Exercise Sheet 10

Knowledge Repositories

Submit your solutions until ***Monday, 15.05.2016, 12h00*** *by uploading them to ILIAS. Later submissions won't be considered. Every solution should contain the* ***name(s), email adress(es)*** *and* ***registration number(s)*** *of its (co-)editor(s).*

1 Knowledge Repositories' Contents (7 Points)

1.1 Procedural vs Declarative Knowledge (3 Points)

Explain the difference between procedural and declarative knowledge. Write down a short example for each of them.

1.2 Knowledge Representation (2 Points)

Humans usually express their knowledge in natural language. Why aren't we using, e.g., English for knowledge representation?

1.3 Terminological vs Concrete Knowledge (2 Points)

What is the role of terminological knowledge (e.g. *Every company is an organization.*), and what is the role of facts (e.g. *Microsoft is an organization. Microsoft is headquartered in Redmond.*) in a knowledge-based system?

Hint: Imagine a knowledge base that only contains facts (or terminological knowledge). What could a knowledge-based system do with it?

2 RDF (8 Points)

RDF graphs consist of *triples* having a *subject*, a *predicate* and an *object*. Different syntactic notations can be used in order to serialize RDF graphs. By now you have seen the *XML* and the *N-Tripes* syntax of RDF. In this task we will use the *Notation3* (N3) format described at http://www.w3.org/2000/10/swap/Primer. Look at the following N3 file:

```
@prefix model: <http://example.com/model1/> .
@prefix cdk: <http://example.com/chemistrydevelopmentkit/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .

model:atom1
      a       cdk:Atom ;
      cdk:hasFormalCharge "1" ;
      cdk:symbol "O" .

model:atom2
      a       cdk:Atom ;
      cdk:symbol "C" .

model:mol1
      a       cdk:Molecule ;
      rdfs:label "Methanol" ;
      owl:sameAs <http://rdf.openmolecules.net/?InChI=1/CH4O/c1-2/h2H,1H3> ;
      cdk:hasAtom model:atom2 ,
                  model:atom1 ;
      cdk:hasBond model:bond1 .

model:bond1
      a       cdk:Bond ;
      cdk:bindsAtom model:atom1 ,
                    model:atom2 ;
      cdk:hasOrder cdk:SingleBond .
```

2.1 RDF Graph (6 Points)

Draw the RDF graph resulting from this file. An example for an RDF graph can be seen at http://www.w3.org/2000/10/swap/Examples.html.

2.2 What You See Is What You Get (2 Points)

Describe the information you have about *model:mol1* in your own words.

3 Modelling (23 Poins)

In this exercise we will look into an RDF vocabulary and use it to create a model. The information will be taken from the following text about the band Metallica:

> Metallica is an American heavy metal band from Los Angeles, California. The band's fast tempos, instrumentals, and aggressive musicianship placed them as one of the founding "big four" of thrash metal alongside Slayer, Megadeth, and Anthrax. Metallica formed in 1981 when James Hetfield responded to an advertisement that drummer Lars Ulrich had posted in a local newspaper. As of 2003, the line-up features long-time lead guitarist Kirk Hammett (who joined the band in 1983) and bassist Robert Trujillo (a member since 2003) alongside Hetfield and Ulrich. Previous members of the band are lead guitarist Dave Mustaine (who went on to found the band Megadeth), and bassists Ron McGovney, Cliff Burton and Jason Newsted. The band also had a long collaboration with producer Bob Rock, who produced all of its albums from 1990 to 2003 and served as a temporary bassist between the departure of Newsted and the hiring of Trujillo. The band earned a growing fan-base in the underground music community and critical acclaim with its first four albums, with their third, Master of Puppets (1986), described as one of the most influential and "heavy" thrash metal albums. Metallica achieved substantial commercial success with their eponymous fifth album (also known as The Black Album), which debuted at number one on the Billboard 200.

Source: https://en.wikipedia.org/wiki/Metallica

3.1 FOAF

One widely known RDF vocabulary is FOAF ("Friend of a Friend") which is useful for expressing metadata about people, and their interests, relationships and activities. You can find an introduction here http://www.xml.com/pub/a/2004/02/04/foaf.html and the complete specification here http://xmlns.com/foaf/spec .

a) (3 Points)

Look at the definitions of the following classes: *foaf:Group*, *foaf:Person*, *foaf:Organization*

Mark (e.g. underline) all relevant entities in the given text with regard to the foaf-classes mentioned above.

3.2 Creating a model

a) (15 Points)

Create an RDF model for the text above. Only use the classes in the FOAF vocabulary that were mentioned before and the following properties: *foaf:birthday*, *foaf:member*, *foaf:knows*. **Store your**

model in a validated XML file. To do so you can use the sevice at http://www.rdfabout.com/demo/validator/ that can translate between formats and validate the input. Another service you can use in order to create the RDF is http://turtled.net/[1]. Remember that each entity should be represented by an URI, e.g. http://dbpedia.org/page/Metallica or if no DBpedia entity exits you can introduce your own URI http://example.org/CustomEntity.

b) (5 Points)

What were problems you encountered when modeling the information in the text? Which information could not be modeled, e.g. because of missing classes and properties?

4 SPARQL (21 Points, 5 Bonus Points)

The DBpedia project is one of the biggest knowledge repositories available. As shown in the lecture you can test queries under http://dbpedia.org/snorql/ using SPARQL. An elaborate introduction to SPARQL can be found here: http://www.w3.org/TR/rdf-sparql-query/#WritingSimpleQueries

Test this query:

```
SELECT ?episodeName, ?season, ?chalkboardGag WHERE {
?episode <http://dbpedia.org/property/blackboard> ?chalkboardGag .
?episode <http://dbpedia.org/property/season> ?season .
?episode <http://dbpedia.org/property/episodeName> ?episodeName .
}
ORDER BY ?season
```

4.1 *The Simpsons*

a) (2 Points)

You might have noticed the PREFIX statements above the input field. Prefixes can be used to abbreviate the URLs in a query to make it more readable.

- Edit the query given above so that no URL is used directly any more.

b) (4 Points)

In order to work with RDF-based data, you need to learn about the things available for querying. DBpedia offers a search page http://dbpedia.org/fct/ where you can find out more about given entries. From the results of the previous query, look up at least one Simpson episode and find the information that enables you to do the following:

[1] This software uses yet another format to store RDF which is called *turtle* and is specified here: http://www.w3.org/TR/turtle/ . If you want to run *turtled* locally, the code is available at https://github.com/mhausenblas/turtled

- Extend the query to list stars that appeared in that episode

4.2 More Queries

a) (6 Points)

- Write a new query that shows all companies residing in Mannheim.
- Extend the query to list the *homepage* property of each company.
- Extend the query to list the *foundingYear* property of each company.

b) (9 Points)

1. Write a query that lists films along with their director and the director's year of birth. Limit the result to 10,000 entries and list them in alphabetical order of the directors' names.
2. List all films starring John Wayne along with their runtimes. Sort the result in descending order so that the longest film is the first and the shortest film is the last entry.
3. Edit the previous query so that only films with a runtime of more than 4,000 seconds are listed. *Hint*: You can apply a FILTER in the WHERE clause.

c) (5 Bonus Points)

Make up your own query. The result should show at least three parameters (e.g. the query above shows the parameters *episodeName*, *season* and *chalkboardGag*). Describe an application scenario where your query could be used.