Übungen zur Lehrveranstaltung Semantic Web Technologien

HTWG Konstanz Wintersemester 2009/2010 Steffen Schlönvoigt

Übung 1 – SPARQL – Definition von Anfragen

Gegen seien folgende Daten:

```
@prefix ex: <http://example.org/> .
ex:Sonne  ex:radius   "1.392e6"^^xsd:double ;
          ex:satellit ex:Merkur, ex:Venus, ex:Erde, ex:Mars .
ex:Merkur ex:radius   "2439.7"^^xsd:double .
ex:Venus  ex:radius   "6051.9"^^xsd:double .
ex:Erde   ex:radius   "6372.8"^^xsd:double ;
          ex:satellit ex:Mond .
ex:Mars   ex:radius   "3402.5"^^xsd:double ;
          ex:satellit ex:Phobos, ex:Deimos .
ex:Mond   ex:name     "Mond@de", "Moon@en" ;
          ex:radius   "1737.1"^^xsd:double .
ex:Phobos ex:name     "Phobos" .
ex:Deimos ex:name     "Deimos" .
```

Definieren Sie SPARQL-Anfragen, welche folgende Ergebnisse in tabellarischer Form liefern:

1. Objekte, die um die Sonne oder um einen Satelliten der Sonne kreisen
2. Objekte mit einem Volumen von über $2*10^{10}$ (km^3)
   (Objekte mit Radius können mit der Formel $V = 4/3\pi r^3$ berechnet werden)
   und falls vorhanden, mit dem Objekt dessen Satellit sie sind.
3. Objekte mit einem Satelliten, für den ein Name in englischer Sprache angegeben worden ist, die außerdem Satellit eines Objekts von über 3000 (km) Durchmesser sind
4. Objekte mit zwei oder mehr Satelliten (nehmen Sie an, dass unterschiedliche URIs hier unterschiedliche Objekte bezeichnen)

Übung 2 – SPARQL – Anfrage auf nicht vorhandene Informationen

Durch Kombination von Filtern mit optionalen Graph-Mustern kann man in SPARQL auch nach Elementen suchen, für die eine bestimmte Information nicht angegeben ist. Formulieren Sie eine Anfrage nach allen Himmelskörpern, die keinen Satelliten haben. Gehen Sie dabei davon aus, dass die o.a. Wissensbasis durch Tripel ergänzt wurde, die sämtlichen Himmelskörpern via rdf:type den Typ Himmelskörper zuweisen.

Übung 3 – SPARQL – Modifikatoren

Diese Aufgabe befasst sich mit Modifikatoren in SPARQL. Betrachten Sie folgendes RDF-Document:

```
@prefix ex: <http://example.org/> .
ex:a  ex:value "1"^^xsd:integer ;
      ex:value "3"^^xsd:integer .
ex:b  ex:value "2"^^xsd:integer .
```

Welches Ergebnis würde für diese RDF-Datenbasis auf die folgenden SPARQL-Anfragen zurück

geliefert?

1. `SELECT ?s ?v WHERE { ?s <http://example.org/value> ?v } ORDER BY ?v`
2. `SELECT ?s WHERE { ?s <http://example.org/value> ?v } ORDER BY ?v`
3. `SELECT ?s WHERE { ?s <http://example.org/value> ?v } ORDER BY DESC(?v) LIMIT 2`
4. `SELECT DISTINCT ?s WHERE { ?s <http://example.org/value> ?v } ORDER BY ?v`

Welches Ergebnis würden Sie erwarten, falls der letzten Anfrage LIMIT 1 hinzugefügt würde?

Übung 4 – OWL – OWL Modellierung

a) Modellieren Sie die folgenden Sätze in OWL DL indem Sie Ausschnitte aus entsprechenden OWL-Dokumenten in RDF/XML oder Turtle-Syntax angeben:

- Die Klasse Gemüse ist eine Unterklasse von PizzaBelag.
- Die Klasse PizzaBelag hat keine gemeinsamen Elemente mit der Klasse Pizza.
- Das Individuum Aubergine ist ein Element der Klasse Gemüse.
- Die abstrakte Rolle hatBelag besteht ausschließlich zwischen Elementen der Klasse Pizza und der Klasse PizzaBelag.
- Pizzen haben immer mindestens zwei Beläge.
- Jede Pizza der Klasse PizzaMargarita hat Tomate als Belag.
- Die Klasse Vegetarische Pizza besteht aus den Elementen, die sowohl in der Klasse PizzaOhneFleisch als auch in der Klasse PizzaOhneFisch sind.
- Keine Pizza der Klasse PizzaMargarita hat Belag aus der Klasse Fleisch.

b) Entscheiden Sie, ob die folgenden Aussagen im Zusammenhang mit der Pizza-Ontologie aus a) sinnvoll wären:

- Die Rolle hatZutat ist transitiv.
- Die Rolle hatBelag ist funktional.
- Die Rolle hatBelag ist invers funktional.

Übung 5 – OWL - Schlussfolgerung

a) Es soll das Konzept „vegetarische Pizza“ definiert werden. Welche der folgenden Definitionen ist dafür angemessen? Geben Sie dazu jeweils eine natürlichsprachliche Beschreibung der logischen Formeln an.

(a) VegetarischePizza ≡ Pizza ⊓ ¬∃ hatZutat.(Fleisch ⊓ Fisch)
(b) VegetarischePizza ≡ Pizza ⊓ ∀ hatBelag.(¬Fleisch ⊔ ¬Fisch)
(c) VegetarischePizza ≡ Pizza ⊓ ¬∃ hatBelag.Fleisch ⊓ ¬∃ hatBelag.Fisch
(d) VegetarischePizza ≡ Pizza ⊓ ∃ hatBelag.¬Fleisch ⊓ ∃ hatBelag.¬Fisch
(e) VegetarischePizza ≡ Pizza ⊓ ∀ hatZutat.(¬Fleisch ⊓ ¬Fisch)

b) Gegeben sei folgende Ontologie in DL-Syntax:

hatBelag ⊑ hatZutat	∃hatBelag.⊤ ⊑ Pizza	⊤ ⊑ ∀hatBelag.PizzaBelag
Gemüse ⊓ Käse ⊑ ⊥	Käse ⊓ Fleisch ⊑ ⊥	
Gemüse ⊓ Fleisch ⊑ ⊥	Käse ⊓ Fisch ⊑ ⊥	
Gemüse ⊓ Fisch ⊑ ⊥	Fleisch ⊓ Fisch ⊑ ⊥	

Betrachten Sie nun zusätzlich die folgenden Klassendefinitionen:

KäsePizza	≡	Pizza ⊓ ∃ hatBelag.Käse
PizzaSpinat	≡	∃ hatBelag.Spinat ⊓ ∃ hatBelag.Käse ⊓ ∀ hatBelag.(Spinat ⊔ Käse)
PizzaCarnivorus	≡	Pizza ⊓ ∀ hatBelag.(Fleisch ⊓ Fisch)
LeerePizza	≡	Pizza ⊓ ¬∃ hatBelag.⊤

(1) Welche der oben aufgeführten Klassen von Pizzas würde durch einen DL-Reasoner als Unterklasse von VegetarischePizza (gemäß einer korrekten Definition aus a) erkannt? Begründen Sie jeweils Ihre Entscheidung.

(2) Die Klassifikation unter (1) zeigt, dass einige der Pizzaklassen nicht das gewünschte Konzept modellieren. Wie könnte man ihre Definition korrigieren?

(3) Wie würde sich das unter (1) ermittelte Ergebnis verändern, wenn man bei der Definition von VegetarischePizza anstelle von ≡ nur ⊑ verwenden würde?

Übung 6 – OWL – Bonny und Clyde

Gegeben seien die Individuennamen `bonny` und `clyde`, die Klassennamen `Ehrlich`, `Klug`, `Verbrechen` und `Mensch` sowie die Rollennamen `verübt`, `verheiratetMit`, und `verdächtigt`.

Welche der folgenden Aussagen können in OWL 1 DL gemacht werden, welche in OWL 2 DL und welche überhaupt nicht? Geben Sie gegebenenfalls die entsprechenden Axiome an.

1. Jeder, der ehrlich ist und ein Verbrechen verübt hat, zeigt sich selbst an.
2. Wer klug und ehrlich ist, verübt kein Verbrechen.
3. Bonnie zeigt Clyde nicht an.
4. Niemand zeigt einen Menschen an, mit dem gemeinsam er ein Verbrechen verübt hat.
5. Clyde hat mindestens 10 Verbrechen verübt.
6. Bonnie und Clyde haben mindestens ein Verbrechen gemeinsam verübt.
7. Wer gemeinsam mit seinem Ehepartner ein Verbrechen verübt hat, der ist nicht ehrlich.
8. Jeder, der einen Verdächtigen kennt, ist selbst verdächtig.