

Web-Informationssysteme, WS 2009/10
Übungsblatt 12

Besprechung am Di 02.02.2010

Hinweis: Ihr könnt sowohl die vereinfachte SPARQL Syntax aus der Vorlesung (AND statt ., OPT statt OPTIONAL, etc.) als auch die offizielle Syntax verwenden. Vermischen ist nicht erwünscht, wird aber auch akzeptiert.

Aufgabe 12-1 Labeling XML

Gegeben sei das folgende XML-Dokument `book.xml` (siehe Anlagen).

Dieses XML Dokument soll in einer accel-Tabelle wie in der Vorlesung gespeichert werden:

tag	start	stop	par
0	27	NULL	country

Gesucht sind Charakterisierungen der pre/post- und start/end-Label der Knoten in den folgenden Knotenmengen für die Kontextknoten c_1 , das p-Element #4 in Zeile 22, und c_2 , das section-Element in Zeile 35 (mit Titel "Audience"):

- | | |
|-----------------------------|----------------------------|
| a) descendant:: $*$ | f) preceding:: p |
| b) ancestor:: $*$ | g) child:: $*$ |
| c) self:: $title$ | h) parent:: $*$ |
| d) descendant-or-self:: $*$ | i) preceding-sibling:: $*$ |
| e) following:: $section$ | j) following-sibling:: $*$ |

Aufgabe 12-2 XPath to SQL

In der Vorlesung wurde eine systematische Übersetzung von XPath-Anfragen nach SQL besprochen. Übersetzen Sie nach diesem Schema folgende Anfragen:

- a) `/descendant::a/parent::b`
- b) `/descendant-or-self::b[child::a]`

Normalisieren (entschachteln) Sie anschließend die Anfragen vollständig. In der Vorlesung wird auch die symmetrische Äquivalenz von XPath-Anfragen besprochen. Die obigen Anfragen sind ein weiteres Beispiel für solche äquivalenten Anfragen. Können Sie diese Äquivalenz anhand Ihrer SQL-Anfragen beweisen?

Aufgabe 12-3 SPARQL

Betrachten Sie das folgende RDF-Dokument mit Informationen über einige Himmelskörper:

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

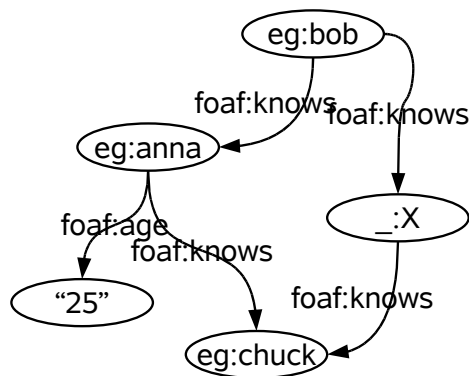
Geben Sie SPARQL-Anfragen an, welche die folgenden Ergebnisse in tabellarischer Darstellung liefern:

- Objekte, die um die Sonne oder um einen Satelliten der Sonne kreisen.
- Objekte mit einem Volumen von über $2 \times 10^{10} \text{ (km}^3\text{)}$ (Himmelskörper mit Radius können als kugelförmig angenommen werden, d.h., $V = \frac{4}{3}\pi r^3$) und, falls vorhanden, dem Objekt, dessen Satellit sie sind.
- Objekte mit einem Satelliten, für den ein englischsprachiger Name gegeben ist, die außerdem Satellit eines Objektes von über 3000 (km) Durchmesser sind.
- Objekte mit zwei oder mehr Satelliten (nehmen Sie an, dass unterschiedliche URIs hier unterschiedliche Objekte bezeichnen).
- 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.

Hinweis: Durch Kombination von Filtern und `bound()` mit optionalen Graph-Mustern kann man in SPARQL auch nach Elementen suchen, für die eine bestimmte Information nicht angegeben ist.

Aufgabe 12-4 SPARQL

Gegeben sei der folgende RDF Graph über Freundschaftsbeziehungen und seine Turtle Serialisierung. In dem Graphen kommen drei Arten von Knoten vor: URIs (wie zum Beispiel `http://www.example.com/anna`), Literale (wie der String `'25'`) und anonyme Knoten (`_:X`). URIs stehen für eine Resource, die man beschreiben möchte; so steht die URI `http://www.example.com/anna` wahrscheinlich für eine Person mit dem Namen Anna. Literale stehen dagegen für sich selbst, und anonyme Knoten sind bloße Platzhalter für Ressourcen, welche man nicht genauer spezifizieren kann oder will. Tripel mit anonymen Ressourcen können als existentielle Aussagen aufgefasst werden.



```

1 @prefix foaf:
2   <http://xmlns.com/foaf/0.1/> .
3 @prefix eg:
4   <http://www.example.com/persons> .
5
6 eg:bob foaf:knows eg:anna .
7 eg:bob foaf:knows _:X .
8 eg:anna foaf:age "25" .
9 eg:anna foaf:knows eg:chuck .
10 _:X foaf:knows eg:chuck .
  
```

Listing 1: Serialisierung des RDF Graphen

- a) Schreiben Sie eine SPARQL-Anfrage, welche die Bekannten von Bob selektiert (unter der Annahme, dass die URI `http://www.example.com/persons/bob` für die Person Bob steht, und die URI `http://xmlns.com/foaf/0.1/knows` für die Bekanntschaftsbeziehung zwischen Personen)!
- b) Ist das die Antwort, die man erwartet?