大纲

- Debugging ontologies: axiom pinpointing;
- Modular ontologies;
- Versioning for ontologies;
- Module extraction;

一、Debugging

> KR最怕的就是bug

Axiom Pinpointing

> 精准定位

- 选取满足某个GCI的最小TBox

Let $\mathcal{T}$ be an ontology and C, D concepts such that

$$\mathcal{T} \models C \sqsubseteq D.$$

The **pinpointing set**

$$\mathsf{Pin}(\mathcal{T}, C \sqsubseteq D)$$

of $\mathcal{T}$ w.r.t. $C \sqsubseteq D$ consists of all minimal subsets $\mathcal{T}'$ of $\mathcal{T}$ such that

$$\mathcal{T}' \models C \sqsubseteq D.$$

Example

$$\mathcal{T} = \{\mathsf{Father} \sqsubseteq \mathsf{Male}, \mathsf{Male} \sqsubseteq \mathsf{Human}, \mathsf{Car} \sqsubseteq \mathsf{Vehicle}\}$$

Then $\mathsf{Pin}(\mathcal{T}, \mathsf{Father} \sqsubseteq \mathsf{Human}\}$ consists of only one set, namely the set

$$\{\mathsf{Father} \sqsubseteq \mathsf{Male}, \mathsf{Male} \sqsubseteq \mathsf{Human}\}$$

algorithm

> 一定能找到一条路径

Input $\mathcal{T} = \{\alpha_1, \ldots, \alpha_n\}$ and $C \sqsubseteq D$.

1. if $C \not\sqsubseteq_{\mathcal{T}} D$, then
2. return $\mathbf{Pin}(\mathcal{T}, C \sqsubseteq D)$ empty
3. set $\mathcal{S} := \mathcal{T}$
4. for $1 \leq i \leq n$ do
5. if $C \sqsubseteq_{\mathcal{S} \setminus \{\alpha_i\}} D$ then
6. $\mathcal{S} := \mathcal{S} \setminus \{\alpha_i\}$
7. return $\mathcal{S}$.

- 遍历所有GCI
- 看删除这条后还成不成立

二、Modularity

> 很大很复杂，但我们希望ontology越小越好，表达力强

signature

- A **signature** is a finite set of concept and role names.
- The **signature** $\mathsf{sig}(C)$ of a concept C is the set of concept and role names that occur in C. For example,

$$\mathsf{sig}(\mathsf{Human} \sqcap \exists \mathsf{has\_child}.\top) = \{\mathsf{Human}, \mathsf{has\_child}\}$$

- The **signature** $\mathsf{sig}(C \sqsubseteq D)$ of a concept inclusion is defined as

$$\mathsf{sig}(C \sqsubseteq D) = \mathsf{sig}(C) \cup \mathsf{sig}(D).$$

- The **signature** $\mathsf{sig}(\mathcal{T})$ of a TBox $\mathcal{T}$ is defined as the union of the signatures of its concept inclusions.

Intuitively, the signature $\mathsf{sig}(\mathcal{T})$ summarises the **subject matter** or topic of $\mathcal{T}$. In contrast, the symbols $\sqcap$, $\exists$, and $\top$ are logical symbols that are not part of the subject matter of $\mathcal{T}$.

- 所有concept和role name

Module

Let $\mathcal{T}$ be a TBox. A subset $\mathcal{M}$ of $\mathcal{T}$ is called a **module** of $\mathcal{T}$ if

$$\mathcal{M} \models C \sqsubseteq D \quad \Leftrightarrow \quad \mathcal{T} \models C \sqsubseteq D$$

for all concept inclusions $C \sqsubseteq D$ with $\mathsf{sig}(C \sqsubseteq D) \subseteq \mathsf{sig}(\mathcal{M})$.

EU? NCI?

Consider an ontology $\mathcal{EU}$ about EU research projects that uses as a module for its medical terms a medical ontology:

$$\begin{aligned}
\textsf{Genetic_Disorder_Project} &\equiv \textsf{Project} \sqcap \exists \textsf{has_Focus}.\textsf{Genetic_Disorder} \\
\textsf{Cystic_Fibrosis_EUProject} &\sqsubseteq \textsf{EUProject} \sqcap \exists \textsf{has_Focus}.\textsf{Cystic_Fibrosis} \\
\textsf{EUProject} &\sqsubseteq \textsf{Project}
\end{aligned}$$

uses NCI (i.e., NCI $\subseteq \mathcal{EU}$):

$$\begin{aligned}
\textsf{Cystic_Fibrosis} &\equiv \textsf{Fibrosis} \sqcap \exists \textsf{located_In}.\textsf{Pancreas} \sqcap \exists \textsf{has_Origin}.\textsf{Genetic_Origin} \\
\textsf{Genetic_Fibrosis} &\equiv \textsf{Fibrosis} \sqcap \exists \textsf{has_Origin}.\textsf{Genetic_Origin} \\
\textsf{Genetic_Fibrosis} &\sqsupseteq \textsf{Fibrosis} \sqcap \exists \textsf{located_In}.\textsf{Pancreas} \\
\textsf{Genetic_Fibrosis} &\sqsubseteq \textsf{Genetic_Disorder} \\
\textsf{DEFBI_Gene} &\sqsubseteq \textsf{Immuno_Protein_Gene} \sqcap \exists \textsf{associated_with}.\textsf{Cystic_Fibrosis}
\end{aligned}$$

NCI

NCI is a **module** of $\mathcal{EU}$ because

$$\text{NCI} \models C \sqsubseteq D \quad \Leftrightarrow \quad \mathcal{EU} \models C \sqsubseteq D$$

for all concept inclusions using medical terms only (i.e., all $C \sqsubseteq D$ such that $\textbf{sig}(C \sqsubseteq D) \subseteq \textbf{sig}(\text{NCI})$).

- proof

Proof that NCI is a module of $\mathcal{EU}$

As $\mathcal{EU} \supseteq$ NCI, we clearly have

$$\text{NCI} \models C \sqsubseteq D \quad \Rightarrow \quad \mathcal{EU} \models C \sqsubseteq D,$$

for **all** $C \sqsubseteq D$.

Conversely, we show

$$\text{NCI} \not\models C \sqsubseteq D \quad \Rightarrow \quad \mathcal{EU} \not\models C \sqsubseteq D,$$

for all $C \sqsubseteq D$ with $\textbf{sig}(C \sqsubseteq D) \subseteq \textbf{sig}(\text{NCI})$.

Assume NCI $\not\models C \sqsubseteq D$ for a $C \sqsubseteq D$ with $\textbf{sig}(C \sqsubseteq D) \subseteq \textbf{sig}(\text{NCI})$.
We show that $\mathcal{EU} \not\models C \sqsubseteq D$.

We find an interpretation $\mathcal{I}$ satisfying NCI with a $d \in \Delta^{\mathcal{I}}$ such that

$$d \in C^{\mathcal{I}}, \quad d \notin D^{\mathcal{I}}$$

Define an extension $\mathcal{I}'$ of $\mathcal{I}$ by setting

- $\textsf{Project}^{\mathcal{I}'} = \textsf{EUProject}^{\mathcal{I}'} \subseteq \Delta^{\mathcal{I}}$ arbitrary;
- $\textsf{has\_Focus}^{\mathcal{I}'} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ arbitrary;
- $\textsf{Genetic\_Disorder\_Project}^{\mathcal{I}'} = (\textsf{Project} \sqcap \exists \textsf{has\_Focus}.\textsf{Genetic\_Disorder})^{\mathcal{I}'}$
- $\textsf{Cystic\_Fibrosis\_EUProject}^{\mathcal{I}'} = (\textsf{EUProject} \sqcap \exists \textsf{has\_Focus}.\textsf{Cystic\_Fibrosis})^{\mathcal{I}'}$.

By definition, $\mathcal{I}'$ satisfies $\mathcal{EU}$.
Moreover, we still have

$$d \in C^{\mathcal{I}'}, \quad d \notin D^{\mathcal{I}'}$$

because $\textbf{sig}(C \sqsubseteq D) \subseteq \textbf{sig}(\text{NCI})$. It follows that $\mathcal{EU} \not\models C \sqsubseteq D$.

General Result

Let $\mathcal{M} \subseteq \mathcal{O}$ be such that $\mathcal{O} \setminus \mathcal{M}$ is an

- acyclic $\mathcal{EL}$ terminology such that no A with $A \equiv C$ or $A \sqsubseteq C$ in $\mathcal{O} \setminus \mathcal{M}$ is in $\textbf{sig}(\mathcal{M})$.

Then $\mathcal{M}$ is a module of $\mathcal{O}$.

Proof as above: Every interpretation $\mathcal{I}$ satisfying $\mathcal{M}$ can be expanded to an interpretation $\mathcal{I}'$ satisfying $\mathcal{O}$ by interpreting the new symbols of $\mathcal{O}$ according to the definitions in $\mathcal{O} \setminus \mathcal{M}$.

In mathematical logic such an $\mathcal{O}$ is called a definitorial extension of $\mathcal{M}$.

$\mathcal{EU} \setminus$ NCI is not a module of $\mathcal{EU}$

NCI influences (and should influence) the meaning of terms in $\mathcal{EU} \setminus$ NCI. Firstly, we have

$$\text{NCI} \models \textsf{Cystic\_Fibrosis} \sqsubseteq \textsf{GeneticDisorder}$$

and so

$$\mathcal{EU} \models \textsf{Cystic\_Fibrosis} \sqsubseteq \textsf{GeneticDisorder}$$

But on the other hand,

$$\mathcal{EU} \setminus \text{NCI} \not\models \textsf{Cystic\_Fibrosis} \sqsubseteq \textsf{GeneticDisorder}$$

and we obtain

$$\mathcal{EU} \setminus \text{NCI} \not\models \textsf{Cystic\_Fibrosis\_EUProject} \sqsubseteq \textsf{Genetic\_Disorder\_Project}$$

and

$$\mathcal{EU} \models \textsf{Cystic\_Fibrosis\_EUProject} \sqsubseteq \textsf{Genetic\_Disorder\_Project}$$

Module checking

- 检查子集M时候是O的module
- 比可满足性更难（是、否）

- For $\mathcal{EL}$ this problem is ExpTime-complete;
- For $\mathcal{ALC}$ it is 2ExpTime-complete;
- For $\mathcal{ALCQIO}$ it is undecidable.

$\mathcal{EL}$ terminologies are an exception. For them "module-checking" is tractable. A variety of heuristic approaches to the definition of modules have been developed as well.

三、versioning

> ontology一直在更新，有不同版本

三种区别

1. versioning based on syntactic difference (syntactic diff);
2. versioning based on structural difference (structural diff);
3. versioning based on logical difference (logical diff).

syntactic句法顺序

- order+redundant冗余
- **syntactic的区别不重要**

- The syntactic diff underlies most existing version control systems used in software development. It works with text files and represents the difference between versions as blocks of text present in one version but not another, ignoring any meta-information about the document.
- Ontology versioning cannot rely on a purely syntactic diff operation: a variety of syntactic differences, for example, the order of ontology axioms or existence of redundant axioms, does not affect the semantics of ontologies.

  For example,
  $$\mathcal{T} = \{A \sqsubseteq B, E \sqsubseteq F\}$$
  is the same ontology as
  $$\mathcal{T}' = \{E \sqsubseteq F, A \sqsubseteq B\}$$
  but they are syntactically different.

structural（语义仍相同logically equivalent）

- 一个conjunction的区别就是不同structure

- The structural diff extends the syntactic diff by taking into account structural meta-information about the distinct versions of files compared.
- It has been suggested for dealing with structured and hierarchical documents such as UML diagrams, database schemas, or XML documents.
- For ontologies, the main characteristic of the structural diff is that it regards them as structured objects, such as a concept hierarchy or a set of concept definitions.

  For example,

  $$\mathcal{T} = \{\textbf{Father} \sqsubseteq \textbf{Human} \sqcap \textbf{Male}, \textbf{Father} \sqsubseteq \exists \textbf{has\_child}.\top\}$$

  is regarded as different from

  $$\mathcal{T}' = \{\textbf{Father} \sqsubseteq \textbf{Human} \sqcap \textbf{Male} \sqcap \exists \textbf{has\_child}.\top\}$$

  although they are logically equivalent.

logical（semantic语义）

Difference should be always given relative to a subject matter. Let S be a signature (subject matter). The logical difference between two ontologies versions $\mathcal{O}_1$ and $\mathcal{O}_2$ of an ontology with respect to S,

$$\textbf{Diff}_S(\mathcal{O}_1, \mathcal{O}_2),$$

consists of the set of subsumptions $C \sqsubseteq D$ with $\textbf{sig}(C \sqsubseteq D) \subseteq S$ which follow from $\mathcal{O}_1$ but not from $\mathcal{O}_2$, or vice versa:

$$\begin{aligned}\textbf{Diff}_S(\mathcal{O}_1, \mathcal{O}_2) \;=\; & \{C \sqsubseteq D \mid \mathcal{O}_1 \models C \sqsubseteq D, \mathcal{O}_2 \not\models C \sqsubseteq D, \textbf{sig}(C \sqsubseteq D) \subseteq S\} \cup \\ & \{C \sqsubseteq D \mid \mathcal{O}_2 \models C \sqsubseteq D, \mathcal{O}_1 \not\models C \sqsubseteq D, \textbf{sig}(C \sqsubseteq D) \subseteq S\}\end{aligned}$$

From a logical viewpoint, $\mathcal{O}_1$ and $\mathcal{O}_2$ say the same about S if, and only if, $\textbf{Diff}_S(\mathcal{O}_1, \mathcal{O}_2) = \emptyset$.

Note: a set $\mathcal{M} \subseteq \mathcal{O}$ is a module if, and only if $\textbf{Diff}_S(\mathcal{M}, \mathcal{O}) = \emptyset$.

- O的module和O的Diff为空

Let $\mathcal{M} \subseteq \mathcal{O}$ and S a signature. $\mathcal{M}$ is called a

S-module of $\mathcal{O}$

if $\textbf{Diff}_S(\mathcal{M}, \mathcal{O}) = \emptyset$.

If one is interested in a set of terms S but does not want to use the whole ontology $\mathcal{O}$, one can instead use any S-module of $\mathcal{O}$.

Problem: Extract a **minimal** S-module from a given ontology $\mathcal{O}$ for re-use.