Foundations of Semantic Web Technologies

Tutorial 1

Sebastian Rudolph

SS 2015

Exercise 1.1. Explain the following elements of the RDF and RDFS vocabulary (presuming the usual namespace definitions).

(a) `rdf:type`	(f) `rdf:resource`
(b) `rdf:about`	(g) `rdf:nil`
(c) `rdf:Property`	(h) `rdfs:label`
(d) `rdf:Seq`	(i) `rdfs:member`
(e) `rdfs:Resource`	(j) `rdf:value`

Exercise 1.2. Decide if the following propositions are true or false:

(a) Blank nodes can stand for arbitrary resources.

(b) URIs can stand for arbitrary resources.

(c) Every blank node has an ID.

(d) Two blank nodes with different IDs can stand for the same resource.

(e) Two different URIs can stand for the same resource.

(f) Blank nodes carrying the same ID that occur in several RDF documents must stand for the same resource.

(g) URIs that occur in several RDF documents must stand for the same resource.

(h) Two different Literals can never stand for the same value.

(i) Two Literals with different datatype can never stand for the same value.

(j) A URI can never stand for a datatype value.

(k) Blank nodes cannot occur in the predicate position of triples.

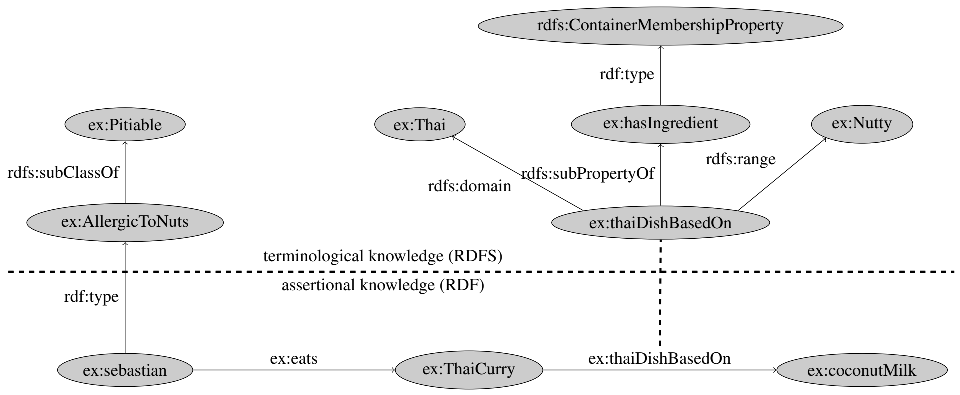
(l) Blank nodes cannot stand for properties (that is, resources that belong to the class `rdf:Property`).

Exercise 1.3. Consider the following RDF document:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:iswww="http://sw.edu/#"
>

<rdf:Description rdf:about="http://sw.edu/#germany">
  <rdf:type rdf:resource="http://sw.edu/#country" />
</rdf:Description>

<rdf:Description rdf:about="http://sw.edu/#capital_of">
  <rdf:type
   rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/
  >
  <rdfs:domain rdf:resource="http://sw.edu/#city" />
  <rdfs:range rdf:resource="http://sw.edu/#country" />
</rdf:Description>

<rdf:Description rdf:about="http://sw.edu/#country">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
  <rdfs:label xml:lang="de">Land</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://sw.edu/#berlin">
  <rdfs:label xml:lang="en">Berlin</rdfs:label>
  <rdf:type rdf:resource="http://sw.edu/#city" />
  <iswww:capital_of rdf:resource="http://sw.edu/#germany" />
</rdf:Description>

<rdf:Description rdf:about="http://sw.edu/#city">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
  <rdfs:label xml:lang="de">Stadt</rdfs:label>
</rdf:Description>

</rdf:RDF>
```

- Describe in natural language the content of this document.
- Draw the graph representation of the above document.
- Translate the document into Turtle syntax.

Exercise 1.4. Represent the following sketch of an RDF graph in RDF/XML syntax:



Exercise 1.5. Explain the difference between open and closed lists and give for each an example in Turtle syntax. What is meant by "open" and "closed"?

Exercise 1.6. Represent the following sentences graphically by means of reified triples (for space reasons, you may use usual prefixes instead of full URIs):

- Romeo thought that Juliet was dead.
- John believes that Mary wants to marry him.
- The dwarf noticed that somebody had been eating from his plate.

Exercise 1.7. Decide whether the following propositions can be satisfactorily modeled in RDFS and, if so, give the corresponding RDF(S) specification.

- Every pizza is a meal.
- Pizzas always have at least two toppings.
- Every pizza from the class `PizzaMargarita` has a `Tomato` topping.
- Everything having a topping is a pizza.
- No pizza from the class `PizzaMargarita` has a topping from the class `Meat`.
- "Having a topping" is a containedness relation.

Exercise 1.8. Describe an RDFS interpretation that is a model of the example ontology from Exercise 1.4.

Exercise 1.9. For the ontology from Exercise 1.4, 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 1.10. As you know, the unique name assumption does not hold in RDF(S), i.e. in a model, several URIs might be assigned to the same resource. Contemplate whether (and if so, how) it is possible to specify in RDFS that two given URIs refer to the same resource.

Exercise 1.11. 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:

1. `rdfs:Resource rdf:type rdfs:Class .`
2. `rdfs:Class rdf:type rdfs:Class .`
3. `rdfs:Literal rdf:type rdfs:Class .`
4. `rdf:XMLLiteral rdf:type rdfs:Class .`
5. `rdfs:Datatype rdf:type rdfs:Class .`
6. `rdf:Seq rdf:type rdfs:Class .`
7. `rdf:Bag rdf:type rdfs:Class .`
8. `rdf:Alt rdf:type rdfs:Class .`
9. `rdfs:Container rdf:type rdfs:Class .`
10. `rdf:List rdf:type rdfs:Class .`
11. `rdfs:ContainerMembershipProperty rdf:type rdfs:Class .`
12. `rdf:Property rdf:type rdfs:Class .`
13. `rdf:Statement rdf:type rdfs:Class .`
14. `rdfs:domain rdf:type rdf:Property .`
15. `rdfs:range rdf:type rdf:Property .`
16. `rdfs:subPropertyOf rdf:type rdf:Property .`
17. `rdfs:subClassOf rdf:type rdf:Property .`
18. `rdfs:member rdf:type rdf:Property .`
19. `rdfs:seeAlso rdf:type rdf:Property .`
20. `rdfs:isDefinedBy rdf:type rdf:Property .`
21. `rdfs:comment rdf:type rdf:Property .`
22. `rdfs:label rdf:type rdf:Property .`

Exercise 1.12. 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.

$$\begin{tikzcd}
& & 23 \arrow[r, "r"] & 33 \\
& 12 \arrow[r, "r"] & 22 \arrow[u, "u"'] & \\
01 \arrow[r, "r"] & 11 \arrow[u, "u"'] & & \\
00 \arrow[u, "u"'] \arrow[rr, "r"] & & 20 \arrow[uu, "u"'] &
\end{tikzcd}$$

Abbildung 1: Visualization of the instance from Exercise 1.12

Exercise 1.13. 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".