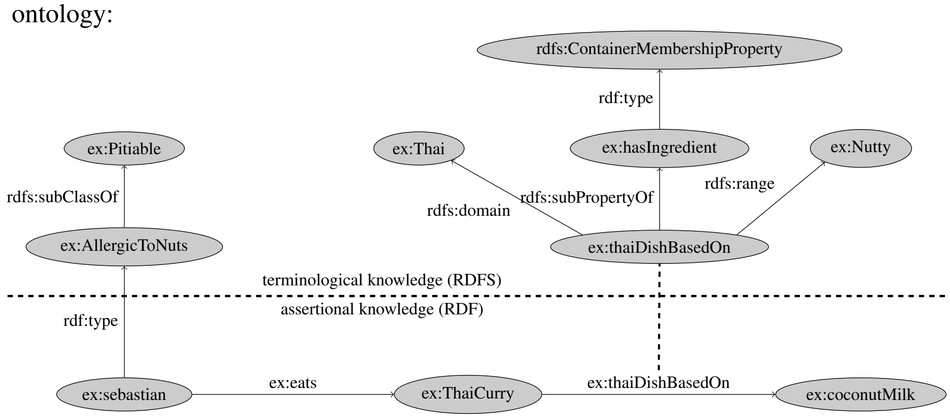
Foundations of Semantic Web Technologies

Tutorial 3

Sebastian Rudolph, Lukas Schweizer

SS 2016

Exercise 3.1. Describe an RDFS interpretation that is a model of the example ontology from Exercise 1.3. For reference, find here the RDFS graph representation of this ontology:



Exercise 3.2. For the ontology from Exercise 1.3, find

- a triple that is simply entailed,
- a triple that is RDF-entailed but not simply entailed,
- a triple that is RDFS-entailed but not RDF-entailed.

Exercise 3.3. The empty graph does not contain any triples (i.e. it corresponds to the empty set). Give derivations showing that the empty graph RDFS-entails the following triples:

(a) `rdfs:Resource rdf:type rdfs:Class .`

(b) `rdfs:Class rdf:type rdfs:Class .`

(c) `rdfs:Literal rdf:type rdfs:Class .`

(d) `rdf:XMLLiteral rdf:type rdfs:Class .`

(e) `rdfs:Datatype rdf:type rdfs:Class .`

(f) `rdf:Seq` `rdf:type` `rdfs:Class` .

(g) `rdf:Bag` `rdf:type` `rdfs:Class` .

(h) `rdf:Alt` `rdf:type` `rdfs:Class` .

(i) `rdfs:Container` `rdf:type` `rdfs:Class` .

(j) `rdf:List` `rdf:type` `rdfs:Class` .

(k) `rdfs:ContainerMembershipProperty` `rdf:type` `rdfs:Class` .

(l) `rdf:Property` `rdf:type` `rdfs:Class` .

(m) `rdf:Statement` `rdf:type` `rdfs:Class` .

(n) `rdfs:domain` `rdf:type` `rdf:Property` .

(o) `rdfs:range` `rdf:type` `rdf:Property` .

(p) `rdfs:subPropertyOf` `rdf:type` `rdf:Property` .

(q) `rdfs:subClassOf` `rdf:type` `rdf:Property` .

(r) `rdfs:member` `rdf:type` `rdf:Property` .

(s) `rdfs:seeAlso` `rdf:type` `rdf:Property` .

(t) `rdfs:isDefinedBy` `rdf:type` `rdf:Property` .

(u) `rdfs:comment` `rdf:type` `rdf:Property` .

(v) `rdfs:label` `rdf:type` `rdf:Property` .

Exercise 3.4. Let the instance I be given as follows (find a visualization in Fig. 1):

$$u(00,01), r(01,11), u(11,12), r(12,22), u(22,23), r(23,33), r(11,31), u(31,33)$$

and the Datalog program

$$\begin{aligned} d(x,z) &\leftarrow u(x,y) \wedge r(y,z) \\ d(x,z) &\leftarrow r(x,y) \wedge u(y,z) \\ d(x,z) &\leftarrow d(x,y) \wedge d(y,z) \end{aligned}$$

Provide a naïve algorithm for computing all instances of the relation (the predicate) d, based on the fixpoint semantics. For each iteration step, note down which tuple belong to the (intermediate) relation.

Exercise 3.5. Note that the theorem linking the RDFS entailment with the presented deduction calculus just guarantees soundness of the latter. When the calculus was provided in the RDF semantics specification, it was also considered complete, but a bit later that turned out not to be the case.

As an example of the calculus' incompleteness, consider the following set of triples:

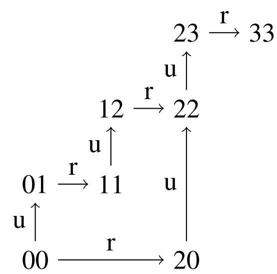


Figure 1: Visualization of the instance from Exercise 3.4

```
ex:isHappilyMarriedTo  rdfs:subPropertyOf     _:bnode .
_:bnode                rdfs:domain            ex:Person .
ex:markus              ex:isHappilyMarriedTo  ex:anja .
```

It is not hard to show that the triple

```
ex:markus rdf:type ex:Person .
```

is a semantic consequence of the above. However, it cannot be derived by means of the given deduction calculus. Make a suggestion how the calculus could be “repaired”.