INM713 Semantic Web Technologies and Knowledge Graphs

Laboratory 4: Reasoning with RDFS Semantics

Ernesto Jiménez-Ruiz

Academic course: 2020-2021
Updated: March 15, 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 Manual RDFS Inference

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#> .
:Person     a                  rdfs:Class .
:Man        a                  rdfs:Class ;
            rdfs:subClassOf    :Person .
:Woman      a                  rdfs:Class ;
            rdfs:subClassOf    :Person .
:Parent     a                  rdfs:Class ;
            rdfs:subClassOf    :Person .
:Father     a                  rdfs:Class ;
            rdfs:subClassOf    :Parent ;
            rdfs:subClassOf    :Man .
:Mother     a                  rdfs:Class;
            rdfs:subClassOf    :Parent ;
            rdfs:subClassOf    :Woman .
:Child      a                  rdfs:Class ;
            rdfs:subClassOf    :Person .
:hasParent  a                  rdf:Property ;
            rdfs:domain        :Person ;
            rdfs:range         :Parent .
:hasFather  a                  rdf:Property ;
            rdfs:subPropertyOf :hasParent ;
            rdfs:range         :Father .
:hasMother  a                  rdf:Property ;
            rdfs:subPropertyOf :hasParent ;
            rdfs:range         :Mother .
:isChildOf  a                  rdf:Property ;
            rdfs:domain        :Child ;
            rdfs:range         :Parent .
:Ann        a                  :Person ;
            :hasFather         :Carl ;
            :hasMother         :Juliet .
```

2.1 Exercises

In the following exercises decide $\mathcal{G}$ derives the given statement(s) and explain why/why not. In the positive case, then indicate the RDFS inference rules from the lecture (also found at `http://www.w3.org/TR/rdf-mt/`) to prove your answer. If the statement is not derived, then explain, informally or formally, why this is so. Formally can be done via a counterexample, *i.e.*, with an interpretation that entails $\mathcal{G}$, but it does not the statement.

Statement 1 `:Father rdfs:subClassOf :Person .`

Statement 2 `:Woman rdfs:subClassOf :Person .`

Statement 3 `:Juliet a :Person .`

Statement 4 `:Ann a :Child .`

Statement 5 `:Ann :isChildOf :Carl .`

Statement 6 `:Ann :hasParent :Juliet .`

Statement 7 `rdfs:range rdf:type rdfs:Resource .`

Statement 8 `:Mother rdfs:subClassOf :Person .`

2.2 Example solution

Example solution for Statement 1:

```
:Father rdfs:subClassOf :Person .
```

True, the statements is derived by $\mathcal{G}$. `:Father` is (transitively) a subclass of `:Person`. Rule **rdfs11**. Statements 1 and 2 below are found in $\mathcal{G}$ and are premises to the application of the inference rule **rdfs11**, which yields the statement we're after (Statement 3).

Proof:

1. `:Father rdfs:subClassOf :Parent` — P
2. `:Parent rdfs:subClassOf :Person` — P
3. `:Father rdfs:subClassOf :Person` — 1, 2, rdfs11

In the proof above each line is marked with "P" if the statement is a premise, *i.e.*, exits in $\mathcal{G}$, or with the rdfs rule and the line identification of the input statements.

3 RDFS Inference Programmatically

Python. We are going to use the OWL-RL python library which builds on top of RDFLib and has a RDFS reasoning component:

- Installation: `https://pypi.org/project/owlrl/5.2.1/`
- Documentation: `https://github.com/RDFLib/OWL-RL`

The file in GitHub `RDFSReasoning.py` expands our example graph $\mathcal{G}$ using the RDFS inference rules.

Java. We are still using the Jena API. The file in GitHub `RDFSReasoning.java` provides an example to set up the reasoner and extend the model with the new triples according to the RDFS inference rules.

Exercise: Check if the above statements are True or False via SPARQL queries over the extended graph or model (*i.e.*, the graph after applying the inference rules). The graph $\mathcal{G}$ is provided within the file `lab4.ttl`.

4 Solutions

4.1 Statement 2

`:Woman rdfs:subClassOf :Person .`

True. This triple is explicitly stated in $\mathcal{G}$

4.2 Statement 3

`:Juliet a :Person .`

True. Proof:

1. `:Ann :hasMother :Juliet .` — P
2. `:hasMother rdfs:range :Mother .` — P
3. `:Juliet rdf:type :Mother` — 1, 2, rdfs3
4. `:Mother rdfs:subclassOf :Woman` — P
5. `:Woman rdfs:subclassOf :Person` — P
6. `:Mother rdfs:subclassOf :Person` — 4, 5, rdfs11
7. `:Juliet rdf:type :Person` — 3, 6, rdfs9

4.3 Statement 4

`:Ann a :Child .`

False. It seems intuitive that `:Ann` is a child, but there is not connection between `:Ann` and `:Child` that can be entailed. To get this inference, `:hasFather` and/or `:hasMother` should be declared as sub-property of `:isChildOf`.

4.4 Statement 5

`:Ann :isChildOf :Carl .`

False. Similarly to Statement 4.

4.5 Statement 6

`:Ann :hasParent :Juliet .`

True. Proof:

1. `:Ann :hasMother :Juliet` — P
2. `:hasMother rdfs:subPropertyOf :hasParent` — P
3. `:Ann :hasParent :Juliet` — 1, 2, rdfs7

4.6 Statement 7

`rdfs:range rdf:type rdfs:Resource .`

True. This statement is an axiomatic triple and is always satisfied in RDFS models.

4.7 Statement 8

`:Mother rdfs:subClassOf :Person .`

True. Similarly to Statement 1.

4.8 RDFS Inference Programmatically

Solutions added to github. The solutions use `ASK` queries over the extended graph.