Exercise Sheet 2

RDFS and SPARQL

*Submit your solutions until **Friday, June 2, 23h00** by by **emailing them via ILIAS (.zip attachment)**. 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).*

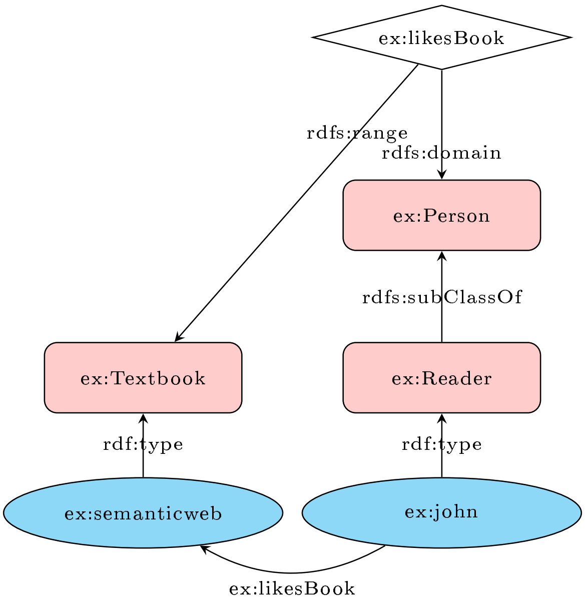
1 RDFS (20 points)

1.1 RDFS Formal Models (7 Points)

Show that the following triple

```
ex:machinelearning ex:likes ex:semanticweb.
```

is not entailed by the following RDFS ontology $\mathcal{O}_b$



by constructing **two** set-theoretical models $\mathcal{I}_1$ and $\mathcal{I}_2$ of $\mathcal{O}_b$, such that

- `ex:machinelearning ex:likes ex:semanticweb.` is true in $\mathcal{I}_1$
- `ex:machinelearning ex:likes ex:semanticweb.` is false in $\mathcal{I}_2$

Why is `ex:machinelearning ex:likes ex:semanticweb.` **not** entailed by $\mathcal{O}_b$?

1.2 Soundness and Completeness (8 Points)

Show that the following RDFS rules seen in course are **sound**:

1. RDFS range rule
2. RDFS domain rule
3. RDFS subclass transitivity rule
4. RDFS subproperty rule

1.3 RDFS Serialization (5 Points)

Write RDFS ontology $\mathcal{O}_b$ as a set of triples in: **(1)** turtle format (.ttl file) and **(2)** its corresponding XML (.rdf file) format. You can use the tools introduced in previous lectures and exercises (Protégé and the tools listed in Exercise 1).

2 SPARQL (20 Points)

The DBpedia project is one of the biggest knowledge repositories available. You can test SPARQL queries using http://dbpedia.org/snorql/ (alternative interface for DBpedia)[1].

2.1 (2 Points)

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
```

What can you say about it?

[1] An elaborate introduction to SPARQL can be found here: http://www.w3.org/TR/rdf-sparql-query/#WritingSimpleQueries.

2.2 *The Simpsons*

2.2.1 (2 Points)

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

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

2.2.2 (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:

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

2.3 More Queries

2.3.1 (6 Points)

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

2.3.2 (10 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.