Foundations of Semantic Web Technologies

Tutorial 8

Dörthe Arndt

WS 2023/24

Exercise 8.1 (Back to the start)**.** Write Wikidata queries (link: `http://query.wikidata.org/`) to answer the following questions from our first exercise sheet:

(a) Find two siblings who were both awarded or nominated for an academy award for acting, but never acted in a movie together.

(b) Find three countries with a population of at least one million people whose tallest building is taller than its highest natural elevation (i.e., its tallest mountain or hill measured from sea level). The building is measured from the ground up (i.e., a 3 metre building on a 1000 metre mountain counts as 3 metres).

(c) Find a German men's footballer from the current team who was born in the time of a FIFA world cup.

Exercise 8.2. Consider the following RDF document with information about celestial bodies.

```
@prefix  ex: <http://example.org/> .
ex:sun   ex:radius "1.392e6"^^xsd:double ;
         ex:satellite     ex:mercury , ex:venus ,
                          ex:earth ,   ex:mars  .
ex:mercury  ex:radius     "2439.7"^^xsd:double  .
ex:venus    ex:radius     "6051.9"^^xsd:double  .
ex:earth    ex:radius     "6372.8"^^xsd:double  ;
            ex:satellite  ex:moon .
ex:mars     ex:radius     "3402.5"^^xsd:double  ;
            ex:satellite  ex:phobos, ex:deimos  .
ex:moon     ex:name       "Mond@de", "Moon@en"  ;
            ex:radius     "1737.1"^^xsd:double  .
ex:phobos   ex:name       "Phobos" .
ex:deimos   ex:name       "Deimos" .
```

Specify SPARQL queries which yield the following results in the form of a table.

- Objects which orbit around the sun or around a satellite of the sun.

- Objects with a volume greater than $2 \cdot 10^{10}$ (km^3) together with the object – if it exists – of which they are a satellite. Assume for this that all celestial bodies are spherical.
- Objects with a satellite for which an English name is given, and which furthermore are satellites of an object with diameter greater than 3000 (km).
- Objects with two or more satellites. Assume for this that different URIs denote different objects.

Exercise 8.3. Translate the queries from Exercise 8.2 into SPARQL algebra expressions.

Exercise 8.4. Compute the solutions to the expressions from Exercise 8.3 with respect to the knowledge base from Exercise 8.2 step by step.

Exercise 8.5. The game *Sudoku* is about completing incomplete tables with numbers while respecting certain rules. We consider the following simple 4×4 Sudoku:

			3
			4
2			
3			

You have to fill in numbers with values 1 to 4 in the empty slots in the table so that no number occurs twice in any row or any column, and so that no number is duplicated within any of the marked 2×2 squares.

We now want to use SPARQL for solving this Sudoku, i.e. we want to obtain all possible solutions by means of answers to a SPARQL query. In order to do this, set up a suitable RDF document and SPARQL query.

Exercise 8.6. This exercise focuses on the OPTIONAL operator in SPARQL. Consider the following RDF document:

```
@prefix   :         <http://example/> .

:x1 :p 1 .
:x2 :p 2 .
:x3 :q 3 .
:x3 :q 4 .
```

Which result would each of the following SPARQL queries return for this RDF input?

(a)
```
SELECT *
WHERE
{ :x1 :p ?v . OPTIONAL { :x3 :q ?w .  OPTIONAL { :x2 :p ?v } }}
```

(b)
```
SELECT *
WHERE {
:x1 :p ?v .
OPTIONAL { :x3 :q ?w }
OPTIONAL { :x3 :q ?w  . :x2 :p ?v }
}
```

(c)
```
SELECT *
WHERE {
?x :p ?v .
OPTIONAL
{
?y :q ?w .
FILTER(?v=2)
}
}
```

(d)
```
SELECT *
WHERE {
?x :p ?v .
OPTIONAL
{
?y :q ?w .
FILTER(?v=2)
FILTER(?w=3)
}
}
```

Exercise 8.7 (Order dependency (bonus exercise))**.** Consider the following RDF graph:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ns0: <http://www.w3.org/2001/vcard-rdf/3.0#> .

<http://somewhere/JohnSmith>
a foaf:Person ;
ns0:FN "John Smith" ;
foaf:name "John Smith" .

<http://somewhere/RebeccaSmith>
a foaf:Person ;
ns0:FN "Becky Smith" ;
foaf:name "Rebecca Smith" .

<http://somewhere/SarahJones>
a foaf:Person ;
ns0:FN "Sarah Jones" .
```

```
<http://somewhere/MattJones>
a foaf:Person ;
foaf:name "Matthew Jones" .

<http://somewhere/AdamJones> a foaf:Person .
```

Which result do you get with the following query and why?

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vCard: <http://www.w3.org/2001/vcard-rdf/3.0#>

SELECT ?name
WHERE
{
?x a foaf:Person .
OPTIONAL { ?x foaf:name ?name }
OPTIONAL { ?x vCard:FN  ?name }
}
```