Weekly exercises IN3020 and IN4020
Week 14

May 5, 2022

1 RDF/RDFS/OWL

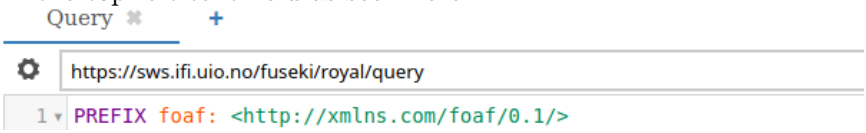
1. What is the purpose of RDF vocabularies?
2. What do we mean when we say that there are several RDF file formats (serializations)?
3. What is RDFS?
4. Use RDF/RDFS vocabulary to describe:
   (a) `:Person`, `:Toyota`, `:Car` and `:Vehicle` have the type `rdfs:Class`
   (b) `:drives` has the type `rdf:Property`.
   (c) `:john` is a `:Person` (individual)
   (d) `:my_car` is a `:Toyota` (individual)
   (e) every `:Toyota` is a `:Car` (subclass)
   (f) every `:Car` is a `:Vehicle` (subclass)
   (g) `:john :drives :my_car`
   (h) `:drives` has the domain `:Person`
   (i) `:drives` has the range `:Vehicle`
5. What is OWL?
6. What is an ontology?
7. OWL uses the so called Open-World assumption, while relational databases use a Closed-World assumption. You can find some information about this here: `https://en.wikipedia.org/wiki/Open-world_assumption`
   (a) What do we mean by open/closed world assumption.

(b) What practical implications can this have?

(c) Why do you think it makes sense that OWL assumes an open world?

2 SPARQL

1. Go to: `https://sws.ifi.uio.no/yasgui/`.
2. Enter `https://sws.ifi.uio.no/fuseki/royal/query` in the top left text field as seen here:



Now you should be able to query a small RDF dataset with some information about the norwegian royal family. The graph is described in more detail in Appendix A on page 4. You will need the following prefixes for these queries:

```
PREFIX fam: <http://www.ifi.uio.no/IN3020/family#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

2.1 URI of Haakon Magnus

Find the URI of a person (a `foaf:Person`) with the name (`foaf:name`) `"Haakon Magnus"`.

2.2 Sonja's age

Find the age (`foaf:age`) of a person (`foaf:Person`) with the name (`foaf:name`) "Sonja".

2.3 Young family members

Find name and age of all persons that have age below 18. You need the datatype propery `foaf:age`, the class `foaf:Person`.

2.4 The name of of Haakon Magnus' spouse

Find the name (`foaf:name`) of the spouse of (`fam:hasSpouse`) a person (a `foaf:Person`) with the name (`foaf:name`) "Haakon Magnus".

2.5 The parents of Olav V

Find the name of both the parents of Olav V. E.g., A person (`foaf:Person`) with the name (`foaf:name`) "Olav V". Two different properties connect a person to his/her parents: `fam:hasFather` and `fam:hasMother`.

2.6 URI, name and homepage

Find the URI, the name and the homepage (if they have one) of all persons in the dataset. You will need the class `foaf:Person` and the properties `foaf:name` and `foaf:homepage`.

2.7 DBPedia

It is possible to query large knowledge graphs online with SPARQL. Examples of such datasets are Wikidata and DBPedia.

Change the SPARQL endpoint to `https://dbpedia.org/sparql`. You need the following prefixes:

```
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
```

Find the population of the capital of a country called (`rdfs:label`) "Norway"@en. Where `dbo:Country` is the class of countries. `dbp:capital` is the relation between a country and its capital. `dbp:populationTotal` is the total population.

3 Neo4j

1. Describe the data model of Neo4j

4 Cypher

Go to `https://neo4j.com/sandbox` and create an account. Choose the *Movie* dataset and work through the Movies Project Tutorial.

A Excerpt RDF graph

Below is a short excerpt from the royal family graph. It consists of persons with names, spouses, ages, homepages, fathers and mothers:

```
fam:Sonja    a  foaf:Person ;
    fam:hasSpouse  fam:haraldv ;
    foaf:age       83 ;
    foaf:name      "Sonja" ;
    foaf:homepage
        <https://www.kongehuset.no/seksjon.html?tid=27197&sek=26953> .

fam:haakonmagnus    a   foaf:Person ;
    fam:hasFather  fam:haraldv ;
    fam:hasMother  fam:Sonja ;
    fam:hasSpouse  fam:mettemarit ;
    foaf:age       47 ;
    foaf:name      "Haakon Magnus" ;
    foaf:homepage
        <https://www.kongehuset.no/seksjon.html?tid=27658&sek=27226> .
```