M164 Knowledge Technologies

Academic Year 2022-2023

Homework I

Out: October 10, 2022

Due: November 20, 2022 at 23:59.

Total marks: 330

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 (https://www.dbpedia.org/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 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 on the same Web page. Load the dataset in the rdf4j RDF store 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 (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 rdf4j RDF store 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 4

In this exercise, you will see how ontologies and KGs can be used for system diagnostics. For this exercise we focus on PEM fuel cells, which are a zero-emission, efficient and high-quality energy source that provides a future economically competitive option with respect to conventional energy sources. A drawback of these systems is that the failure of one component can cause multiple failures to the rest of the fuel cell stack, leading to a decrease in efficiency and significantly increasing maintenance and repair costs. Thus, it is crucial to notify early enough the end user of any forthcoming failures and to enable the performance of question answering.

To develop the question-answering system we need first to create a dataset with questions, queries, and answers. For this, we ask your help.

For this exercise, you are asked to do the following tasks:

i) Download the ontology PEMFC.owl http://cgi.di.uoa.gr/~pms509/projects.htm and you will need to translate in SPARQL 1.1 ten questions sent to you privately in piazza. You will run these queries in Protégé and verify manually the results. Can any of these questions be written in natural language using different words? If so, please provide an alternative question.

ii) Additionally, we ask you to write ten more NL questions that you believe that would be helpful for the diagnostics of the system and miss from the given list. Check if these questions can be answered from the ontology and, if not, indicate what is missing.

The report that we expect from you for this exercise is a spreadsheet with two tabs -one for each task- named: “Quadruples” (task (i)) and “NL Questions” (task (ii)). The “Quadruples” will have the following columns: Question, Paraphrase (optional), Query, Answer. The “NL Questions” will have two columns: Question, Missing Knowledge (where in the Missing Knowledge column you will add the classes and the properties that would enable the generation of the query).

[100 marks]