RDF(S)

Michal Med

October 1, 2020

1 Background

This seminar will be devoted to the RDF(S) model. Please refer to lecture 2 for details on RDF(S).

To ensure data you put into your repository will be dereferenceable, everyone will use IRIs of the form `http://onto.fel.cvut.cz/ontologies/REPOSITORYNAME/WHATEVERYOUWANT`, where

`REPOSITORYNAME` is the name of the repository in GraphDB.

`WHATEVERYOUWANT` is a local identifier, according to your local identification scheme.

For example, in a repository named `testrepo`, we would like to create a new reference to a person John Doe. We decide to represent the IRI as `http://onto.fel.cvut.cz/ontologies/testrepo/person/doe-john` and the class person as `http://onto.fel.cvut.cz/ontologies/testrepo/person`. Note, that this is not the only option and it is a matter of design decision, how a IRI is constructed, e.g.

generic identification scheme creates unified IRIs for all individuals, another for all IRIs, e.g. `http://onto.fel.cvut.cz/ontologies/testrepo/object-1`,

class-prefixed identification scheme creates unified IRIs for all individuals of a particular class, e.g. `http://onto.fel.cvut.cz/ontologies/testrepo/person-1`,

class-related identification scheme creates unified IRIs for all individuals of a particular class, e.g. `http://onto.fel.cvut.cz/ontologies/testrepo/person/1`.

2 GraphDB repository

Each student has repository in GraphDB triplestore located at `http://onto.fel.cvut.cz:7300/`. Log in with your username. Default password is set to your username, too.

3 Excercises

Ex. 1 — Open a Turtle editor at `http://onto.fel.cvut.cz/turtle-editor` and explore the default turtle document. Take a look at its graphical view as well. Delete german labels from all resources, producing a valid turtle document.

Ex. 2 — Consider the RDF graph G in Figure 1.

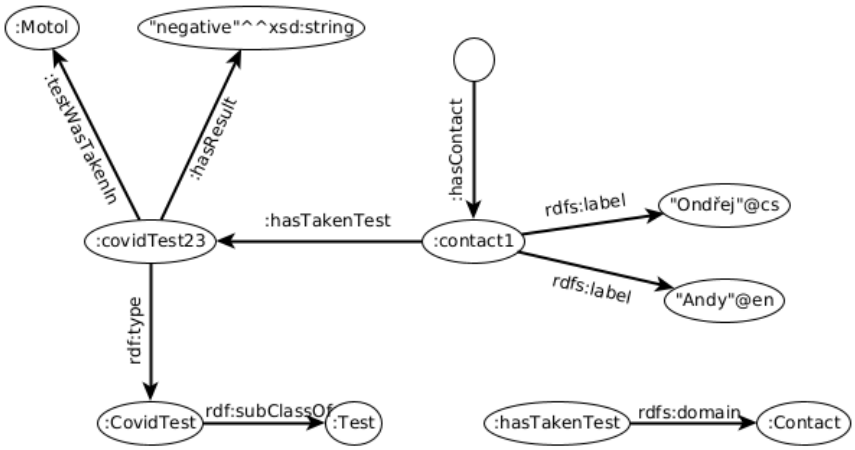

Figure 1: An example RDF graph

1. explain in one sentence what information does the graph contains,
2. decide, whether G is ground,
3. decide, whether G is lean, if no, simplify it, so that it becomes lean,
4. rewrite the graph into the Turtle syntax, use your default namespace as : (check validity in turtle editor),
5. save turtle graph into .ttl file and upload it into your GraphDB repository,
6. which triples are entailed by G under simple entailment,
7. which triples are entailed by G under RDF entailment,
8. which triples are entailed by G under RDFS entailment,
9. write a statement describing that the information about result of Ondřej's test was provided by a person with IRI `:LabRatTom`.

Ex. 3 — Create an RDF document in Turtle syntax, representing the following knowledge. Define your own IRIs for named resources. Try to express every bullet with one expression:

- Peter lives in the red house,
- White house and red house have the same (unknown) delivery person,
- Inhabitant of the white house is 165 cm tall.

Ex. 4 — Create a schema document to the previous example, formalizing the knowledge about people – namely classes `Person`, `Inhabitant`, `DeliveryPerson`, `House`, and properties `lives-in`, `has-inhabitant`, `delivers-to`. Try to express as much knowledge about these classes/properties, as possible, using RDF Schema 1.1 constructs.

Ex. 5 — Using a text editor, create an RDF document (in Turtle) with your public RDF profile (i.e basic data, your interests, etc.). Use FOAF vocabulary (`http://xmlns.com/foaf/spec/`), where possible. Upload this file to your GraphDB repository.

4 Relevant References

- RDF Validator – `http://www.w3.org/RDF/Validator/`
- Any23 (transformation between RDF formats) – `http://any23.org/`
- FOAF – `http://xmlns.com/foaf/spec/`