Knowledge bases projects - topics and requirements (summer semester 2020/2021)

1. Topics

A. Demonstration of queries and reasoning with GraphDB and SPARQL

You need to download and install the free version of GraphDB: https://ontotext.com/products/graphdb/.

Load the created RDF data set: Specifically for this task, do not try loading more than 1GB in GraphDB, as it takes a long time. (9GB take more than 4 hours just to save as data.) Write at least 10 SPARQL queries:

- at least 2 on classes or properties;
- at least 5 on individuals;
- at least 3 of them demonstrate the reasoning in the database, meaning that when reasoning is on there are more results extracted, compared to when reasoning is off;
- at least 3 use more than one concept in the knowledge base and use more than once '*FILTER';*
- at least 2 use set operations, such as *'UNION', 'MINUS', etc;*
- at least 2 use *'BIND'* for assigning variables.

In the project documentation, describe the RDF dataset you are using - main classes and properties, and if suitable - visualization on the graph. In your own words, describe the principles and methods of reasoning and inference that GraphDB has. For this, refer to the official documentation: http://graphdb.ontotext.com/documentation/standard/reasoning.html. Provide screenshots of query results in GraphDB.

You should deliver:

- Project documentation - .pdf, around 900 words, written in Bulgarian or English.
- All SPARQL queries listed in a .txt file.
- The ontology you are using (.owl, .rdf, .ttl).
- Bonus points: Demo video.

B. Using an RDF dataset and SPARQL for an applied task

Choose an applied task you want to use an ontology for. Such a task might be:

- building a recommender system;
- building a chatbot;
- text search engine;
- data enrichment for a data mining problem;
- organizing available information into a knowledge base;
- any other AI application you can think of.

Choose a topic and find an RDF dataset that may be useful for it. Find some papers or articles on the topic. Demonstrate how you can use RDF datasets and SPARQL queries for such a problem. Show how you can integrate SPARQL (and GraphDB, if you choose to use it) with your project. Include at least 3 relevant SPARQL queries.

This project can be realized both as a separate project and as an extension of another course project.

In the project documentation, describe the task, the ontology you are using, and how do you use it in your solution. Provide screenshots of the results of the SPARQL queries.

You should deliver:

- Project documentation - .pdf, around 900 words, written in Bulgarian or English.
- All SPARQL queries listed in a .txt file.
- The ontology you are using (.owl, .rdf, .ttl).
- Project code (preferably a Jupyter notebook, but anything would work).
- Bonus points: Demo video.

C. Your own topic

For this course, you can choose to work on your own topic in the field of knowledge bases. The topic should be approved by the course teaching team by **16/05/2021**.

2. Requirements

What ontology you can use?

For projects of type A and B, you can use:

- your own ontology;
- one of the example ontologies;
- an ontology you have found on the Internet.

However, you CAN'T use:

- the Art ontology we use in the course.

What software systems you can use?

For a project of type A, you should use GraphDB. For projects of types B or C, you can use whatever software system is best for your project's goals, as long as it is relevant to the content of this course.

What should you submit - and in what terms?

Defining the project topic (short description) - deadline: 12/05/2021.

Short description of the project (100 - 150 words, NO more than 2 pages with the introductory), which contains:

- name of the student; faculty number; bachelor/master's program;
- project type (Type A, B, C);
- name of the project (subject area);
- a short written description of the idea and the technologies with which it will be realized.

The description is formed as a separate file (.pdf format), whose name contains the faculty number of the student. Project descriptions can be in English or Bulgarian. The document should be submitted in Moodle before the deadline.

Submission of projects (full description, code, demo) - three days before the exam date

1. Full description:
   - Theoretical justification and understanding of the project - what principles and formalism for knowledge presentation the project uses. Demonstrate an understanding of the operation of inference mechanisms - what are the algorithms used for inference in your chosen knowledge base implementation? Please paraphrase the online sources you will use.
   - Indicate the problems you encountered in the implementation and what knowledge you gained in solving them. What conclusions have you reached?
   - Future development;
   - Bibliography.

If you need to refer to the code, please include only the significant parts of it in the document. No need to add it all to the Appendix.

The full description should be about 900 words. This is an approximate number.

2. Implementation - program code and/or queries.

3. Optional: Demo - here you can show your understanding of the implementation and theory in the project. Video presentation of the main points of the full description. If more than one student participates in a project, each participant must present the part they have implemented, show their contribution. The video should not be longer than 10 minutes. Here you have complete freedom, you can show a presentation or the documentation, or you can show code - as long as it is understandable, clear, accurate and on the topic. Advertise your project.

The full description and program code must be submitted to Moodle archived and the archive must contain the student's faculty number. The presentation can be on a separate link.

Project defense

On the date of the defense, based on the documentation and the demo presentations, we might ask questions, related to the implementation of the projects, and if there are irregularities, you will have the opportunity to correct them. Please be prepared to demonstrate your implementations on site. The order of project defense will be determined by the order of submission of projects. There will be a list of students, which we will publish 1 day before the exam, after reviewing all submitted projects.