based modules
2. Tool support
3. Summary and outlook

Testing safety using locality

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

if

$\mathcal{O}$ is a model Σ-conservative 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 $X^\mathcal{J} = \emptyset$, $\forall X \notin \Sigma$

iff

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

iff

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

iff

$\forall \alpha \in \mathcal{O}$: $\underbrace{\text{“}\alpha \text{ with all } X \notin \Sigma \text{ replaced by } \bot\text{” is a tautology}}_{\alpha \text{ is } \emptyset\text{-local w.r.t. } \Sigma}$

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

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 $\top$
- If $C_1' \sqcap \neg C_2'$ is satisfiable % *can find countermodel*
  then return "probably not safe"

Return "safe"

Answers "safe" **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

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 $\bot$
    If $C_1' \sqcap \neg C_2'$ is satisfiable   % *can find countermodel*
        then return "probably not safe"
Return "safe"

Answers "safe" **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.
  (see all highlighted "**if**"s)
- We still need to perform **reasoning:**
  for each axiom α, test satisfiability of $C_1' \sqcap \neg C_2'$
  - Testing satisfiability in $\mathcal{ALC}$ is ExpTime-complete!
  - Testing satisfiability in $\mathcal{SROIQ}$ is N2ExpTime-complete!
  - There are highly optimised reasoners available,
    but optimised largely for **classification**.

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

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

Syntactic approximation of locality

- Define sets $\mathcal{C}^{\emptyset}, \mathcal{C}^{\Delta}$ of **$\bot$-equivalent** and **$\top$-equivalent** concepts:

if $A \notin \Sigma$,	then $A \in \mathcal{C}^{\emptyset}$		$\top \in \mathcal{C}^{\Delta}$
if $C \in \mathcal{C}^{\Delta}$,	then $\neg C \in \mathcal{C}^{\emptyset}$	if $C \in \mathcal{C}^{\emptyset}$,	then $\neg C \in \mathcal{C}^{\Delta}$
if $C \in \mathcal{C}^{\emptyset}$,	then $C \sqcap D \in \mathcal{C}^{\emptyset}$	if $C, D \in \mathcal{C}^{\Delta}$,	then $C \sqcap D \in \mathcal{C}^{\Delta}$
if $C \in \mathcal{C}^{\emptyset}$,	then $\exists r.C \in \mathcal{C}^{\emptyset}$		
if $r \notin \Sigma$,	then $\exists r.C \in \mathcal{C}^{\emptyset}$	(minimal rule set for $\mathcal{ALC}$)	

- Axiom $\alpha = (C \sqsubseteq D)$ is **syntactically Σ-local**
  if $C \in \mathcal{C}^{\emptyset}$ or $D \in \mathcal{C}^{\Delta}$

- Ontology $\mathcal{O}$ is **syntactically Σ-local** if all $\alpha \in \mathcal{O}$ are

Theorem

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

[Cuenca Grau et al. 2009]

Exercise: which of these axioms are syntactically local?

(A, B, C: atomic concepts; $\overline{X}$ means $X \in \Sigma$)

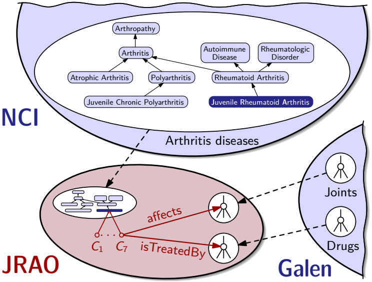
$\overline{B} \sqsubseteq A$	$B \notin \mathcal{C}^{\emptyset}$, $A \notin \mathcal{C}^{\Delta} \rightsquigarrow$ not $\{\overline{B}, \dots\}$-local
$A \sqsubseteq \overline{B} \sqcap \exists r.\overline{C}$	$A \in \mathcal{C}^{\emptyset} \rightsquigarrow \{\overline{B}, \overline{C}\}$-local
$X \sqcap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

Reminder

if $A \notin \Sigma$,	then $A \in \mathcal{C}^{\emptyset}$	$\top \in \mathcal{C}^{\Delta}$	
if $C \in \mathcal{C}^{\Delta}$,	then $\neg C \in \mathcal{C}^{\emptyset}$	if $C \in \mathcal{C}^{\emptyset}$,	then $\neg C \in \mathcal{C}^{\Delta}$
if $C \in \mathcal{C}^{\emptyset}$,	then $C \sqcap D \in \mathcal{C}^{\emptyset}$	if $C, D \in \mathcal{C}^{\Delta}$,	then $C \sqcap D \in \mathcal{C}^{\Delta}$
if $C \in \mathcal{C}^{\emptyset}$,	then $\exists r.C \in \mathcal{C}^{\emptyset}$		
if $r \notin \Sigma$,	then $\exists r.C \in \mathcal{C}^{\emptyset}$		

$\alpha = (C \sqsubseteq D)$ is **syntactically Σ-local** if $C \in \mathcal{C}^{\emptyset}$ or $D \in \mathcal{C}^{\Delta}$

Back to our real example



In *JRAO*, we can reuse

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

and "syntactically safely" write:

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

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

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

- What if we want to **generalise** terms?
  Then 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. $\mathcal{L}$.

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

Algorithm

Input: Sig. Σ, TBox $\mathcal{O}$

$\mathcal{M} \leftarrow \emptyset$, $\Sigma_+ \leftarrow \Sigma$

Repeat $\Sigma_{\text{prev}} \leftarrow \Sigma_+$

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

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

Until $\Sigma_{\text{prev}} = \Sigma_+$

Return $\mathcal{M}$

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

Example: see blackboard

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

Pitfall 1: what to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

(1) Your axiom might violate locality, but not safety.
(Remember: locality *approximates* safety.)

$\rightsquigarrow$ Call 0800-inseparability,
ask your favourite logician to decide whether the axiom is safe.

Pitfall 1: what to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

(2) Your axiom violates safety?
Do you have a good reason to write it?
If yes, keep it, but be aware that you've amended the topic!

Pitfall 1: what to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

(3) Want to repair a non-local axiom?

- Delete it.
- Modify it:

$$\text{Bird} \sqsubseteq \text{Flies} \quad \rightsquigarrow \quad \text{Bird} \sqcap \neg\text{Penguin} \sqsubseteq \text{Flies}$$

$$\text{Bird} \sqsubseteq \text{Flies} \quad \rightsquigarrow \quad \text{Bird} \sqsubseteq \text{Flies} \sqcup \text{Penguin}$$

- Explanations . . .

Pitfall 1: what to do if safety is violated?

Q: Help, my tool found a non-local axiom! What shall I do?

A: There are several possibilities:

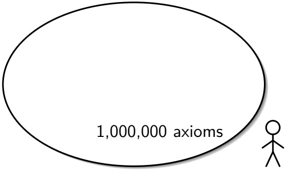
(4) Prescriptive/analytic safety checking . . .

Pitfall 2: independence

- Required property: If $\mathcal{O}_1$ is safe for Σ_2 and Σ_3, then $\mathcal{O}_1 \cup \mathcal{O}_2$ should be safe for Σ_3.

- Difficult to achieve prescriptively: only holds under restrictive preconditions
- Advice: treat independence analytically.

Pitfall 3: specifying the topic



Which terms do I want to import?

- Ask 0800-domainexpert for a list of terms.
- Browse through the class hierarchy and find suitable terms.
- Shopping for symbols:

  - Select terms.
  - Get a preview of the module.
  - If you're satisfied, check out the module.
- Prototype: Hancock, to be shown later

Summary: locality

- Safety and economy/coverage are important guarantees (not only) for reuse.
- They can be defined using inseparability.
- They can be approximated using locality.
- Modules based on syntactic locality can be extracted efficiently in logics up to OWL.
- Determining a signature for a module is still a non-trivial task.

And now . . .

1. Locality and locality-based modules
2. Tool support
3. Summary and outlook

Overview

What there is

- Command line tool for extracting MEX modules
  `http://cgi.csc.liv.ac.uk/~konev/software/`
- Java libraries for extracting locality-based mod.s in OWL API
  `http://owlapi.sourceforge.net/`
- ▶ Web module extractor for locality-based modules
  `http://owl.cs.manchester.ac.uk/modularity`
- ▶ Prototype of module extraction GUI: Hancock
  (not publicly available, but on ESSLLI Wiki soon)

What there isn't

- A Protégé plugin that fully supports the specification of the signature

And now . . .

Summary and outlook

- Safety and economy/coverage are important guarantees (not only) for reuse.
- Modules based on syntactic locality can be extracted efficiently in logics up to OWL, *and are often close to minimal.* ➺ *Thursday*
- Modules based on MEX can be extracted efficiently from acyclic $\mathcal{ELI}$ ontologies.
- There is tool support for extracting modules.
  `http://owl.cs.manchester.ac.uk/modularity`
  `http://owlapi.sourceforge.net/`
- Tool support for checking safety and determining seed signatures is still needed.

Course overview

4. Module extraction
    - MEX modules
    - Comparison

   Decomposing ontologies
    - Atomic decomposition
5. Related notions and recent advances
    - Forgetting and interpolation
    - Logical difference
    - Incremental/modular reasoning

Semantic vs. syntactic LBMs: affected ontologies (1)

Ontology	Abbreviation	DL expressivity	#axioms	#terms
MiniTambis-repaired	MiniT	$\mathcal{ALCN}$	170	226
Tambis-full	Tambis	$\mathcal{SHIN}(\mathcal{D})$	592	496
Bleeding History Phenotype	BHO	$\mathcal{ALCIF}(\mathcal{D})$	1,925	581
Neuro Behavior Ontology	NBO	$\mathcal{AL}$	1,314	970
Pharmacogenomic Relationsh...	PhaRe	$\mathcal{ALCHIF}(\mathcal{D})$	459	311
Terminological and Ontological...	TOK	$\mathcal{SRIQ}(\mathcal{D})$	466	330

Table 1. Ontologies that exhibit differences in modules

Semantic vs. syntactic LBMs: affected ontologies (2)

Ontol.	Types affected	#diffs	size of diffs		size of $\Delta\emptyset^*$-modules				culprit	
			#axs	(rel.)	T1 (%)		T2		type	
					range	avg.	range	avg.	+ freq.	
miniT	bot, nested	14–25%	1–7	0–600%[b]	48–79	66	0–8	2	*c*	3
Tambis	bot, nested	32–57%	2–41[c]	1–62%[c]	75–88	82	0–34	9	*c*	8
BHO[a]	nested	17%	1–12	0–300%	55–72	65	0–31	4	*b*	31
NBO[a]	nested	3%	2	0–200%	64–78	71	0–3	0	*d*	3
PhaRe[a]	top, nested	1–8%	1–326[d]	0–6,520%[d]	50–70	60	0–8	1	*d*	10
TOK	top, nested	49–100%	1–7	0–9%	48–68	59	9–17	10	*d*	3

[a]differences only for genuine modules

[b]differences > 5% only for genuine modules

[c]differences > 11 axioms (> 2%) only for genuine modules

[d]differences > 13 axioms (> 1,300%) only for top-modules

Semantic vs. syntactic LBMs: checking Δ-locality

Δ-modules cannot always be extracted using DL reasoners:

- Remember – locality check: replace non-Σ symbols with $\top$ and test for tautology
- Global restrictions of $\mathcal{SROIQ}$ don't allow $\top$-role in number restrictions or role chains
- This affects some 40 ontologies in our corpus