INM713 Semantic Web Technologies and Knowledge Graphs

Laboratory 7: Basic OWL 2 RL Reasoning and SPARQL 1.1

Ernesto Jiménez-Ruiz

Academic course: 2020-2021
Updated: March 10, 2021

Contents

1 Git Repositories

Support codes for the laboratory sessions are available in *GitHub*. There are two repositories, one in Python and another in Java:

https://github.com/city-knowledge-graphs

2 Entailment

Consider the following set of triples (we will refer to them as the graph $\mathcal{G}$).

```
@PREFIX : <http://city.ac.uk/kg/lab4/>
@PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@PREFIX owl: <http://www.w3.org/2002/07/owl#> .
:Person     a                  owl:Class .
:Man        a                  owl:Class ;
            rdfs:subClassOf    :Person .
:Woman      a                  owl:Class ;
            rdfs:subClassOf    :Person .
:Parent     a                  owl:Class ;
            rdfs:subClassOf    :Person .
:Father     a                  owl:Class ;
            rdfs:subClassOf    :Parent ;
            rdfs:subClassOf    :Man .
:Mother     a                  owl:Class;
            rdfs:subClassOf    :Parent ;
            rdfs:subClassOf    :Woman .
:hasChild   a                  owl:ObjectProperty ;
            owl:inverseOf      :hasParent .
:hasParent  a                  owl:ObjectProperty ;
            rdfs:domain        :Person ;
            rdfs:range         :Parent .
:hasFather  a                  owl:ObjectProperty ;
            rdfs:subPropertyOf :hasParent ;
            rdfs:range         :Father .
:hasMother  a                  owl:ObjectProperty ;
            rdfs:subPropertyOf :hasParent ;
            rdfs:range         :Mother .
:Ann        a                  :Person ;
            :hasFather         :Carl ;
            :hasMother         :Juliet .
```

Task 2.1. Indicate if the following statements are derived by $\mathcal{G}$. $\mathcal{G}$ is within the OWL 2 RL profile so one could apply, among many others, similar inference rules to those for RDFS. Indicate in your proof which are the involved triples from $\mathcal{G}$.

Statement 1 `:Juliet :hasChild :Ann .`

Statement 2 `:Ann a :Child .`

Task 2.2. Check programmatically if the above statements are True or False via SPARQL queries over the extended graph (*i.e.*, after applying reasoning). The graph $\mathcal{G}$ is provided within the file `lab7.ttl` in the corresponding `lab7` folders.
Python. We are using the OWL-RL python library. The file in GitHub `OWLReasoning.py` expands our example graph $\mathcal{G}$ using the OWL 2 RL reasoning.
Java. We are using the Jena API. The file in GitHub `OWLReasoning.java` provides an example to set up the reasoner and extend the model with the new triples. Jena does not exactly support the OWL 2 RL profile, but includes a set of predefined reasoners. *OWLMiniReasoner* is the closest to OWL 2 RL.

Task 2.3. Check the solution for lab session 6 where OWL 2 RL reasoning has been enabled and its impact.

3 SPARQL 1.1 queries

Task 3.1. Over the RDF graph for the world cities dataset from lab 6, create a SPARQL query that counts the cities in each country. Order by number of cities. Test it programmatically.

Task 3.2. Using the Nobel Prizes kwnoledge graph from the lab session 3, create a SPARQL query that return countries with more than 10 Nobel laureates. Test it programmatically (via Nobel Prize Endpoint) or via the Nobel Prize query interface.

Task 3.3. You should now be able to do understand and create most of the queries in the SPARQL Playground (`http://sparql-playground.sib.swiss/`) we saw during the lab session 3. Try some of them.