Working Modularly with Ontologies

Chiara Del Vescovo  Bijan Parsia  Uli Sattler  Thomas Schneider

School of Computer Science, University of Manchester, UK

21 April 2010
And now ...

1. Ontologies and Description Logic
2. Why modularity?
3. A reuse scenario
4. Understanding ontologies via modules
Ontology

= collection of statements about a domain \textit{(axioms)}

- Language used: usually logic, often \textit{description logic (DL)}
- \textit{Inferences} can be drawn from axioms

Domains:
biology, medicine, chemistry, business processes, natural language, ...
Example axioms + inferences

\[ \text{Arm} \sqsubseteq \exists \text{hasPart} \cdot \text{Hand} \]

\[ \forall x (\text{Arm}(x) \rightarrow \exists y (\text{hasPart}(x,y) \land \text{Hand}(y))) \]
Example axioms + inferences

\[ \text{Arm} \sqsubseteq \exists \text{hasPart}. \text{Hand} \]

\[ \forall x \left( \text{Arm}(x) \rightarrow \exists y \left( \text{hasPart}(x, y) \land \text{Hand}(y) \right) \right) \]

\[ \text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \]

\[ \forall x \left( \text{Limb}(x) \leftrightarrow \left( \text{Arm}(x) \lor \text{Leg}(x) \right) \right) \]
Example axioms + inferences

\[
\text{Arm} \subseteq \exists \text{hasPart}. \text{Hand} \quad \forall x \left( \text{Arm}(x) \rightarrow \exists y \left( \text{hasPart}(x, y) \land \text{Hand}(y) \right) \right)
\]

\[
\text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \quad \forall x \left( \text{Limb}(x) \leftrightarrow \left( \text{Arm}(x) \lor \text{Leg}(x) \right) \right)
\]

\[
\models \text{Limb} \cap \neg \text{Leg} \subseteq \exists \text{hasPart}. \text{Hand} \quad \forall x \left( \left( \text{Limb}(x) \land \neg \text{Leg}(x) \right) \rightarrow \exists y \left( \text{hasPart}(x, y) \land \text{Hand}(y) \right) \right)
\]
Example axioms + inferences

\[ \text{Arm} \sqsubseteq \exists \text{hasPart} \cdot \text{Hand} \]

\[ \forall x \left( \text{Arm}(x) \rightarrow \exists y \left( \text{hasPart}(x,y) \land \text{Hand}(y) \right) \right) \]

\[ \text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \]

\[ \forall x \left( \text{Limb}(x) \leftrightarrow (\text{Arm}(x) \lor \text{Leg}(x)) \right) \]
Example axioms + inferences

\[
\text{Arm} \subseteq \exists \text{hasPart} . \text{Hand}
\]

\[
\forall x \left( \text{Arm}(x) \rightarrow \exists y \left( \text{hasPart}(x,y) \land \text{Hand}(y) \right) \right)
\]

\[
\text{Limb} \equiv \text{Arm} \sqcup \text{Leg}
\]

\[
\forall x \left( \text{Limb}(x) \leftrightarrow \left( \text{Arm}(x) \lor \text{Leg}(x) \right) \right)
\]

\[
\text{LeftArm} : \text{Arm}
\]

\[
\text{Arm(LeftArm)}
\]
Example axioms + inferences

\[ \text{Arm} \sqsubseteq \exists \text{hasPart}. \text{Hand} \]
\[ \forall x (\text{Arm}(x) \rightarrow \exists y (\text{hasPart}(x,y) \land \text{Hand}(y))) \]

\[ \text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \]
\[ \forall x (\text{Limb}(x) \leftrightarrow (\text{Arm}(x) \lor \text{Leg}(x))) \]

\[ \text{LeftArm} : \text{Arm} \]

\[ \models \text{LeftArm} : \exists \text{hasPart}. \text{Hand} \]
\[ \exists y (\text{hasPart}(\text{LeftArm}, y) \land \text{Hand}(y)) \]
Reasoning tasks

- **Consistency:**
  Does ontology $\mathcal{O}$ have a model?

- **Satisfiability:**
  Is there a model of $\mathcal{O}$ that interprets concept $C$ as nonempty?

- **Subsumption:**
  Does $C \sqsubseteq D$ hold in every model of $\mathcal{O}$?

- **Instance checking:**
  Is individual $x$ an instance of $C$ in every model of $\mathcal{O}$?

Inter-reducible; optimised reasoners available
The Web Ontology Language OWL

- W3C-recommended standard since 2004
The Web Ontology Language OWL

- W3C-recommended standard since 2004

**OWL 2**

Based on DL **SROIQ**

∃, ∀, counting, role chains and hierarchies, transitivity, inverse roles, nominals

**OWL 2 EL, QL, RL**

Sub-profiles for efficient reasoning and application orientation
And now ...
Three scenarios

Import/reuse

Collaboration

Understanding
Three scenarios

- **Import/reuse**
- **Collaboration**
- **Understanding**
And now ...
A reuse scenario

Import/reuse one external ontology
A reuse scenario

Import/reuse one external ontology

NCI knowledge about “Disease” and “Arthritis”

JRA Ontology
A reuse scenario

Import/reuse one external ontology

NCI

knowledge about “Disease” and “Arthritis”

JRA Ontology

How much of NCI do we need?
A reuse scenario

Import/reuse a part of an external ontology

Disease, Arthritis

NCI

JRA Ontology

How much of NCI do we need?

- **Coverage:** Import *everything* relevant for the chosen terms.
- **Economy:** Import *only* what’s relevant for them. Compute that part quickly.
A reuse scenario

Import/reuse parts of several external ontologies

NCI

Disease, Arthritis

JRA Ontology

Drug, Joint

Galen
A reuse scenario

Ontologies+DL Why modularity? Reuse Understanding ontologies

Arthritis diseases

NCI

JRAO

Galen

Arthropathy

Arthritis

Autoimmune Disease

Rheumatologic Disorder

Atrophic Arthritis

Polyarthritis

Rheumatoid Arthritis

Juvenile Chronic Polyarthritis

Juvenile Rheumatoid Arthritis

C1 C7 affects

isTreatedBy

Joints

Drugs
A working cycle

1. Edit your ontology
2. Import a module
A working cycle

- Edit your ontology $O$
- Load an external ontology $E$
- $T \leftarrow \text{Specify terms from } E$
- $M \leftarrow \text{mod}(T, E)$
- $O \leftarrow O \cup M$
A working cycle

Edit your ontology $O$

Load an external ontology $E$

$T \leftarrow$ Specify terms from $E$

$M \leftarrow \text{mod}(T, E)$

Module Coverage

$O \leftarrow O \cup M$
Module coverage

**Goal:** Import everything the external ontology knows about the topic that consists of the specified terms.
Module coverage

**Goal:** Import everything the external ontology knows about the topic that consists of the specified terms.

**Example 1:**
- **Topic:** Arm, Hand, hasPart
- **On-topic:**
  - Arm ⊑ ∀ hasPart. Hand
  - Arm ⊔ Hand ⊑ ∃ hasPart. T
  - Hand ⊑ ¬ Arm
  - Hand ⊑ Hand ⊔ Arm
- **Off-topic:**
  - HandWith4Fingers ⊑ Hand

**Goal =** preserve all on-topic knowledge
Module coverage

**Goal:** Import everything the external ontology knows about the topic that consists of the specified terms.

**Question:** Which axioms do we need to import?
Module coverage

**Goal:** Import everything the external ontology knows about the topic that consists of the specified terms.

**Question:** Which axioms do we need to import?

**Example 2:**

\[
\text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \\
\text{Leg} \equiv \text{LLeg} \sqcup \text{RLeg} \\
\text{LLeg} \sqsubseteq \exists \text{hasPart} \cdot \text{LBToe} \\
\text{RLeg} \sqsubseteq \exists \text{hasPart} \cdot \text{RBToe} \\
\text{Arm} \sqsubseteq \exists \text{hasPart} \cdot \text{Finger}
\]

\[
\text{O} \cup \text{Medical} \\
\models \\
\text{Limb} \sqsubseteq \exists \text{hasPart} \cdot \top
\]
**Goal:** Import everything the external ontology knows about the topic that consists of the specified terms.

**Question:** Which axioms do we need to import?

**Example 2:**

\[
\text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \\
\text{Leg} \equiv \text{LLeg} \sqcup \text{RLeg} \\
\text{LLeg} \sqsubseteq \exists \text{hasPart}.\text{LBToe} \\
\text{RLeg} \sqsubseteq \exists \text{hasPart}.\text{RBToe} \\
\text{Arm} \sqsubseteq \exists \text{hasPart}.\text{Finger}
\]

\[
\text{O} \cup \text{Medical}_1 \\
\notin \\
\text{Limb} \sqsubseteq \exists \text{hasPart}.\top
\]
Module coverage

Goal: Import everything the external ontology knows about the topic that consists of the specified terms.

Question: Which axioms do we need to import?

Example 2:

\[
\text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \\
\text{Leg} \equiv \text{LLeg} \sqcup \text{RLeg} \\
\text{LLeg} \sqsubseteq \exists \text{hasPart}.\text{LBToe} \\
\text{RLeg} \sqsubseteq \exists \text{hasPart}.\text{RBToe} \\
\text{Arm} \sqsubseteq \exists \text{hasPart}.\text{Finger} \
\]

\[O \cup \text{Medical}_2 \]

\[\not\subseteq\]

\[\text{Limb} \sqsubseteq \exists \text{hasPart}.\top\]
Module coverage

**Goal:** Import everything the external ontology knows about the topic that consists of the specified terms.

**Question:** Which axioms do we need to import?

**Example 2:**

\[
\text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \\
\text{Leg} \equiv \text{LLeg} \sqcup \text{RLeg} \\
\text{LLeg} \sqsubseteq \exists \text{hasPart}.\text{LBToe} \\
\text{RLeg} \sqsubseteq \exists \text{hasPart}.\text{RBToe} \\
\text{Arm} \sqsubseteq \exists \text{hasPart}.\text{Finger} \\
\text{O} \cup \text{Medical}_3 \\
\text{⊬} \\
\text{Limb} \sqsubseteq \exists \text{hasPart}.\top
\]
Module coverage

Goal: Import everything the external ontology knows about the topic that consists of the specified terms.

Question: Which axioms do we need to import?

Example 2:

\[ \text{Limb} \equiv \text{Arm} \sqcup \text{Leg} \]
\[ \text{Leg} \equiv \text{LLeg} \sqcup \text{RLeg} \]
\[ \text{LLeg} \sqsubseteq \exists \text{hasPart.LBToe} \]
\[ \text{RLeg} \sqsubseteq \exists \text{hasPart.RBToe} \]
\[ \text{Arm} \sqsubseteq \exists \text{hasPart.Finger} \]

\[ \text{O} \cup \text{Medical}_4 \]
\[ \models \]
\[ \text{Limb} \sqsubseteq \exists \text{hasPart.} \top \]
Module coverage

Module $\mathcal{E}'$ covers ontology $\mathcal{E}$ for the specified topic $\mathcal{T}$ if for all concepts $C, D$ built from terms in $\mathcal{T}$:

if 
\[ O \cup \mathcal{E} \models C \sqsubseteq D, \]
then 
\[ O \cup \mathcal{E}' \models C \sqsubseteq D. \]
Module coverage

- Module $\mathcal{E}'$ covers ontology $\mathcal{E}$ for the specified topic $\mathcal{T}$ if for all concepts $C, D$ built from terms in $\mathcal{T}$:
  
  if $\mathcal{O} \cup \mathcal{E} \models C \sqsubseteq D$, 
  then $\mathcal{O} \cup \mathcal{E}' \models C \sqsubseteq D$.

- Coverage $\hat{=} \,$ preserving entailments

- $\mathcal{O} \cup \mathcal{E}$ is called *conservative extension (CE)* of $\mathcal{O} \cup \mathcal{E}'$
Module coverage

- Module $\mathcal{E}'$ covers ontology $\mathcal{E}$ for the specified topic $\mathcal{T}$ if for all concepts $C, D$ built from terms in $\mathcal{T}$:
  
  if $\mathcal{O} \cup \mathcal{E} \models C \sqsubseteq D$, then $\mathcal{O} \cup \mathcal{E}' \models C \sqsubseteq D$.

- Coverage $\models$ preserving entailments

- $\mathcal{O} \cup \mathcal{E}$ is called *conservative extension (CE)* of $\mathcal{O} \cup \mathcal{E}'$

- No coverage $\leadsto$ no encapsulation $\leadsto$ no *module*

- With coverage: trade-off minimality $\leftrightarrow$ computation time
Module coverage

- Module $\mathcal{E}'$ covers ontology $\mathcal{E}$ for the specified topic $\mathcal{T}$ if for all concepts $C, D$ built from terms in $\mathcal{T}$:
  
  \[
  \text{if } \big( \mathcal{O} \cup \mathcal{E} \models C \sqsubseteq D, \big) \text{, then } \big( \mathcal{O} \cup \mathcal{E}' \models C \sqsubseteq D. \big)
  \]

- Coverage $\models$ preserving entailments

- $\mathcal{O} \cup \mathcal{E}$ is called conservative extension (CE) of $\mathcal{O} \cup \mathcal{E}'$

- Minimal covering modules via CEs
- CEs hard to impossible to decide
- Tractable approximation: syntactic locality
A working cycle

Edit your ontology $\mathcal{O}$

Load an external ontology $\mathcal{E}$

$\mathcal{T} \leftarrow$ Specify terms from $\mathcal{E}$

$\mathcal{M} \leftarrow \text{mod}(\mathcal{T}, \mathcal{E})$

$\mathcal{O} \leftarrow \mathcal{O} \cup \mathcal{M}$

Safety

Module Coverage
Safety

Goal: Don’t change the meaning of imported terms.
     = Don’t add new knowledge about the imported topic.

Question: Which axioms are we allowed to write?
Safety

**Goal:** Don’t change the meaning of imported terms.

= Don’t add new knowledge about the imported topic.

**Question:** Which axioms are we allowed to write?

**Example:**

- Nose $\sqsubseteq \neg$ Lateral
- Nose $\sqsubseteq$ Organ
- Med
- Organ $\sqsubseteq$ Lateral
- O
Safety

**Goal:** Don’t change the meaning of imported terms. 
= Don’t add new knowledge about the imported topic.

**Question:** Which axioms are we allowed to write?

**Example:**

\[
\text{Nose} \sqsubseteq \neg \text{Lateral} \\
\text{Nose} \sqsubseteq \text{Organ} \\
\text{Med} \\
\text{Organ} \sqsubseteq \text{Lateral}
\]
**Safety**

**Goal:** Don’t change the meaning of imported terms.  
≡ Don’t add new knowledge about the imported topic.

**Question:** Which axioms are we allowed to write?

**Example:**

- \( \text{Nose} \sqsubseteq \lnot \text{Lateral} \)
- \( \text{Nose} \sqsubseteq \text{Organ} \)

\( \text{O} \cup \text{Med} \models \text{Organ} \sqsubseteq \text{Lateral} \)

but  \( \text{Med} \nmid \models \text{Organ} \sqsubseteq \text{Lateral} \)
Safety

- *Our* ontology $\mathcal{O}$ uses the imported terms safely if for all concepts $C, D$ built from the imported terms:

  if $\mathcal{E}' \notin C \sqsubseteq D$,

  then $\mathcal{O} \cup \mathcal{E}' \notin C \sqsubseteq D$,

- Safety $\hat{=} \text{preserving non-entailments}$
## Comparison of different approaches

<table>
<thead>
<tr>
<th>Kind of “module”</th>
<th>Covrg.</th>
<th>Min.</th>
<th>Covered DLs</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All ax’s referencing $\mathcal{T}$</td>
<td>✗</td>
<td>any</td>
<td>any</td>
<td>easy</td>
</tr>
<tr>
<td>Seidenberg/Rector</td>
<td>✗</td>
<td>any</td>
<td>any</td>
<td>easy</td>
</tr>
<tr>
<td>Prompt</td>
<td>✗</td>
<td>?</td>
<td></td>
<td>easy</td>
</tr>
</tbody>
</table>
## Comparison of different approaches

<table>
<thead>
<tr>
<th>Kind of “module”</th>
<th>Covrg.</th>
<th>Min.</th>
<th>Covered DLs</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All ax’s referencing $\mathcal{T}$</td>
<td>✗</td>
<td></td>
<td>any</td>
<td>easy</td>
</tr>
<tr>
<td>Seidenberg/Rector</td>
<td>✗</td>
<td></td>
<td>any</td>
<td>easy</td>
</tr>
<tr>
<td>Prompt</td>
<td>✗</td>
<td>?</td>
<td></td>
<td>easy</td>
</tr>
<tr>
<td>The whole ontology</td>
<td>✓</td>
<td>✗✗</td>
<td>any</td>
<td>easy</td>
</tr>
<tr>
<td>conserv.-based mod.</td>
<td>✓</td>
<td>✓</td>
<td>few</td>
<td>hard</td>
</tr>
<tr>
<td>MEX (Liverpool)</td>
<td>✓</td>
<td>✓</td>
<td>acyclic $\mathcal{EL}$</td>
<td>easy</td>
</tr>
<tr>
<td>locality-based mod.</td>
<td>✓</td>
<td>✗</td>
<td>$\approx$ OWL 2</td>
<td>easy</td>
</tr>
<tr>
<td>E-connections</td>
<td>✓</td>
<td>✗</td>
<td>OWL 1</td>
<td>easy</td>
</tr>
</tbody>
</table>
## Comparison of different approaches

<table>
<thead>
<tr>
<th>Kind of “module”</th>
<th>Covrg.</th>
<th>Min.</th>
<th>Covered DLs</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All ax’s referencing $\mathcal{T}$</td>
<td>✗</td>
<td></td>
<td>any</td>
<td>easy</td>
</tr>
<tr>
<td>Seidenberg/Rector</td>
<td>✗</td>
<td></td>
<td>any</td>
<td>easy</td>
</tr>
<tr>
<td>Prompt</td>
<td>✗</td>
<td></td>
<td>?</td>
<td>easy</td>
</tr>
<tr>
<td>The whole ontology</td>
<td>✔</td>
<td>✔</td>
<td>any</td>
<td>easy</td>
</tr>
<tr>
<td>conserv.-based mod.</td>
<td>✔</td>
<td>✔</td>
<td>few</td>
<td>hard</td>
</tr>
<tr>
<td>MEX (Liverpool)</td>
<td>✔</td>
<td>✔</td>
<td>acyclic $\mathcal{EL}$</td>
<td>easy</td>
</tr>
<tr>
<td>locality-based mod.</td>
<td>✔</td>
<td>✗</td>
<td>$\approx$ OWL 2</td>
<td>easy</td>
</tr>
<tr>
<td>E-connections</td>
<td>✔</td>
<td>✗</td>
<td>OWL 1</td>
<td>easy</td>
</tr>
<tr>
<td>interpolants-based</td>
<td>✔</td>
<td>✔</td>
<td>few</td>
<td>hard</td>
</tr>
<tr>
<td>(no subsets!)</td>
<td>✔</td>
<td>✔</td>
<td>few</td>
<td>hard</td>
</tr>
</tbody>
</table>
Module extraction in Protégé 4

Nightly build:

http://owl.cs.manchester.ac.uk/2008/iswc-mo...
Web service for module extraction

http://owl.cs.manchester.ac.uk/modularity

**OWL Module Extractor**

**Ontology source**
Paste your ontology, or enter a URL of a document, into the text box below.

```
http://www.co-ode.org/ontologies/pizza/pizza.owl
```

**Signature**
Enter a signature. Put each entity name on a new line. (Accepts full URLs or URL fragments)

```
Pizza
```

**Modularity type**
Select the module type

- Top (lower) module
- Bottom (upper) module
- Bottom-of-top (upper-of-lower) module
- Top-of-bottom (lower-of-upper) module

![Show axioms view](image)

**Module: http://www.co-ode.org/ontologies/pizza/pizza.owl_module.owl**

**Selected signature**

```
Pizza (http://www.co-ode.org/ontologies/pizza/pizza.owl#Pizza)
```

**Module metrics**
Number of axioms: 112
Number of logical axioms: 112
Number of classes: 35
Number of object properties: 7
Number of data properties: 0
Number of individuals: 5

**Module axioms**
- CheeseTopping SubClassOf PizzaTopping
- CheeseTopping DisjointWith FishTopping
- CheeseTopping DisjointWith FruitTopping
- CheeseTopping DisjointWith HerbSpiceTopping
- CheeseTopping DisjointWith MeatTopping
- CheeseTopping DisjointWith NutTopping
- CheeseTopping DisjointWith SauceTopping
- CheeseTopping DisjointWith VegetableTopping
- CheeseyPizza EquivalentTo Pizza and (hasTopping some CheeseTopping)
- Country EquivalentTo DomainConcept and ((America, England, France, Germany, Italy))
- DeepPenBase SubClassOf PizzBase
- DeepPenBase DisjointWith ThinWalledCrispyBase
- DomainConcept DisjointWith ValuePartition
- FishTopping SubClassOf PizzaTopping
- FishTopping SubClassOf hasSpiciness some Mild
- FruitTopping DisjointWith FruitTopping
- FruitTopping DisjointWith HerbSpiceTopping
- FruitTopping DisjointWith MeatTopping
- FruitTopping DisjointWith NutTopping
- FruitTopping DisjointWith SauceTopping
- FruitTopping DisjointWith VegetableTopping
- Food SubClassOf DomainConcept
- FruitTopping SubClassOf PizzaTopping
- FruitTopping DisjointWith HerbSpiceTopping
- FruitTopping DisjointWith MeatTopping
- FruitTopping DisjointWith NutTopping
- FruitTopping DisjointWith SauceTopping
- FruitTopping DisjointWith VegetableTopping
- HerbSpiceTopping SubClassOf PizzaTopping

![Extract module](image)
And now ...

1. Ontologies and Description Logic
2. Why modularity?
3. A reuse scenario
4. Understanding ontologies via modules
Scenario: Understanding

Visualise the modular structure of an ontology
Scenario: Understanding

Visualise the modular structure of an ontology

We’re working on it.
We bet Robert Stevens . . .

- Ontology about periodic table of the chemical elements
- What is its modular structure?
- What is “the meat” of it?
- We can find it using locality-based modules.
Impetus for the “Meat” idea

Partition of koala.owl via E-connections in Swoop

- **importing part**
- **imported but non-importing part**
- **isolated part**

“imports vocabulary from”
Partition for the periodic table ontology

- Importing part
- Imported but non-importing part
- Isolated part

→ “imports vocabulary from”
“Meat” via locality-based modules

Hopes:

- Fine-grained analysis
- Guidance for users to choose the right topic(s)
- Draw conclusions on characteristics of an ontology: topicality, connectedness, axiomatic richness, superfluous parts, modelling
“Meat” via locality-based modules

Problem:

- Ontologies of size $n$ can have up to $2^n$ modules
- But do real-life ontologies fall into the worst case?
## Results so far

- Optimised algorithm to extract all modules

<table>
<thead>
<tr>
<th>Ontology</th>
<th>#Ax</th>
<th>#Terms</th>
<th>#mods</th>
<th>Theor. Max.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Koala</td>
<td>42</td>
<td>25</td>
<td>3660</td>
<td>33 554 432</td>
<td>9s</td>
</tr>
<tr>
<td>Mereology</td>
<td>44</td>
<td>25</td>
<td>1952</td>
<td>33 554 432</td>
<td>3min</td>
</tr>
</tbody>
</table>
Results so far

- Optimised algorithm to extract all modules

<table>
<thead>
<tr>
<th>Ontology</th>
<th>#Ax</th>
<th>#Terms</th>
<th>#mods</th>
<th>Theor. Max.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Koala</td>
<td>42</td>
<td>25</td>
<td>3660</td>
<td>33 554 432</td>
<td>9s</td>
</tr>
<tr>
<td>Mereology</td>
<td>44</td>
<td>25</td>
<td>1952</td>
<td>33 554 432</td>
<td>3min</td>
</tr>
</tbody>
</table>

- Single module numbers don’t say much
- Not scalable
Subset sampling

- Modularised randomly generated parts of 8 ontologies
- Example growth of module numbers:

  Trendline equation: $y = O(1.5^x)$, confidence 0.96
Weight analysis

- Ordered all 3660 modules of Koala by weight

\[
\text{Weight}(\mathcal{M}) = \text{PullingPower}(\mathcal{M}) \cdot \text{Cohesion}(\mathcal{M})
\]

- Inspected heaviest modules

- How many terms are needed to “pull” all the terms into \(\mathcal{M}\)?
- How strongly are the terms in \(\mathcal{M}\) held together?
Weight analysis

Figure 1: Structure of Koala ontology

Ontologies+DL

Why modularity?

Reuse

Understanding ontologies

1, 10 Quokka

20, 37 KoalaWithPhD, hasDegree

7, 15, 25, 34, 35 Koala

4, 12, 16, 19 isHardWorking, University

3, 6, 29 Forest

9, 17, 21, 31, 39 Degree

11 Habitat

27 TasmanianDevil

2, 18, 28 Gender

14, 22, 26, 32, 33 hasChildren, Parent

8, 13, 23, 30, 38, 40, 41 Male, Female, hasHabitat, Animal, hasGender
Outlook

- Find heaviest modules without computing all modules
- How many modules can ontologies have?
- Relation module number ↔ justificatory structure
Outlook

- Find heaviest modules without computing all modules
- How many modules can ontologies have?
- Relation module number $\leftrightarrow$ justificatory structure
- Collaborative ontology development using modules
- Complete the support for incremental reasoning
- Modules that are no subsets
- Modularity for belief revision
- Decidability of conservativity for FOL fragments
Outlook

- Find heaviest modules without computing all modules
- How many modules can ontologies have?
- Relation module number ↔ justificatory structure
- Collaborative ontology development using modules
- Complete the support for incremental reasoning
- Modules that are no subsets
- Modularity for belief revision
- Decidability of conservativity for FOL fragments

Thank you.