M164/CS2 Knowledge Technologies

Fall 2018-2019

Homework I

Out: October 3, 2018

Due: November 6, 2018 at 24:00.

Total marks: 370

Exercise 1 (DBpedia)

The most central data source in the linked data cloud is DBpedia (http://wiki.dbpedia.org/), a big knowledge base which is essentially a "translation" of parts of Wikipedia into RDF. In this exercise you will become familiar with DBpedia by examining its contents and posing SPARQL queries. More specifically, you have to do the following:

- Become familiar with DBpedia by browsing its web site. Pay special attention to the DBpedia ontology (http://wiki.dbpedia.org/services-resources/ontology), which you will use to formulate your queries. Browse the Wikipedia knowledge captured by DBpedia starting from a resource that you know well e.g., the writer Nikos Kazantzakis (http://dbpedia.org/page/Nikos_Kazantzakis) and following links to other DBpedia resources. Do the example queries given in http://wiki.dbpedia.org/OnlineAccess
- Use the public SPARQL endpoint over the DBpedia data set at http://dbpedia.org/sparql to pose the following queries:
    - Find all Greek wines known by DBpedia and the region of Greece where they are produced.
    - Find all the prime ministers of Greece known to DBpedia. Output their name, the party or parties they have been members of and the University (-ies) that they have graduated from.
    - Find all the Greek universities known to DBpedia. Output their name, the city that they are located and the number of prime ministers of Greece that have graduated from them (order answers by this number).

[30 marks]

Exercise 2 (Querying the Greek administrative geography dataset using SPARQL)

Our group has initiated the development of a linked open data portal of interest to Greece (http://www.linkedopendata.gr/). In the context of this effort, we have developed an ontology and a corresponding dataset for the new administrative system of Greece known as the Kallikratis plan (http://en.wikipedia.org/wiki/Administrative_divisions_of_Greece). This exercise involves posing SPARQL 1.1 queries against this ontology and dataset.

First, we ask you to understand the Kallikratis ontology `gag-ontology.rdf` given at the Web page of the exercises for the course (http://cgi.di.uoa.gr/~pms509/projects.htm). See also the brief documentation available there.

Then consider the dataset for Kallikratis given at the same Web page. Load the dataset in Sesame and use SPARQL 1.1 to express the following queries:

- Give the official name and population of each municipality (δήμος) of Greece.
- For each region (περιφέρεια) of Greece, give its official name, the official name of each regional unit (περιφερειακή ενότητα) that belongs to it, and the official name of each municipality (δήμος) in this regional unit. Organize your answer by region, regional unit and municipality.
- For each municipality of the region Peloponnese with population more than 5,000 people, give its official name, its population, and the regional unit it belongs to. Organize your answer by municipality and regional unit.
- For each municipality of Peloponnese for which we have no seat (έδρα) information in the dataset, give its official name.
- For each municipality of Peloponnese, give its official name and all the administrative divisions of Greece that it belongs to according to Kallikratis. Your query should be the simplest one possible, and it should not use any explicit knowledge of how many levels of administration are imposed by Kallikratis.
- For each region of Greece, give its official name, how many municipalities belong to it, the official name of each regional unit (περιφερειακή ενότητα) that belongs to it, and how many municipalities belong to that regional unit.
- Check the consistency of the dataset regarding stated populations: the sum of the populations of all administrative units A of level L must be equal to the population of the administrative unit B of level L+1 to which all administrative units A belong to. (You have to write one query only.)
- Give the decentralized administrations (αποκεντρωμένες διοικήσεις) of Greece that consist of more than two regional units. (You cannot use SPARQL 1.1 aggregate operators to express this query.)

[150 marks]

Exercise 3 (Greek administrative geography and GeoNames)

GeoNames (http://www.geonames.org/) is a gazetteer that collects both geospatial and thematic information for various placenames around the world. GeoNames data is available through various Web services but it is also published as linked data (http://www.geonames.org/ontology/documentation.html).

In order to make accessible to users of the Kallikratis dataset mentioned above, the rich geographical information held by Geonames, we have interlinked the two datasets by creating `owl:sameAs` links for each municipality in the Kallikratis dataset. For example, the assertion

```
<http://geo.linkedopendata.gr/gag/id/9325>          owl:sameAs
<http://sws.geonames.org/8133762/>
```

links the Kallikraties entity "ΔΗΜΟΣ ΧΑΝΙΩΝ" with the same entity in the Geonames dataset. Your job in this exercise is to use these links to answer the queries given below. You should use the SPARQL endpoint for the Kallikratis dataset available at http://geo.linkedopendata.gr/gag-endpoint/ (instead of your own Sesame deployment as you did in Exercise 3) and try its various ways of returning answers to your queries.

- Find all information that Geonames has for "Dimos Chania" (you have to use only Geonames here, not the Kallikratis dataset).
- Find all information held by Geonames for municipalities in the regional unit of Chania (περιφερειακή ενότητα Χανίων).
- For every municipality of the region of Crete according to Kallikratis, find its population and its population given by Geonames. Is the population information in the two datasets the same? Discuss the quality of the results.
- What kind of hierarchical administrative information for Greece is provided by Geonames and how does it compare to the Kallikratis dataset? Explain your answer using appropriate SPARQL queries on the joint datasets and their results.

[40 marks]

Exercise 4 (http://schema.org)

As we have discussed in class, http://schema.org is a major effort from the top search engine companies (Google, Bing, Yahoo and Yandex) to help web designers annotate their pages with structured information which can then be used by search engines for better indexing of these web pages. You can read about this effort at https://developers.google.com/search/ and http://schema.org/ .

As you can see http://schema.org/ provides an ontology for annotating Web pages. This exercise asks you to write queries that navigate this ontology and are evaluated using RDFS reasoning. First browse the ontology starting from the page

https://schema.org/docs/schemas.html. You should also read about the data model and other information about this ontology at http://schema.org/docs/documents.html. Then download the latest version of the core ontology from https://schema.org/version/3.1/, store it in a Sesame repository that supports inferencing, and use SPARQL 1.1 to express the following queries:

- Find all subclasses of class CollegeOrUniversity (note that http://schema.org/ prefers to use the equivalent term "type" for "class").
- Find all the superclasses of class CollegeOrUniversity.
- Find all properties defined for the class CollegeOrUniversity together with all the properties inherited by its superclasses.
- Find all classes that are subclasses of class Thing and are found in at most 2 levels of subclass relationships away from Thing.
- Finally, express the above queries on the ontology and dataset but without the use of inferencing.

[50 marks]

Exercise 5 (Using http://schema.org to annotate Web pages)

Now that we have understood what http://schema.org/ is, let us use it to annotate the Web page of the professor giving this class (http://cgi.di.uoa.gr/~koubarak/). First of all read about this task on http://schema.org/docs/gs.html and familiarize yourself with the relevant technologies of Microdata, RDFa and JSON-LD. You can also see examples of using http://schema.org/ on the main web page of each of its elements (e.g., see examples of using the class CollegeOrUniversity at the bottom of the page https://schema.org/CollegeOrUniversity). Google recommends to use JSON-LD to annotate Web pages (see https://developers.google.com/search/docs/guides/intro-structured-data).

Your job in this exercise is to use the format of JSON-LD to prepare a new version of the Web page http://cgi.di.uoa.gr/~koubarak/ (but please use a different name!) and the Web pages found by following the links Teaching and Publications/Publications By Year (for the latter Web page you only need to encode information about 3 publications, again use different imaginary names of authors).

You should use the Google tool available at https://search.google.com/structured-data/testing-tool/u/0/ to verify your code. The JSON-LD code should be embedded in your HTML code to have fully functional Web pages which can be explored using your favorite browser.

[100 marks]