

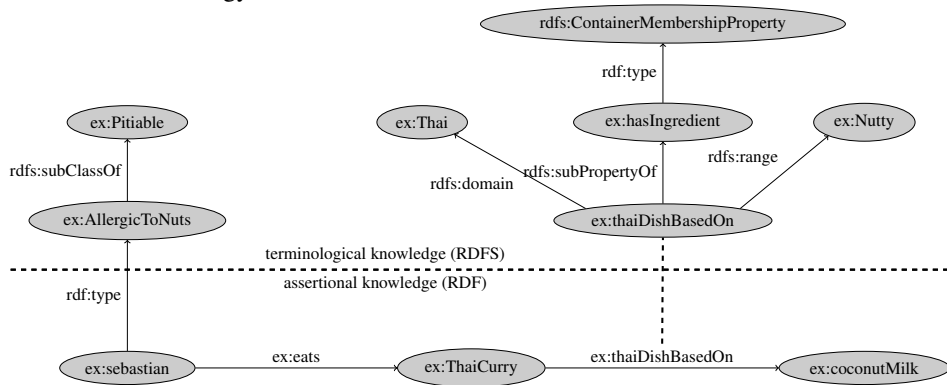
Foundations of Semantic Web Technologies

Tutorial 4

Dörthe Arndt

WS 2023/24

Exercise 4.1. Describe an RDFS interpretation that is a model of the example ontology from Exercise 1.4 from Tutorial 2. For reference, find here the RDFS graph representation of this ontology:



Exercise 4.2. For the ontology from Exercise 2.1, 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 4.3. 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:

- `rdfs:Resource rdfs:type rdfs:Class .`
- `rdfs:Class rdfs:type rdfs:Class .`
- `rdfs:Literal rdfs:type rdfs:Class .`
- `rdf:XMLLiteral rdfs:type rdfs:Class .`
- `rdfs:Datatype rdfs: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 4.4. 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:

```

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".