PhD course on Knowledge Graphs in the era of Large Language Models

First steps with Reasoning and the SPARQL query language

Ernesto Jiménez-Ruiz

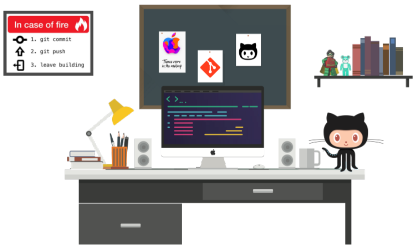
April 2024
Updated: April 24, 2024

Contents

1 Git Repository

Support codes for the laboratory sessions are available in *github*.

`https://github.com/city-knowledge-graphs/phd-course`



2 OWL 2 RL entailment

Consider the following set of triples (we will refer to them as the graph $\mathcal{G}_{\text{owl2rl}}$).

```
@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 .
```

Rule	If	Then add
rdf1	(x p y)	(p rdf:type rdf:Property)
rdfs2	(p rdfs:domain c) (x p y)	(x rdf:type c)
rdfs3	(p rdfs:range c) (x p y)	(y rdf:type c)
rdfs4a	(x p y)	(x rdf:type rdfs:Resource)
rdfs4b	(x p y)	(y rdf:type rdfs:Resource)
rdfs5	(p rdfs:subPropertyOf q) (q rdfs:subPropertyOf r)	(p rdfs:subPropertyOf r)
rdfs6	(p rdf:type rdf:Property)	(P rdfs:subPropertyOf p)
rdfs7	(p rdfs:subPropertyOf q) (x p y)	(x q y)
rdfs8	(c rdf:type rdfs:Class)	(c rdfs:subClassOf rdfs:Resource)
rdfs9	(c rdfs:subClassOf d) (x rdf:type c)	(x rdf:type d)
rdfs10	(c rdf:type rdfs:Class)	(c rdfs:subClassOf c)
rdfs11	(c rdfs:subClassOf d) (d rdfs:subClassOf e)	(c rdfs:subClassOf e)
rdfs12	(p rdf:type rdfs:ContainerMembershipProperty)	(p rdfs:subPropertyOf rdfs:Member)
rdfs13	(x rdf:type rdfs:Datatype)	(x rdfs:subClassOf rdfs:Literal)
	(p owl:inverseOf q)	(q owl:inverseOf p)
	(p owl:inverseOf q) (x p y)	(y q x)
	(p rdf:type owl:SymmetricProperty) (x p y)	(y p x)

schema only

data + schema

not relevant

Figure 1: Figure adapted from “Towards Efficient Schema-Enhanced Pattern Matching over RDF Data Streams”. International Semantic Web Conference (ISWC) 2011. Slides.

2.1 Inference rules (cheatsheet)

As seen in the lecture, Figure 1 summarizes the necessary inference rules we need for the lab. Following Figure 1 and the triples in $\mathcal{G}_{\text{owl2rl}}$ we can check if a given statement holds. For example:

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

True, the statements is derived by $\mathcal{G}_{\text{owl2rl}}$. `:Father` is (transitively) a subclass of `:Person` (Rule **rdfs11**). Statements 1 and 2 below are found in $\mathcal{G}_{\text{owl2rl}}$ 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 $\mathcal{G}_{\text{owl2rl}}$, or with the rdfs rule and the line identification of the input statements.

2.2 Manual inference

Task 1. Indicate if the following statements are derived by $\mathcal{G}_{\text{owl2rl}}$. $\mathcal{G}_{\text{owl2rl}}$ is within the OWL 2 RL profile so one could apply, among many others, similar inference rules to those for RDFS.[1] Indicate in your proof which are the involved triples from $\mathcal{G}_{\text{owl2rl}}$.

Statement 1 `:Juliet :hasChild :Ann .`

Statement 2 `:Ann a :Child .`

2.3 OWL 2 RL inference programmatically

We are using the OWL-RL python library. The file `OWLReasoning.py` in GitHub expands our example graph $\mathcal{G}_{\text{owl2rl}}$ using OWL 2 RL reasoning. A Jupyter notebook is also provided.

Task 2. Check programmatically if the above statements (Task 1) are True or False via SPARQL (ASK) queries over the extended graph (*i.e.*, after applying reasoning). The graph $\mathcal{G}_{\text{owl2rl}}$ is provided within the file `example-owl2rl.ttl` in the corresponding `lab-session-2` folder.

3 SPARQL Playground

We are using the SPARQL Playground dataset, a very simple data to learn SPARQL developed by researchers from the Swiss Institute of Bioinformatics `https://www.sib.swiss/`.

The SPARQL Playground is a very intuitive dataset to practice with both simple and sophisticated queries. Figure 2 shows a simplified version of the data and ontology. The same environment has also been used over more complex scenarios to understand the neXtProt and UniProt (knowledge bases about proteins) RDF models.

Task 3: Create the following queries. Test them programmatically in Python. Use the `playground.ttl` data, and the codes in the GitHub repositories (`lab-session-2`) as example.

Query 3.1 Query to return Eve's grandfather.

Query 3.2 Things that are dogs with color and sex. (Tip: give a look to the data in `playground.ttl`)

Query 3.3 Query that shows pets with their owners (Tip: owner may not exist, *i.e.*, it is optional)

Query 3.4 Select people with their gender and birth date ordered by gender and birth date (oldest first).

Query 3.5 Get the number of people by sex.

Query 3.6 Select persons that DO NOT have any pets

Query 3.7 For each pet species get the number of pets and their average weight.

[1] `https://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules`

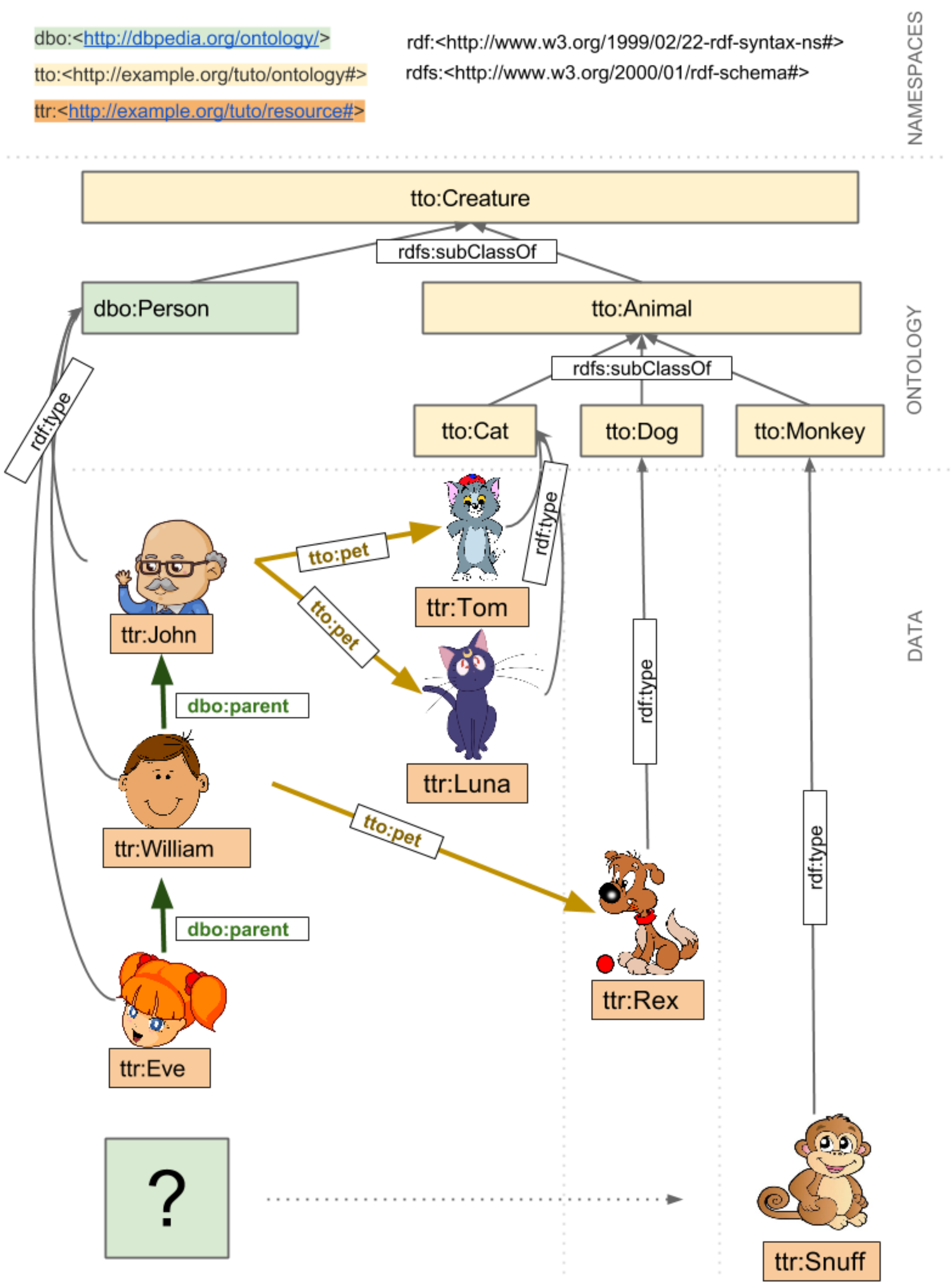


Figure 2: Simplified diagram of the data and “ontology” (from SPARQL Playground).