Foundations of Semantic Web Technologies

Tutorial 2

Dörthe Arndt

WS 2022/23

Exercise 1.1 (Datatypes)**.** There are several predefined datatypes in XML Schema. You find more information in the W3C-documentation: `http://www.w3.org/TR/xmlschema-2/`. Use that document in order to answer the following questions. You can use text search to find the information you need. Do not read the whole document.

(a) Read the information about the datatypes `xsd:decimal` and `xsd:dateTime`. What are the lexical space and the value space of these types (an informal description is enough). Provide an example (which differs from the ones given in the specification) for syntactical correct values of this type.

(b) XML Schema defines several numerical datatypes. Read the information about the types `xsd:int`, `xsd:double`, and `xsd:float`. Provide example instances for these datatypes.

Exercise 1.2 (True or false)**.** 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.

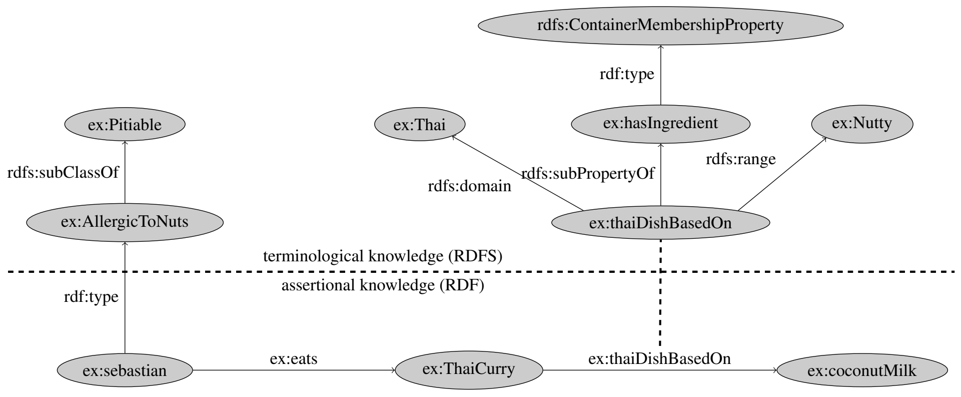
(k) Blank nodes cannot occur in the predicate position of triples.

Exercise 1.3 (From triples to graphs)**.** Consider the following RDF document:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ns0: <http://example.org/> .

<http://example.org/deutschland> a <http://example.org/Land> .
<http://example.org/hauptstadt_von>
        a rdf:Property ;
        rdfs:domain <http://example.org/Stadt> ;
        rdfs:range <http://example.org/Land> .

<http://example.org/Land>
        a rdfs:Class ;
        rdfs:label "country"@en .

<http://example.org/berlin>
        rdfs:label "Berlin"@en ;
        a <http://example.org/Stadt> ;
        ns0:hauptstadt_von ns0:deutschland .

ns0:Stadt
        a rdfs:Class ;
        rdfs:label "city"@en .
```

- Describe in natural language the content of this document.
- Draw the graph representation of the above document.

Exercise 1.4 (From graphs to triples)**.** Represent the following sketch of an RDF graph in RDF turtle syntax:



Exercise 1.5 (Hands On)**.** The goal of the exercise is to practice modelling data as RDF graphs using Turtle syntax. You should model the data contained in the given paragraph below as an RDF graph (using Turtle syntax). First some guidelines:

- You should use `http://rdfplayground.dcc.uchile.cl/`. Paste your Turtle data into the Text field on the left-hand side and hit `Check Syntax` to validate; then hit `Graph` to preview the RDF graph.

- The following example provides the prefixes you will need. Try it with the system linked previously.

```
@prefix ex:  <http://ex.org/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

ex:Ireland rdf:type ex:Country ; ex:capital ex:DublinCity ;
        ex:name "Ireland"@en , "Irlanda"@es ;
        ex:nationalDay "--03-17"^^xsd:gMonthDay .

ex:DublinCity rdf:type ex:City ;
        ex:name "Dublin"@en , "Baile Átha Cliath"@ga ;
        ex:county ex:DublinCounty ; ex:population 1000000 .

ex:DublinCounty rdf:type ex:County ;
        ex:name "Dublin"@en , "Baile Átha Cliath"@ga ;
        ex:country ex:Ireland .
```

- For brevity, in this exercise you need only explicitly provide the string labels/names where there are multiple given, e.g., in different languages (in practice you should give a human-readable label for everything). String literals should represent strings, names, titles, etc., not people, places, books, etc. You need only model data explicitly given; for example, you need only provide types where specified in the text (if it does not say that Chile is a country, you do not need to add this). Be sure to use the appropriate datatypes. Also avoid RDF lists/collections and n-ary relations where *not* necessary as they make the graph unnecessarily complicated. You can invent IRIs with an example `ex:` prefix but do use `rdf:type` and `xsd` datatypes (see above).

Your goal is to model the data represented in the following paragraph as an RDF graph (in Turtle syntax). Remeber, that there is no single correct answer (but of course, it is enough if you propose one).

- *Friday the 13th* (”*Viernes 13”* in Spanish, and ”*Freitag der 13*ïn German) was a film of the slasher genre directed by Sean Cunningham, and written by Victor Miller. It was the first film of the *Friday the 13th* series. In this series, the protagonist, Jason Voorhees, is the son of Pamela Voorhees. The original *Friday the 13th* movie was released on May 9$^{\text{th}}$, 1980 in the United States and on December 1$^{\text{st}}$, 1980 in Brazil. In the original movie, Jason was portrayed by Ari Lehman. Later, a reboot of the original film, also called *Friday the 13th*, was released in the United States on February 13$^{\text{th}}$, 2009; this time the movie was directed by Marcus Nispel, and Jason Voorhees was portrayed by Derek Mears. Over time,

> the *Friday the 13th* series would comprise twelve films in the following order: *Friday the 13th*, *Friday the 13th Part 2*, *Friday the 13th Part III*, *Friday the 13th: The Final Chapter*, *Friday the 13th: A New Beginning*, *Friday the 13th Part VI: Jason Lives*, *Friday the 13th Part VII: The New Blood*, *Friday the 13th Part VIII: Jason Takes Manhattan*, *Jason Goes to Hell: The Final Friday*, *Jason X*, *Freddy vs. Jason*, and *Friday the 13th (2009)*.

We might want to work on this example in class, so bring your laptop (for groups of 2). If you are confident enough (and I encourage you here!), send your solution in a single text file with ttl-extension to me. The idea would be that we model and discuss possible solutions in class.