Foundations of Semantic Web Technologies
Solutions for Tutorial 1: RDF and RDF Schema
Sebastian Rudolph
SS 2015

Solution of Exercise 1.2

(a) Blank nodes can stand for arbitrary resources.
   ✓ wrong: not for predicates – they always have to be defined by an URI.

(b) URIs can stand for arbitrary resources.
    ✓ correct

(c) Every blank node has an ID.
    ✓ wrong: e.g. the value of the resource of the attribute rdf:parseType autom-
    atically generates a blank node without ID.

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

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

(f) Blank nodes carrying the same ID that occur in several RDF documents must
    stand for the same resource.
    ✓ wrong: An ID of a blank node is local.

(g) URIs that occur in several RDF documents must stand for the same resource.
    ✓ correct: URI stands for “Unique Resource Identifier”.

(h) Two different Literals can never stand for the same value
    ✓ wrong: 2.0 and 2.00 stand for the same value in xsd:decimal.

(i) Two Literals with different datatypes can never stand for the same value.
    ✓ wrong: 2 (xsd:integer) and 2.0 (xsd:decimal) stand for the same value,
    i.e. 2.

(j) A URI can never stand for a datatype value.
    ✓ wrong: stands for the value of datatype xsd:anyURI.

(k) Blank nodes cannot occur in the predicate position of triples.
    ✓ correct
(l) Blank nodes cannot stand for properties (that is, resources that belong to the class rdfProperty).

\*_wrong

Solution for Exercise 1.3 (b)

Solution of Exercise 1.3 (c)

@prefix rdf: <http://www.w3.org/1990/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix ex: <http://example.org/>

ex:germany     rdf:type     ex:country .
ex:capital_of  rdf:type     rdf:Property ;
               rdfs:domain   ex:city ;
               rdfs:range    ex:country .
ex:country     rdf:type     rdfs:Class ;
               rdfs:label    "Land"@de .
ex:berlin      rdf:type     ex:city ;
               rdfs:label    "Berlin"@en ;
ex:capital_of  ex:germany .
ex:city        rdf:type     rdfs:Class ;
               rdfs:label    "Stadt"@de .
Solution of Exercise 1.4

<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:ex="http://example.org/">

    <rdf:Description rdf:about="http://example.org/ThaiCurry">
        <ex:thaiDishBasedOn rdf:resource="http://example.org/CoconutMilk"/>
    </rdf:Description>

    <rdf:Description rdf:about="http://example.org/Sebastian">
        <rdf:type rdf:resource="http://example.org/AllergicToNuts"/>
        <ex:isst rdf:resource="http://example.org/ThaiCurry"/>
    </rdf:Description>

    <rdf:Description rdf:about="http://example.org/AllergicToNuts">
        <rdfs:subClassOf rdf:resource="http://example.org/Pitiable"/>
    </rdf:Description>

    <rdf:Description rdf:about="http://example.org/thaiDishBasedOn">
        <rdfs:subPropertyOf rdf:resource="http://example.org/hasIngredient"/>
        <rdfs:domain rdf:resource="http://example.org/Thai"/>
        <rdfs:range rdf:resource="http://example.org/Nutty"/>
    </rdf:Description>

    <rdf:Description rdf:about="http://example.org/hasIngredient">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#ContainerMembershipProperty"/>
    </rdf:Description>

</rdf:RDF>
Solution for Exercise 1.6
(a): Romeo thought that Julia was dead.

(b): John believes that Mary wants to marry him.

(c): The dwarf noticed that somebody had been eating from his plate.

Solution for Exercise 1.7  (b), (c), and (e): cannot be modeled in RDF(S).

@prefix rdf: <http://www.w3.org/1990/02/22-rdf-syntax-ns#>  .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>  .
@prefix ex: <http://example.org/>  .

- Every pizza is a meal.
  ex:Pizza rdfs:subClassOf ex:Meal.
• Everything having a topping is a pizza.
  ex:hasTopping rdfs:Domain ex:Pizza.

• “Having a Topping” is a containedness relation.
  ex:hasTopping rdf:type rdfs:ContainerMembershipProperty.