How Modular Are Modular Ontologies?

Logic-Based Metrics for Ontologies with Imports

CC-BY-2.0 kennymatic Flickr

Robin Nolte and *Thomas Schneider* University of Bremen DL 2019, Oslo

Modularity via imports

Large ontologies with 100,000s of axioms

e.g. SNOMED CT The global language of healthcare — National Cancer Institute

… are often built modularly, using imports

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

NCBO BioPortal

e.g., out of the 438 ontologies in the 2017 snapshot of BioPortal,

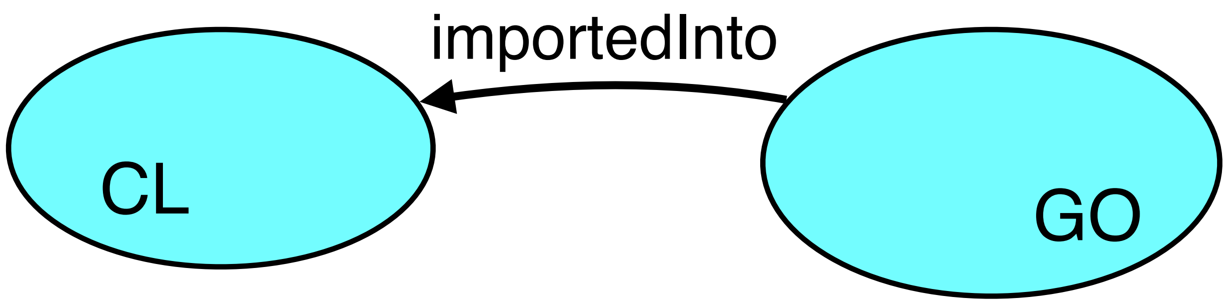
69 use imports;

some import up to 31 ontologies (directly & indirectly)

e.g., Cell Ontology (CL) imports 8 ontologies,

including the Gene Ontology (GO)

Modularity via imports

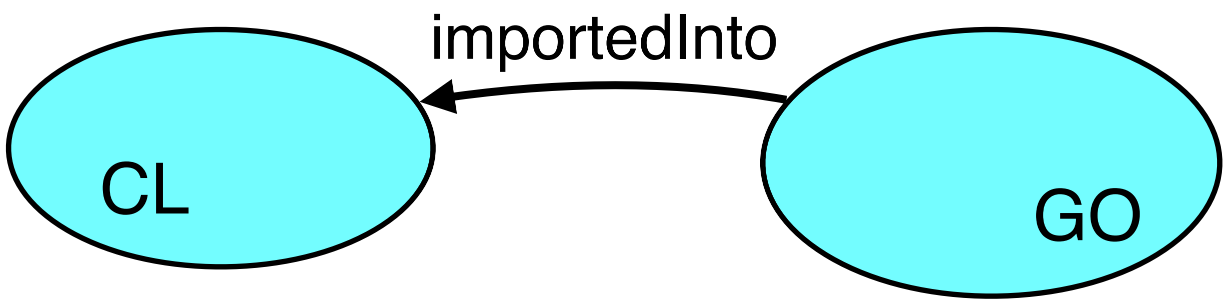


Import structures provide …

✓ Separation of concerns

Import structure helps separate (sub-)domains of interest

Modularity via imports ?

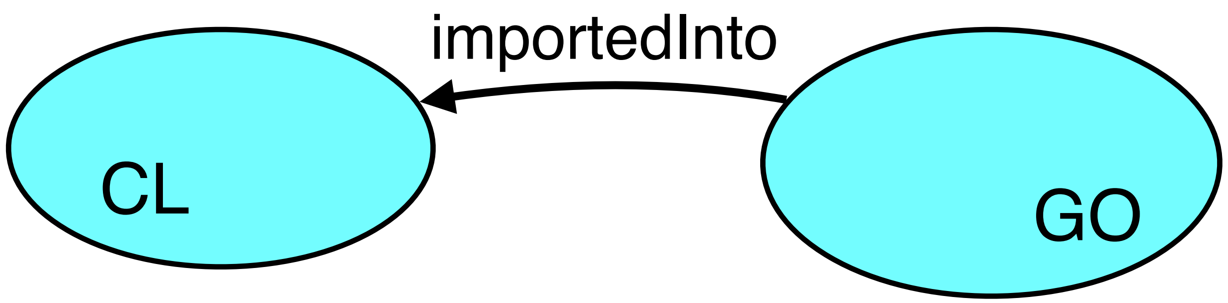


Import structures provide …

✓ Separation of concerns

Import structure helps separate (sub-)domains of interest

Modularity via imports ?



Import structures provide …

✓ Separation of concerns

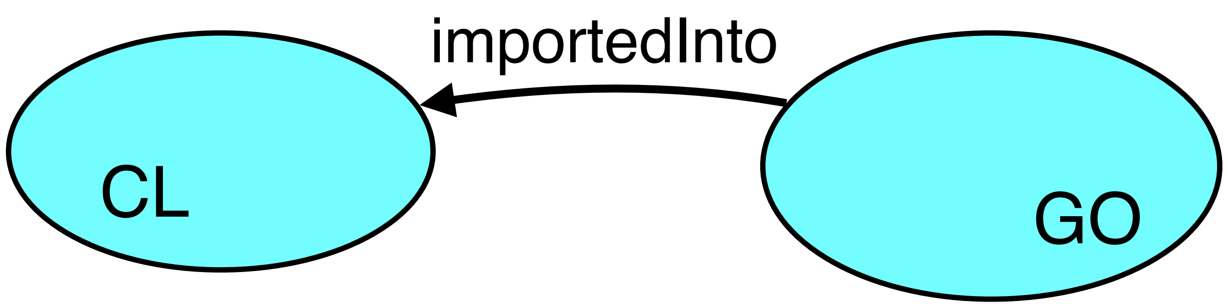
Import structure helps separate (sub-)domains of interest

✗ No logical guarantees

GO does not need to be a module of CL in a strict logical sense, i.e., it does not provide guarantees such as:

- $\forall\alpha$ with sig$(\alpha) \subseteq$ sig(GO): CL $\cup$ GO $\models \alpha$ iff GO $\models \alpha$
  (local completeness)

Modularity via imports ?



Import structures provide …

✓ Separation of concerns

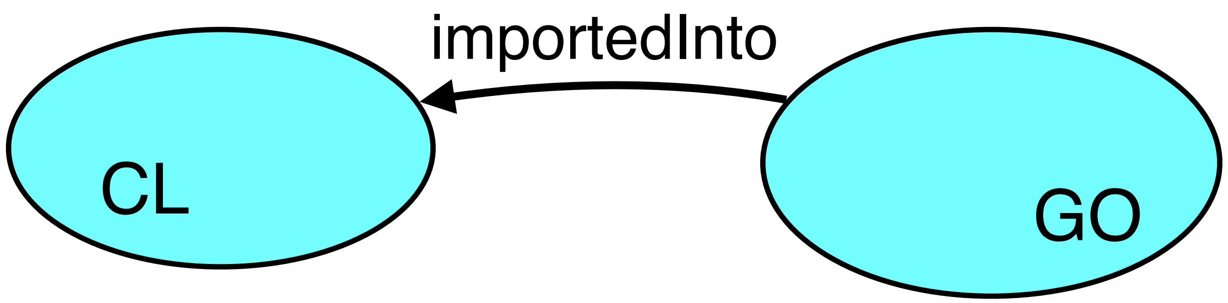
Import structure helps separate (sub-)domains of interest

✗ No logical guarantees

GO does not need to be a module of CL in a strict logical sense, i.e., it does not provide guarantees such as:

- $\forall\alpha$ with sig$(\alpha) \subseteq$ sig(GO): CL $\cup$ GO $\models \alpha$ iff GO $\models \alpha$
  (local completeness)
- $\exists\alpha$ with sig$(\alpha) \subseteq$ sig(CL): CL $\cup$ GO $\models \alpha$ & CL $\not\models \alpha$
  (relevance)

Logical guarantees and inseparability

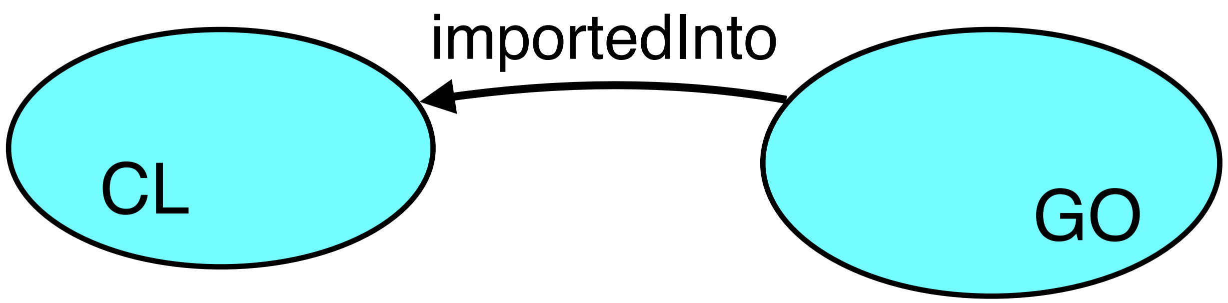


Local completeness:

$$\forall\alpha \text{ with } \mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathsf{GO}): \quad \mathsf{CL} \cup \mathsf{GO} \models \alpha \quad \text{iff} \quad \mathsf{GO} \models \alpha$$

In other words: CL ∪ GO is sig(GO)-inseparable from GO,
written $\mathsf{CL} \cup \mathsf{GO} \equiv_{\mathsf{sig}(\mathsf{GO})} \mathsf{GO}$

Logical guarantees and inseparability



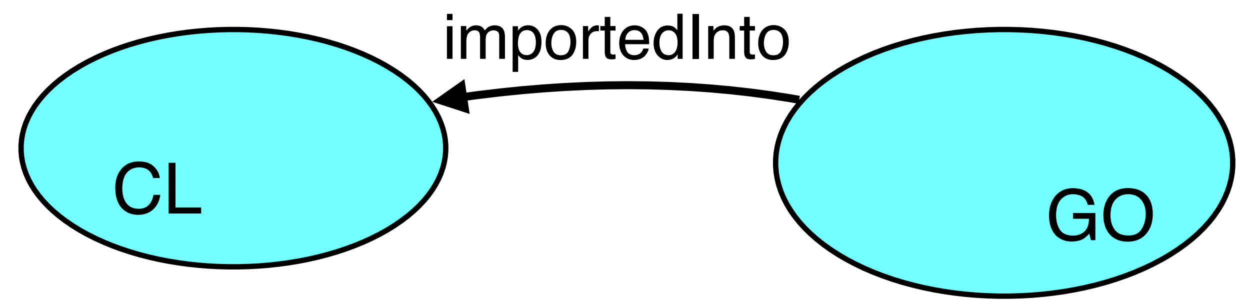
Local completeness:

$\forall\alpha$ with sig$(\alpha) \subseteq$ sig(GO): CL $\cup$ GO $\models \alpha$ iff GO $\models \alpha$

In other words: CL $\cup$ GO is sig(GO)-inseparable from GO,
written CL $\cup$ GO $\equiv_{\text{sig(GO)}}$ GO

Alas, inseparability is undecidable for many DLs above $\mathcal{ALC}$ [Lutz et al. 2007]

Logical guarantees and inseparability



Local completeness:

$$\forall \alpha \text{ with } \mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathsf{GO}): \quad \mathsf{CL} \cup \mathsf{GO} \models \alpha \quad \text{iff} \quad \mathsf{GO} \models \alpha$$

In other words: CL ∪ GO is sig(GO)-inseparable from GO,
written $\mathsf{CL} \cup \mathsf{GO} \equiv_{\mathsf{sig}(\mathsf{GO})} \mathsf{GO}$

Alas, inseparability is undecidable for many DLs above $\mathcal{ALC}$ [Lutz et al. 2007]

–> To measure local completeness, approximations are required:

- via locality [Cuenca Grau et al. 2007]
- via related module notions (locality-based etc.)

Both kinds of approximations provide sufficient conditions for local compl.

Our goal

We want to provide quantitative measures that …

- determine the extent to which imports in existing ontologies meet logical guarantees
- capture even stronger versions of these guarantees (i.e., relative to the other ontologies in the import closure)
- do not depend on a particular approximation (e.g. locality) or module notion

Main idea

- Consider the given import structure as a directed graph
- Compute a "reference graph" using some module notion that provides the logical guarantees
- Measure the similarity between both graphs

Import structures as graphs

Ographs

- are directed graphs capturing the import structure of a single ontology or a repository
- nodes = ontologies;   edges = "imported into" relation

e.g.

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

(i.e., $\mathcal{O}_1$ imports all other ontologies (in)directly)

Inseparability and modules

Consider arbitrary inseparability relation $\equiv_\Sigma$

and module notion $\text{mod}(\Sigma, \mathcal{O})$ with the following properties

- $\text{mod}(\Sigma, \mathcal{O}) \subseteq \mathcal{O}$ (uniquely determined)
- $\text{mod}(\Sigma, \mathcal{O}) \equiv_\Sigma \mathcal{O}$

$\text{mod}(\Sigma, \mathcal{O})$ is not necessarily minimal with these properties.

such as

- locality-based modules
- (A)MEX modules
- reachability-based modules
- datalog-based modules
- etc.

Previous example ograph:

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

(i.e., $\mathcal{O}_1$ imports all other ontologies (in)directly)

(1) Import of $\mathcal{O}_3$ into $\mathcal{O}_2$ is "safe" if $\mathcal{O}_3$ is locally complete w.r.t. $\mathcal{O}_2$, i.e., $\mathcal{O}_2 \cup \mathcal{O}_3 \equiv_{\mathsf{sig}(\mathcal{O}_3)} \mathcal{O}_3$

“Safe” imports

Previous example ograph:

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

(i.e., $\mathcal{O}_1$ imports all other ontologies (in)directly)

(1) Import of $\mathcal{O}_3$ into $\mathcal{O}_2$ is “safe” if $\mathcal{O}_3$ is locally complete w.r.t. $\mathcal{O}_2$, i.e., $\mathcal{O}_2 \cup \mathcal{O}_3 \equiv_{\mathsf{sig}(\mathcal{O}_3)} \mathcal{O}_3$

(2) Import of $\mathcal{O}_2, \ldots, \mathcal{O}_5$ into $\mathcal{O}_1$ is “safe” if $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_5 \equiv_{\mathsf{sig}(\mathcal{O}_2 \cup \cdots \cup \mathcal{O}_5)} \mathcal{O}_2 \cup \cdots \cup \mathcal{O}_5$

"Safe" imports

Previous example ograph:

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

(i.e., $\mathcal{O}_1$ imports all other ontologies (in)directly)

(1) Import of $\mathcal{O}_3$ into $\mathcal{O}_2$ is "safe" if $\mathcal{O}_3$ is locally complete w.r.t. $\mathcal{O}_2$, i.e., $\mathcal{O}_2 \cup \mathcal{O}_3 \equiv_{\mathsf{sig}(\mathcal{O}_3)} \mathcal{O}_3$

(2) Import of $\mathcal{O}_2, \ldots, \mathcal{O}_5$ into $\mathcal{O}_1$ is "safe" if $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_5 \equiv_{\mathsf{sig}(\mathcal{O}_2 \cup \cdots \cup \mathcal{O}_5)} \mathcal{O}_2 \cup \cdots \cup \mathcal{O}_5$

Sufficient condition for (1):

(1′) $\mathsf{mod}(\mathsf{sig}(\mathcal{O}_3), \mathcal{O}_2 \cup \mathcal{O}_3) = \mathcal{O}_3$ (for suitable module notion mod)

… and similarly for **(2)**

“Safe” imports

Hence …

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

(i.e., $\mathcal{O}_1$ imports all other ontologies (in)directly)

. . . to check whether $\mathcal{O}_2$ “safely” imports $\mathcal{O}_3$, we can test whether

(1′) $\text{mod}(\text{sig}(\mathcal{O}_3), \mathcal{O}_2 \cup \mathcal{O}_3) = \mathcal{O}_3$

"Safe" imports

Hence ...

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

(i.e., $\mathcal{O}_1$ imports all other ontologies (in)directly)

... to check whether $\mathcal{O}_2$ "safely" imports $\mathcal{O}_3$, we can test whether

(1′) $\text{mod}(\text{sig}(\mathcal{O}_3), \mathcal{O}_2 \cup \mathcal{O}_3) = \mathcal{O}_3$

But is this enough?

"Safe" imports

Hence …

$$\begin{array}{lllll} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

(i.e., $\mathcal{O}_1$ imports all other ontologies (in)directly)

… to check whether $\mathcal{O}_2$ "safely" imports $\mathcal{O}_3$, we can test whether

(1′) $\text{mod}(\text{sig}(\mathcal{O}_3), \mathcal{O}_2 \cup \mathcal{O}_3) = \mathcal{O}_3$

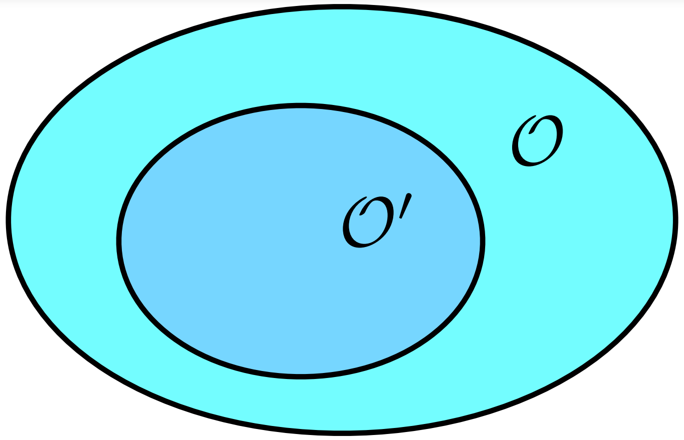
But is this enough?

$\mathcal{O}_2$ alone might not add new knowledge about sig$(\mathcal{O}_3)$
– but it may do so jointly with $\mathcal{O}_1, \mathcal{O}_4, \mathcal{O}_5$!

–> We need to be "more global" than local completeness!

… and we have relevance to check, too.

Significance



Definition

Let Σ be a signature and $\mathcal{O}' \subseteq \mathcal{O}$ ontologies.

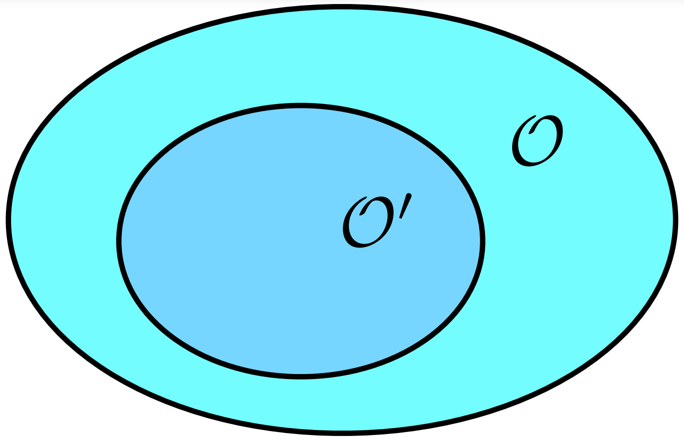
$\mathcal{O}'$ is Σ-significant in $\mathcal{O}$ if $\mathcal{O} \not\equiv_\Sigma \mathcal{O} \setminus \mathcal{O}'$.

Significance

Definition

Let Σ be a signature and $\mathcal{O}' \subseteq \mathcal{O}$ ontologies.

$\mathcal{O}'$ is Σ-significant in $\mathcal{O}$ if $\mathcal{O} \not\equiv_{\Sigma} \mathcal{O} \setminus \mathcal{O}'$.



This notion captures both …

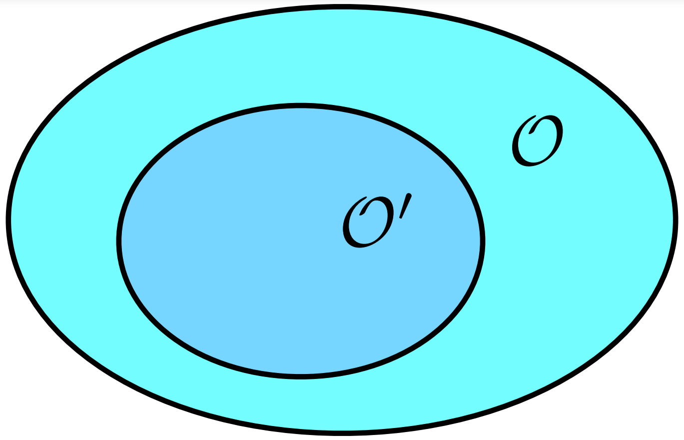
- **Relevance:**

  If $\mathcal{O}_3$ is sig$(\mathcal{O}_2)$-significant in $\mathcal{O}$,
  then its import adds knowledge about sig$(\mathcal{O}_2)$.

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

$$\mathcal{O} := \bigcup \mathcal{O}_i$$

Significance



Definition

Let Σ be a signature and $\mathcal{O}' \subseteq \mathcal{O}$ ontologies.

$\mathcal{O}'$ is Σ-significant in $\mathcal{O}$ if $\mathcal{O} \not\equiv_\Sigma \mathcal{O} \setminus \mathcal{O}'$.

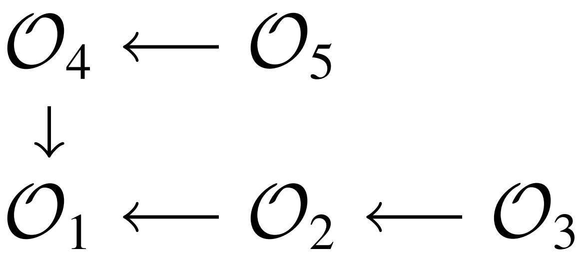
This notion captures both …

- **Relevance:**

  If $\mathcal{O}_3$ is sig$(\mathcal{O}_2)$-significant in $\mathcal{O}$,
  then its import adds knowledge about sig$(\mathcal{O}_2)$.

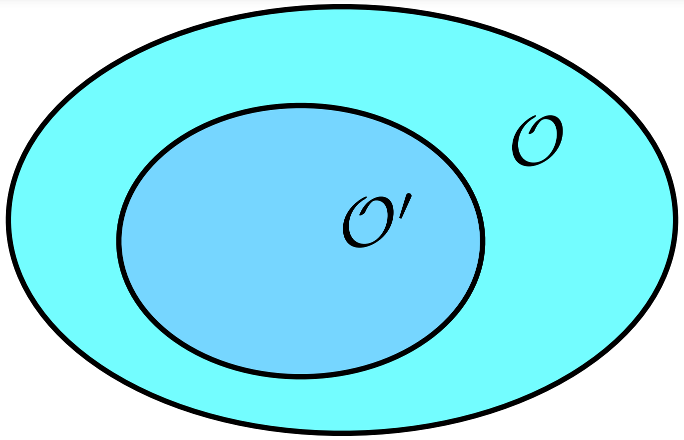
- **Completeness:**

  If $\mathcal{O}_2$ is sig$(\mathcal{O}_3)$-insignificant in $\mathcal{O}$,
  then it does not add knowledge about sig$(\mathcal{O}_3)$.



$$\mathcal{O} := \bigcup \mathcal{O}_i$$

Significance



Definition

Let Σ be a signature and $\mathcal{O}' \subseteq \mathcal{O}$ ontologies.

$\mathcal{O}'$ is Σ-significant in $\mathcal{O}$ if $\mathcal{O} \not\equiv_\Sigma \mathcal{O} \setminus \mathcal{O}'$.

This notion captures both ...

- **Relevance:**

  If $\mathcal{O}_3$ is sig$(\mathcal{O}_2)$-significant in $\mathcal{O}$,
  then its import adds knowledge about sig$(\mathcal{O}_2)$.

- **Completeness:**

  If $\mathcal{O}_2$ is sig$(\mathcal{O}_3)$-insignificant in $\mathcal{O}$,
  then it does not add knowledge about sig$(\mathcal{O}_3)$.

$$\begin{array}{ccccc} \mathcal{O}_4 & \longleftarrow & \mathcal{O}_5 & & \\ \downarrow & & & & \\ \mathcal{O}_1 & \longleftarrow & \mathcal{O}_2 & \longleftarrow & \mathcal{O}_3 \end{array}$$

$$\mathcal{O} := \bigcup \mathcal{O}_i$$

Hence:

An edge from $\mathcal{O}_i$ to $\mathcal{O}_j$ in the ograph is justified
if $\mathcal{O}_i$ is sig$(\mathcal{O}_j)$-significant in $\mathcal{O}$.

Verifying significance

Goal:

Given ograph $G = (V, E)$,

determine the ratio of edges in G that are justified and of non-edges that are not justified

Verifying significance

Goal:

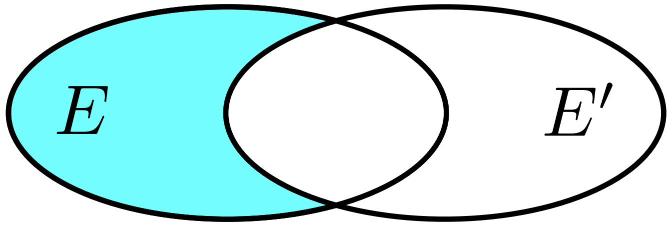
Given ograph $G = (V, E)$,

determine the ratio of edges in G that are justified
and of non-edges that are not justified

In other words:

Create a reference graph G' that captures all significances within G;
determine the relative similarity between their edge sets

$$\text{RSim}(G, G') := 1 - \frac{|E \setminus E'|}{|E|}$$



($E = E' \;\Rightarrow\; \text{RSim}(G, G') = 1 \qquad E \cap E' = \emptyset \;\Rightarrow\; \text{RSim}(G, G') = 0$)

Verifying significance

Goal:

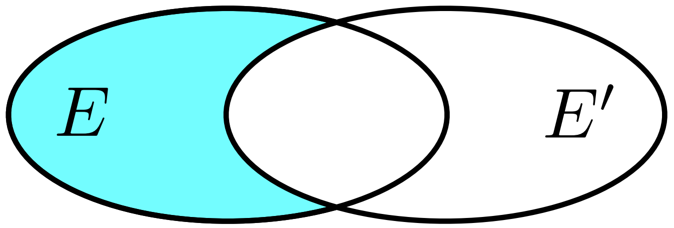
Given ograph $G = (V, E)$,

determine the ratio of edges in G that are justified
and of non-edges that are not justified

In other words:

Create a reference graph G' that captures all significances within G;
determine the relative similarity between their edge sets

$$\text{RSim}(G, G') := 1 - \frac{|E \setminus E'|}{|E|}$$



($E = E' \;\Rightarrow\; \text{RSim}(G, G') = 1$ $\qquad$ $E \cap E' = \emptyset \;\Rightarrow\; \text{RSim}(G, G') = 0$)

But significance is undecidable!

$\leadsto$ define G' using a sufficient condition for insignificance

Module-induced dependency graph

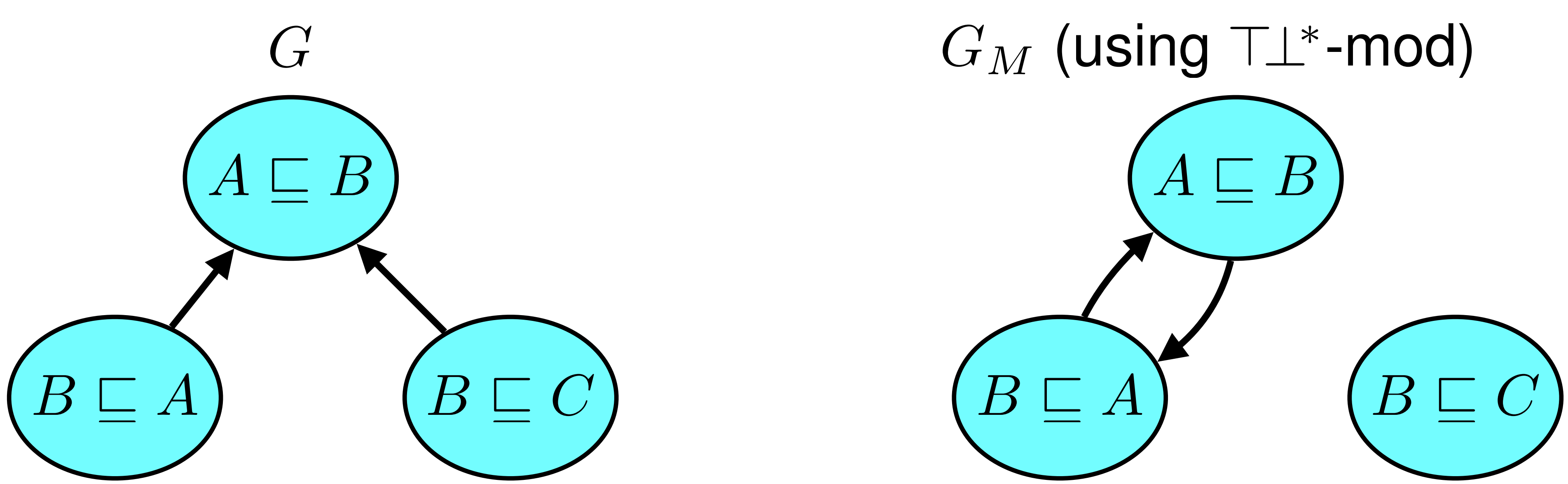
Definition

Let $G = (V, E)$ be an ograph and $\mathcal{O}$ the union of all ontologies in G.

The module-induced dependency graph of G is the ograph $G_M := (V, E')$ with edges

$$E' := \{(\mathcal{O}_1, \mathcal{O}_2) \mid \underbrace{\mathcal{O}_1 \cap \mathsf{mod}(\mathsf{sig}(\mathcal{O}_2), \mathcal{O}) \neq \emptyset}_{\text{(sufficient for “}\mathcal{O}_1\text{ is sig}(\mathcal{O}_2)\text{-insignificant in }\mathcal{O}\text{”)}}\}$$

Example

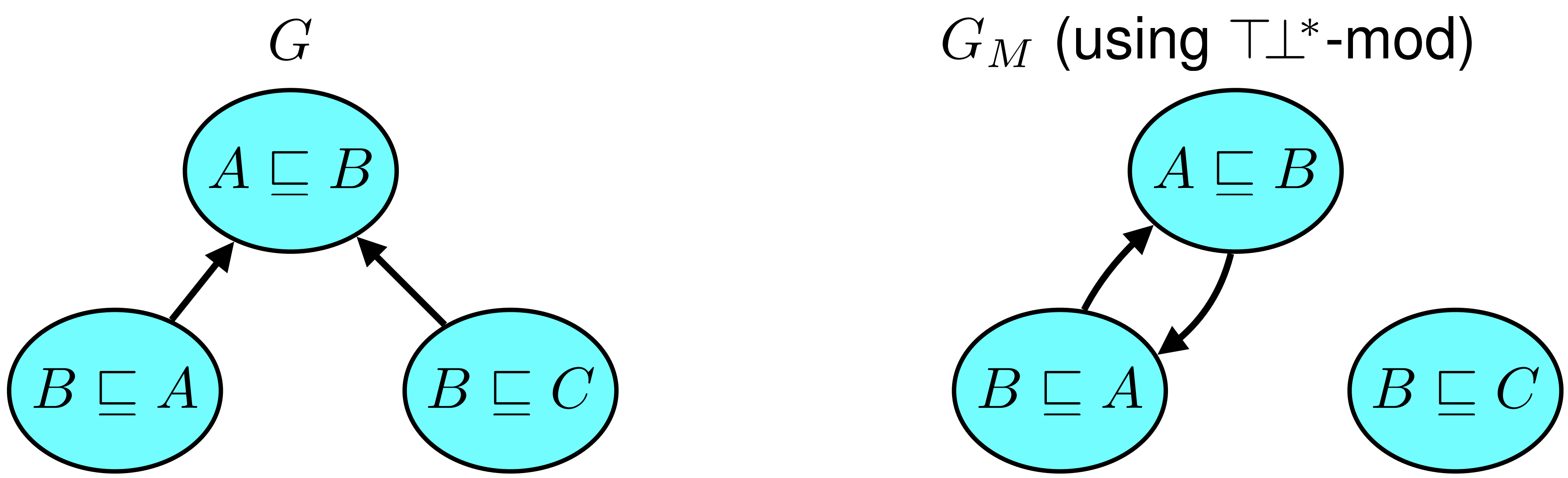


Module-induced relevance/completeness

Based on G_M, we define:

- the module-induced relevance of G MIR(G) := RSim(G, G^M)
- the module-induced completeness of G MIC(G) := RSim(G^M, G^*)

Example (continued)



$$\text{MIR}(G) = 1 - \frac{|E \setminus E'|}{|E|} = 1 - \frac{1}{2} = 0.5$$

$$\text{MIC}(G) = 1 - \frac{|E' \setminus E^*|}{|E'|} = 1 - \frac{1}{2} = 0.5$$

Atom-induced measures

Variant of our measures:

Reference graph based on dependency relation from Atomic Decomposition

[Del Vescovo, Parsia, Sattler, S. 2011]

Atomic Decomposition (AD)

- is an efficient method for automatically decomposing an ontology, based on a (nearly) arbitrary module notion mod$(\cdot, \cdot)$
- atoms (parts of the decomposition) are highly cohesive subsets of $\mathcal{O}$: maximal sets of axioms that always co-occur in modules for all Σ
- dependency relation between atoms represents logical dependencies within $\mathcal{O}$, again defined in terms of modules

Atom-induced measures

Atom-induced dependency graph G_A

. . . is defined similarly to G_M but with the following edge set:

$$E' := \{(\mathcal{O}_1, \mathcal{O}_2) \mid \text{some atom overlapping with } \mathcal{O}_2 \text{ depends on some atom overlapping with } \mathcal{O}_1\}$$

. . . is a subgraph of G_M (we have a simple proof)

Atom-induced relevance/completeness (AIR, AIC)

. . . are defined analogously to MIR and MIC, based on G_A

Experiments

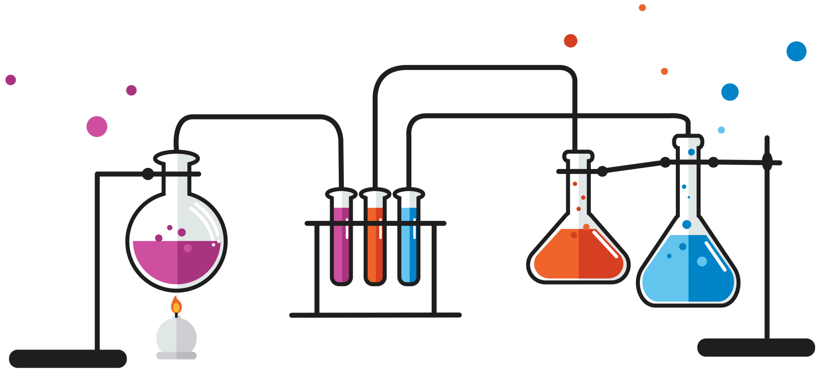


image: Freepik.com

Implementation

… based on the modularity/AD code in the OWL API

Evaluation

- corpus: 45 ontologies from the BioPortal snapshot
  (with 1 to 31 imports per ontology; altogether > 200 ontologies)
- median MIC and AIC: ≈0.75 (stddev ≈0.28, min ≈0.09)
  median MIR and AIR: ≈0.89 (stddev ≈0.22, min ≈0.22)
- MIC, AIC = 1 for 18 ontologies (import closures ≤ 4!)
- MIR, AIR = 1 for 21 ontologies (import closures ≤ 9)
- strong, significant correlation between MIx and AIx

Hypotheses tested

(H1) Are ontologies with many imports less likely to be "modular"?

Yes: strong, significant negative correlation between MIC/AIC and size of import closure

(but not for MIR/AIR)

(H2) Do "non-modular" ontologies tend to have both low relevance and low completeness?

No: no significant correlation between MIC and MIR, or AIC and AIR

Discussion

G_M and G_A are not “repairs” of G.

The precise numerical values are to be taken with caution.

In some scenarios, it is reasonable to assume relevance and completeness; in others it is not.

There is no precise general understanding of “modular” and “logical dependency”. Our definitions capture only 2 possible variants.

Outlook

Possible next steps

- Investigate further guarantees, e.g.: is all imported knowledge reused?
- When do the two reference graphs differ?
- Experiments with module notion providing minimal modules, e.g. MEX?
- Use of our measures in an optimisation problem for automatically calculating a “good” modular structure?

Thank you.

CC-BY-2.0 kennymatic Flickr

Thank you.

¿Preguntas?

Vrae?

Otázky?

Fragen?

Questões?

Pytania?

Questioni?

Ερωτήσεις;

Вопросы?

CC-BY-2.0 kennymatic Flickr

Questions?

Spørsmål?