

# Poznáváme deskripční logiku $\mathcal{ALC}$

Petr Křemen

## 1 Rychlokurz Protégé

Systém Protégé je ontologický editor nad rámcovými formalizmy (Protégé 3 a nižší) a zejména pak nad standardy OWL (Protege 3) a OWL 2 (Protege 4). Je možné jej stáhnout z <http://protege.stanford.edu>.

- Nemáte-li nainstalován systém Protégé 4.1, nainstalujte si jej<sup>1</sup>.
- Nainstalujte si pluginy – **File** → **Check for plugins...**
  - Pellet
  - OWL2Query
  - OWLDiff
- Stáhněte si z webu <http://www.co-ode.org/ontologies/pizza/2007/02/12/pizza.owl> ukázkovou Pizza ontologii.
- Spusťte Protégé a otevřete tuto ontologii.
- Po otevření ontologie se zobrazí okno aplikace s několika záložkami – podívejte se do každé z nich. Několik poznámek:
  - Active Ontology** – zde se dozvíte užitečnou informaci o expresivitě dané ontologie (v případě `pizza.owl` byste měli vidět *SHOIN*), dále různé statistiky týkající se ontologie.
  - Entities/Classes** – bude přinejmenším ze začátku asi vaše nejpoužívanější záložka. V levé části vidíte taxonomii (TBOX) před klasifikací (Asserted class hierarchy) a po klasifikaci (Inferred class hierarchy). V centrální části pak jsou anotace týkající třídy zvolené v levé části a zejména pak její axiomy, ve kterých se tato třída vyskytuje.
  - Object/Data Properties/Individuals** – obsahují analogické zobrazení jako pro třídy.

---

<sup>1</sup>HTTP proxy nastavíte v souboru `Protege.lax`. Přidejte/změňte řádek `lax.nl.java.option.additional=-Dhttp.proxySet=true -Dhttp.proxyHost=proxy.felk.cvut.cz -Dhttp.proxyPort=80`

**OWLViz** – poskytuje jednoduchý, ale přehledný způsob grafické vizualizace struktury TBOXu.

**DL Query** – umožňuje pokládat jednoduché dotazy nad ontologií, typu “Nalezni všechny podtřídy/instance/... dané třídy”.

- Vyzkoušejte si klasifikaci ontologie – menu Reasoner a zvolte např. Pellet, nebo FaCT++. Podívejte se, v čem se odlišují výsledky (Asserted class hierarchy) a po klasifikaci (Inferred class hierarchy).

Další reference:

- Tutoriál na modelování ontologií s využitím Protégé:  
[http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf)
- Getting Started with Protégé 4:  
<http://protegewiki.stanford.edu/index.php/Protege4GettingStarted>

## 2 *ALC* prakticky

Uvažujme ontologii jazyka *ALC* :

$$\begin{aligned} \textit{Man} &\sqsubseteq \textit{Person} \\ \textit{Woman} &\sqsubseteq \textit{Person} \sqcap \neg \textit{Man} \\ \textit{Father} &\equiv \textit{Man} \sqcap \exists \textit{hasChild} \cdot \textit{Person} \\ \textit{GrandFather} &\equiv \exists \textit{hasChild} \cdot \exists \textit{hasChild} \cdot \top \\ \textit{Sister} &\equiv \textit{Person} \sqcap \neg \textit{Man} \sqcap \exists \textit{hasSibling} \cdot \textit{Person} \end{aligned}$$

1. Co jednotlivé axiomy vyjadřují? Zkuste je zformulovat v přirozeném jazyce.
2. Přepište poslední z axiomů do sémanticky ekvivalentní formule predikátového kalkulu.
3. Máme následující strukturu:

$$\begin{aligned} \Delta^{\mathcal{I}} &= \textit{Person}^{\mathcal{I}} = \{\textit{Jan}, \textit{Eva}\} \\ \textit{Man}^{\mathcal{I}} &= \{\textit{Jan}\} \\ \textit{Woman}^{\mathcal{I}} &= \{\textit{Eva}\} \\ \textit{Sister}^{\mathcal{I}} &= \{\} \\ \textit{Father}^{\mathcal{I}} &= \textit{GrandFather}^{\mathcal{I}} = \{\textit{Jan}\} \\ \textit{hasChild}^{\mathcal{I}} &= \{\langle \textit{Jan}, \textit{Jan} \rangle\} \\ \textit{hasSibling}^{\mathcal{I}} &= \{\} \end{aligned}$$

- a) Rozhodněte zda je tato struktura modelem naší ontologie. Pokud ne, upravte ji, tak aby modelem byla. Pokud ano, rozhodněte, zda tento model může odpovídat realitě.
- b) Víme, že jazyk  $\mathcal{ALC}$  má tzv. *tree model property* a *finite model property*. Má tento model stromovou strukturu ? Pokud ne, nalezněte model, který ji má.
- c) Je tento model konečný ? Pokud ne, nalezněte model, který je.
4. Pomocí dalších axiomů definujte následující koncepty:
- “Otec čistě chlapecké rodiny”
  - “Někdo, kdo má alespoň jednu sestru, ale žádného bratra.”
5. Zatím uvažujeme pouze dvě role *hasChild* a *hasSibling*. Při znalostním modelování je důležité specifikovat :
- globální doménu a obor hodnot** dané role, tedy tvrzení typu “Rolí *hasChild* spojujeme vždy osobu (instanci třídy *Person* – doména) s jinou osobou (instanci třídy *Person* – obor hodnot)”.
- lokální doménu a obor hodnot** dané role jednu konkrétní definici konceptu, např. “Každý otec čistě chlapecké rodiny může být rolí *hasChild* spojen nejvýše s muži (instancemi třídy *Man* – obor hodnot)”.
- Ukažte, jakým způsobem lze v deskripční logice  $\mathcal{ALC}$  namodelovat globální doménu a obor hodnot těchto rolí.
6. V systému Protégé 4 si založte novou ontologii a vložte do ní všechny tyto definice. Ověřte si prakticky správnost vašeho řešení z předchozího bodu (např. v záložce DL Query).