Semantic Web (CSE632)

Assignment-1

Winter 2024 Max marks: 100

Instructions

- Each student has to work on this by themselves.
- Plagiarism checks and policies will be strictly enforced. If you use ChatGPT and similar tools, please clearly indicate in a ReadMe where and how they were used.
- Your zip folder should be named RollNo_Assignment1.zip.

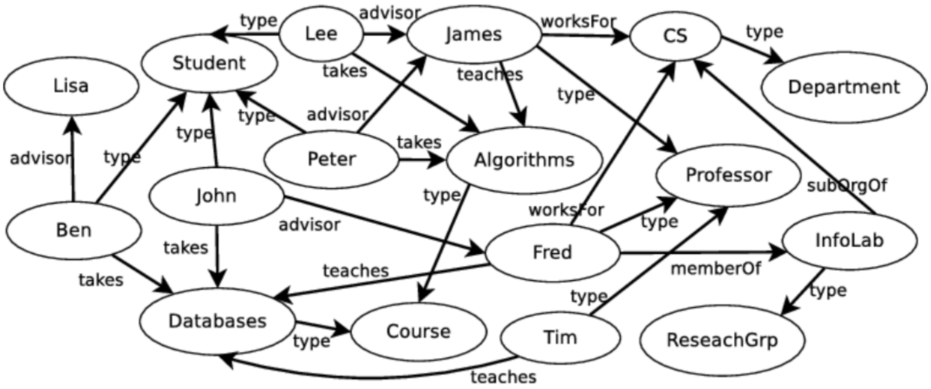

Figure 1: RDF Graph

1. **RDF Graph.** Translate the RDF Graph from Figure 1 into Turtle syntax (`https://www.w3.org/TR/turtle/`) and validate the triples (`https://www.w3.org/RDF/Validator/`). The IRI should start with `http://iiitd.ac.in/course/sweb/student/<rollno>`. Define the appropriate classes, instances, their relationships, and object values, choosing IRIs as appropriate. Define type/datatype for each object's values and domain, range for relationships. Use rdf:type for the "type" relation in the figure. The triples can be constructed either manually or by using an RDF library such as rdflib (`https://github.com/RDFLib/rdflib`), Apache Jena (`https://jena.apache.org/documentation/rdf/index.html`), RDF4J (`https://rdf4j.org/`). It is preferable to use a library. The turtle file will already be validated. Please submit the following. [20]

- Turtle file.
- Source code, in case a library is used.

2. **Entailment Rules.** There are 15 RDF entailment rules mentioned in the slides in the classroom (slides 50-64). Pick any 10 rules and implement them using the general purpose rule engine of Apache Jena (`https://jena.apache.org/documentation/inference/#rules`). Apply them (using the Jena API) on the turtle file from Question 1. Generate all possible triples using the entailment rules. In the report, for each entailment rule, mention at least one example of the triples from the turtle file and inferences generated using the rule. Please submit the following. [40]

   - Rules file.
   - Java code for applying the rules and generating the inference.
   - A report with the description of the generated inferences.

3. **RDF-star and Visualization.** The triples from rdf-star_ext_ikgrc2023.nt available at `https://github.com/aistairc/KGRC-RDF-star/` are categorized as follows.

   - Standard triple
   - Single-nested QT
   - Double-nested QT
   - Triple-nested QT
   - Quadruple-nested QT

   From each of the four categories of quoted triples (RDF-star), randomly pick 10 triples. Select any two reification techniques (Lecture 3; slides 42-45) and convert the RDF-star triples to those forms using either Apache Jena or rdflib. Load these triples into Neo4j (`https://neo4j.com/`) using the techniques suggested at `https://neo4j.com/docs/getting-started/data-import/`. Visualize them using any of the options suggested at `https://neo4j.com/docs/getting-started/graph-visualization/graph-visualization/`. Please submit the following. [40]

   - Java/Python code to parse rdf-star_ext_ikgrc2023.nt and convert them into two other reification forms.
   - Screenshots showing the loading of data into Neoj4 and data visualization for each of the two reification forms.
   - A report capturing the reification techniques used, the mechanism to load the data, and any other details/assumptions made.