

INF3580, Semantic Technologies

Mandatory Exercise (oblig) 1

Published: 26.02.2010
Due: 12.03.2010

Formalities

All questions need to be solved individually. On demand, you have to be able to explain your answers.

Put all files constituting your answers into one directory that has the same name as your UiO login ID. Make a ZIP or TAR archive of this directory, and send it by mail to `martingi@ifi.uio.no`, with a subject of “INF3580 Oblig 1”.

Question 1

Create a file containing RDF statements that describe the name and location of a number of places. It should contain at least 5 different places, including:

- Where you were born
- Where you went to school
- Where you currently live

Pick some favorite places for the rest. You will need the longitude and latitude of the places expressed in decimal degrees. You can find this by looking up the place in Wikipedia and clicking on “Coordinates” or “Koordinater” in the upper right corner of the page. This will tell you for instance that Hammerfest has latitude 70.66° and longitude 23.68° . (Positive numbers are used for north and east, negative ones for south and west.) You can also easily find coordinates by going to <http://www.itouchmap.com/latlong.html> and clicking on a place in the map.

The description should use the following namespaces:

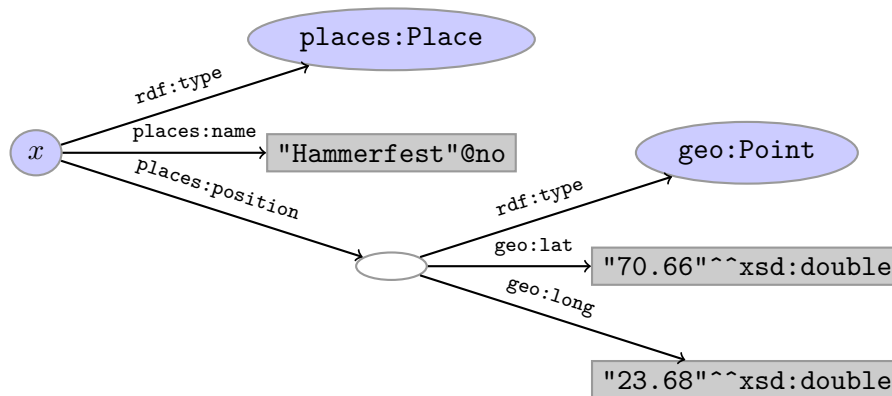
```
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix places: <http://inf3580.ifi.uio.no/places#>
```

```

@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>

```

For every place, you should give triples according to the following structure which shows information about Hammerfest:



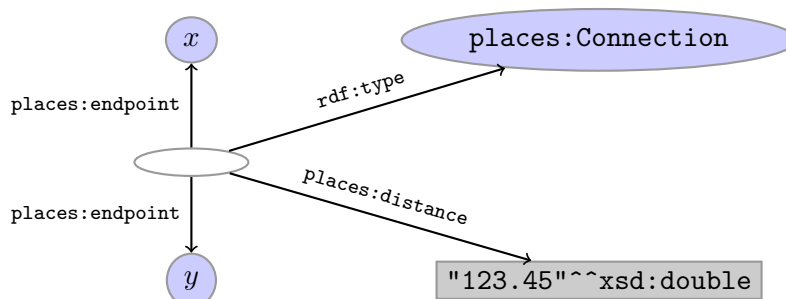
So you should use a blank node as illustrated between x and the location data. Which URIs x you choose to identify places is up to you. You could take them from geonames, from dbpedia, or invent your own.

It is up to you how you create this file. You can use a text editor to write it in Turtle format, or write a Java program that writes the file.

Deliverable: A file `places.ttl` containing the description of the ≥ 5 places in Turtle format.

Question 2

Write a Java program that reads a file containing the location of places, encoded as in Question 1, and adds information about the distance between places. Namely, for any pair of resources x and y with `rdf:type places:Place`, you should add nodes like this:



The literal (123.45 in the example) should give the distance between x and y in kilometres. You can download a file `Distances.java` from the course web site that contains a method `distanceDEG` that computes this distance, given the longitude and latitude of two points in degrees.

The program should write out the extended set of triples with distance information to another file in Turtle format. It should be possible to call the program like this:

```
java AddDistances places.ttl places-with-distances.ttl
```

Deliverable: The source code `AddDistances.java`

Question 3

For any three places x, y, z , it should hold that

$$d(x, y) + d(y, z) \geq d(x, z) \quad ,$$

known as the triangle inequality, where $d(x, y)$ is the distance between x and y .

Write a SPARQL query that checks that this holds by selecting all `?x ?y ?z` such that $d(x, y) + d(y, z) < d(x, z)$, given distance data in the format of Question 2. Executing the query on your data should of course find no such combination of places!

To test your query, you can either use the query execution program you wrote in the weekly exercises, or you can use a program like `Twinkle` (<http://www.ldodds.com/projects/twinkle/>).

Deliverable: A file `triangle.rq` containing the SPARQL query that finds any three places violating the triangle inequality.

Question 4

Explain in informal language (that is, no formal proofs or countermodels required), whether

1. the RDF graph in `places.ttl` from Question 1 *entails* that any resource of type `places:Place` has a value of type `geo:Point` for the property `places:position`.
2. whether it is consistent, according to `places.ttl`, to relate two different resources of type `places:Place` to the same `geo:Point`. That is, does such a graph have an RDF-interpretation?

Deliverable: A text file `semantics.txt` or a PDF file `semantics.pdf` with your answer.