

Human Language Technology – Praktikum 7

ÜBUNG ZU ONTOLOGIEN

(06.06.2013, keine Abgabe erforderlich)

Ziel dieser kurzen Übung ist die Einführung in den *TopBraid Composer* und das Kennenlernen des automatischen Schlussfolgerns mit Hilfe des *Pellet* Reasoners. Hierzu wird schrittweise folgende Ontologie erstellt:

1. Erstellen der Klassen:
 - *Person*
 - *Chef*
 - *Mitarbeiter*
 - *Projekt*
2. Erstellen der Instanzen
 - *Meier* und *Müller* in Klasse *Mitarbeiter*
 - *Schulze* in Klasse *Chef*
 - *A* und *B* in *Projekt*
3. Erstellen der Relationen (properties)
 - *bearbeitet*
 - *betreut*
4. Festlegen von Domain und Range der Properties
 - *bearbeitet*: domain=*Mitarbeiter*, range=*Projekt*
 - *betreut*: domain=*Chef*, range=*Projekt*
5. Properties den Instanzen zuordnen
 - *Meier* bearbeitet *Projekt A*
 - *Müller* bearbeitet *Projekt B*
 - *Schulze* betreut *Projekt A* und *B*
6. SPARQL-Abfrage
 - Welche Mitarbeiter gehören zu Projekten, die von Schulze betreut werden?

```
SELECT ?mitarbeiter
WHERE{ :Schulze :betreut ?projekt .
       ?mitarbeiter :bearbeitet ?projekt . }
```
7. Erstellen eines inversen Properties *bearbeitet_von* (Inverse von *bearbeitet*)

8. SPARQL-Abfrage

- Wie 6., jedoch nun mit der Inverse formuliert

```
SELECT ?mitarbeiter
WHERE{ :Schulze :betreut ?projekt .
       ?projekt :bearbeitet_von ?mitarbeiter . }
```

9. Einfügen eines neuen *Chefs* namens „Peter“

10. *Peter* soll Projekt A übernehmen

11. Anzeigen der *Chefs* von Projekt A, danach von Projekt B

- **SELECT** ?chef
WHERE{ ?chef :betreut :A. }

12. Ändern des Properties betreut in *inversefunctional* (ein Projekt darf nur einen Chef haben)

13. SPARQL-Abfragen

- **SELECT** ?chef
WHERE{ ?chef :betreut :B. }
- **ASK**
WHERE{ :Schulze owl:sameAs :Peter. }

14. Ansehen der Inferences