



Universität Freiburg  
Institut für Informatik  
Prof. Dr. G. Lausen  
Michael Schmidt

Georges-Köhler Allee, Geb. 51  
D-79110 Freiburg  
Tel. (0761) 203-8120  
Tel. (0761) 203-8127

**Formale Grundlagen von Informationssystemen**  
**Sommersemester 2009**  
28.05.2009

## 5. Übungsblatt: RDF und SPARQL

### Übung 17 (Auswertung von SPARQL Anfragen, 1+1+2+2+2=8 Punkte)

Betrachten Sie die RDF-Datenbank

$$D := \{ (.:P1, \text{rdf:type}, \text{Person}), (.:P1, \text{name}, \text{"Pete"}), (.:P1, \text{age}, \text{"17"}), (.:P1, \text{email}, \text{"pete@abc.com"}), \\ (.:P2, \text{rdf:type}, \text{Person}), (.:P2, \text{name}, \text{"John"}), (.:P2, \text{email}, \text{"john@abc.com"}), \\ (.:P3, \text{rdf:type}, \text{Person}), (.:P3, \text{name}, \text{"Sue"}), (.:P3, \text{age}, \text{"21"}), \\ (.:P1, \text{knows}, .:P2), (.:P1, \text{knows}, .:P3), (.:P2, \text{knows}, .:P1), (.:P2, \text{knows}, .:P3) \}.$$

Zeichnen Sie den RDF Graphen. Werten Sie die folgenden SPARQL Graph Patterns schrittweise mittels der in der Vorlesung behandelten Semantik aus und beschreiben Sie die Semantik der Anfrage in Worten.

- $((?p, \text{rdf:type}, \text{Person}) \text{ AND } (?p, \text{age}, ?age)) \text{ FILTER } (?age > 20)$
- $((?p, \text{rdf:type}, \text{Person}) \text{ AND } (?p, \text{name}, ?name)) \text{ OPT } (?p, \text{age}, ?age)$
- $((?p, \text{rdf:type}, \text{Person}) \text{ AND } (?p, \text{age}, ?age)) \text{ UNION } ((?p, \text{rdf:type}, \text{Person}) \text{ AND } (?p, \text{email}, ?email))$
- $((?p, \text{rdf:type}, \text{Person}) \text{ OPT } (?p, \text{email}, ?email)) \text{ FILTER } (!\text{bound}(?email))$

Verifizieren Sie Ihre Ergebnisse mit der ARQ SPARQL Engine. Eine kurze Installationsanleitung, das RDF Dokument  $D$ , sowie die Beispielquery a) finden Sie auf der Übungsseite zum Download.

### Übung 18 (SPARQL Anfragen, 1+1+1+2+2=7 Punkte)

Betrachten Sie die Beispieldatenbank  $D$  aus der vorigen Aufgabe. Formulieren Sie die folgenden Anfragen in SPARQL und geben Sie das Endergebnis Ihrer Anfrage auf Dokument  $D$  an.

- Alle Paare von unterschiedlichen Personen, die einen gemeinsamen Bekannten haben.
- Die Namen aller Personen, die mindestens einen Bekannten haben oder jünger als 20 Jahre sind. Falls vorhanden, soll die Emailadresse und, ebenfalls falls vorhanden, das Alter dieser Personen mit ausgegeben werden.
- Konstruieren Sie mittels CONSTRUCT einen neuen Graphen, der alle Personen (inclusive Name) enthält, die mindestens zwei Personen kennen.
- Schreiben Sie eine ASK-Anfrage, die (auf beliebigem Eingabedokument  $D$ ) *yes* liefert, wenn der Constraint

$$\forall p(D(p, \text{rdf:type}, \text{Person}) \rightarrow \exists n D(p, \text{name}, n))$$

verletzt ist und *no* andernfalls (wir interpretieren die RDF Datenbank  $D$  hier als ternäre Relation  $D(\text{subject}, \text{predicate}, \text{object})$ , die alle RDF Tripel enthält).

e) Schreiben Sie eine ASK-Anfrage, die (auf beliebigem Eingabedokument  $D$ ) *yes* liefert wenn der Constraint

$\forall p_1, p_2, n (D(p_1, \text{rdf:type}, \text{Person}), D(p_2, \text{rdf:type}, \text{Person}), D(p_1, \text{foaf:name}, n), D(p_2, \text{foaf:name}, n) \rightarrow p_1 = p_2)$   
verletzt ist und *no* andernfalls (wir interpretieren die RDF Datenbank  $D$  hier als ternäre Relation  $D(\text{subject}, \text{predicate}, \text{object})$ , die alle RDF Tripel enthält).

Verifizieren Sie Ihre Ergebnisse mit ARQ.

### Übung 19 (Auswertung von SPARQL Anfragen, 1+1+1+2=6 Punkte)

Wir schreiben  $P_1 \equiv P_2$  genau dann wenn Graph Pattern  $P_1$  äquivalent zu Graph Pattern  $P_2$  auf jedem möglichen RDF Dokument ist, d.h. wenn  $\llbracket P_1 \rrbracket_D = \llbracket P_2 \rrbracket_D$  für alle RDF Dokumente  $D$  gilt.

Seien nun  $A, B$  und  $C$  beliebige SPARQL Graph Patterns. Beweisen Sie für die folgenden Äquivalenzen entweder, dass sie gelten, oder zeigen Sie anhand eines Gegenbeispiels, dass sie nicht gelten.

- a)  $A \text{ UNION } A \equiv A$
- b)  $A \text{ OPT } A \equiv A$
- c)  $A \text{ AND } A \equiv A$
- d)  $A \text{ UNION } (B \text{ AND } C) \equiv (A \text{ UNION } B) \text{ AND } (A \text{ UNION } C)$

### Übung 20 (Modellierung mit RDF, 3 Punkte)

Modellieren Sie das folgende Szenario als RDF Datenbank.

*Die Grundschule Freiburg hat drei MitarbeiterInnen: Lehrer Herr Maier, Lehrerin Frau Schmidt, sowie die Direktorin Frau Koster, die – zusätzlich zu ihren administrativen Aufgaben als Rektorin – auch Schüler unterrichtet. Herr Maier ist den Erstklässlern zugeteilt, während Frau Schmidt und Frau Koster gemeinsam die Klassen 2-4 unterrichten. Herr Maier hat eine Spezialausbildung als Sportlehrer und unterrichtet deshalb alle Klassen im Fach Sport. Jede Klasse hat einen Klassensprecher und mindestens einen Schüler. Marie ist Klassensprecherin der vierten Klasse. Ihre Lieblingsfächer sind Sport, Malen und Mathematik.*

Benutzen Sie bei der Modellierung URIs, Blank Nodes, Literale und Container. Wo immer es sinnvoll ist, greifen Sie auf das rdfs-Vokabular zurück, insbesondere auf `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:domain` und `rdfs:range`. Diskutieren Sie abschließend, welche Fakten sich gemäß der RDFS Semantik aus Ihrem Graphen herleiten lassen.

### Übung 21 (Relationale Speicherung von RDF Daten, 1+1+4=6 Punkte)

Wir betrachten die RDF Datenbank  $D$  aus Übung 17.

- a) Geben Sie die relationale Datenbankinstanz an, die die Daten aus  $D$  gemäß dem Triple Table Ansatz mit Dictionary Encoding speichert.
- b) Geben Sie die relationale Datenbankinstanz an, die die Daten aus  $D$  gemäß dem Ansatz der Vertikalen Partitionierung ohne Dictionary Encoding speichert.
- c) Übersetzen Sie die Anfragen aus Übung 17a)-d) in SQL-Anfragen über dem vertikal partitionierten Schema aus Teil b).

Abgabe 16.06.2009