

# EXERCISES WEEK 8 INF3580 SPRING 2012

Martin G. Skjæveland

06 March 2012

## 8 Model Semantics

Read

- Semantic Web Programming: chapter 4.
- Foundations of Semantic Web Technologies: chapter 3, 2.

### 8.1 Definitions

First, some notation and definitions collected from the lecture slides.

#### 8.1.1 Syntax: Triple abbreviations

Triple pattern	Triple instance	Abbreviation
indi prop indi .	$i_1 r i_2$	$r(i_1, i_2)$
indi rdfs:type class .	$i_1 \text{ rdfs:type } C$	$C(i_1)$
class rdfs:subClassOf class .	$C \text{ rdfs:subClassOf } D$	$C \sqsubseteq D$
prop rdfs:subPropertyOf prop .	$r \text{ rdfs:subPropertyOf } s$	$r \sqsubseteq s$
prop rdfs:domain class .	$r \text{ rdfs:domain } C$	$\text{dom}(r, C)$
prop rdfs:range class .	$r \text{ rdfs:range } C$	$\text{rg}(r, C)$

#### 8.1.2 Interpretation

An *interpretation*  $\mathcal{I}$  consists of:

- A set  $\Delta^{\mathcal{I}}$ , called the *domain*  $\mathcal{I}$
- For each individual URI  $i$ , an element  $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
- For each class URI  $C$ , a subset  $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
- For each property URI  $r$ , a relation  $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

#### 8.1.3 Validity in Interpretations (RDF)

Given an interpretation  $\mathcal{I}$ , define  $\models$  as follows:

- $\mathcal{I} \models r(i_1, i_2)$  iff  $\langle i_1^{\mathcal{I}}, i_2^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$
- $\mathcal{I} \models C(i)$  iff  $i^{\mathcal{I}} \in C^{\mathcal{I}}$

### 8.1.4 Validity in Interpretations, cont. (RDFS)

Given an interpretation  $\mathcal{I}$ , define  $\models$  as follows:

- $\mathcal{I} \models C \sqsubseteq D$  iff  $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $\mathcal{I} \models r \sqsubseteq s$  iff  $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$
- $\mathcal{I} \models \text{dom}(r, C)$  iff  $\text{dom } r^{\mathcal{I}} \subseteq C^{\mathcal{I}}$
- $\mathcal{I} \models \text{rg}(r, C)$  iff  $\text{rg } r^{\mathcal{I}} \subseteq C^{\mathcal{I}}$

## 8.2 Exercises

In these exercises use the notation and definitions above in your answers.

### 8.2.1 Exercise

Let  $\Gamma$  be the RDF graph below.

1. Create an interpretation  $\mathcal{I}_1$  such that  $\mathcal{I}_1 \models \Gamma$ .
2. Create an interpretation  $\mathcal{I}_2$  such that  $\mathcal{I}_2 \not\models \Gamma$ .
3. Create an interpretation  $\mathcal{I}_3$  such that  $\mathcal{I}_3 \models \Gamma$  and  $|\Delta^{\mathcal{I}_3}| = 1$ , i.e., the domain of the interpretation contains only one element.

```
1: @prefix : <http://www.example.org#> .
2: @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3: :Tweety rdf:type :Bird .
4: :Nixon rdf:type :Republican .
5: :Nixon rdf:type :Quacker .
6: :Nixon :listensTo :Tweety .
7: :Tweety :likes :Tux .
```

### 8.2.2 Exercise

Let  $\Gamma$  be the RDFS graph listed below.

1. Create an interpretation  $\mathcal{I}_1$  such that  $\mathcal{I}_1 \models \Gamma$ .
2. Create an interpretation  $\mathcal{I}_2$  such that  $\mathcal{I}_2 \not\models \Gamma$ .

```
1:
2: @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3: @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
4: @prefix owl: <http://www.w3.org/2002/07/owl#> .
5: @prefix : <http://example.org#> .
6: :Person a rdfs:Class .
7: :Man a rdfs:Class ;
8: rdfs:subClassOf :Person .
9: :Parent a rdfs:Class ;
10: rdfs:subClassOf :Person .
11: :Father a rdfs:Class ;
12: rdfs:subClassOf :Parent ;
13: rdfs:subClassOf :Man .
```

```

14:   :Child      a          rdfs:Class ;
15:         rdfs:subClassOf  :Person .
16:   :hasParent  a          rdf:Property ;
17:         rdfs:domain      :Person ;
18:         rdfs:range       :Parent .
19:   :hasFather  a          rdf:Property ;
20:         rdfs:subPropertyOf :hasParent ;
21:         rdfs:range       :Father .
22:   :isChildOf  a          rdf:Property ;
23:         rdfs:domain      :Child ;
24:         rdfs:range       :Parent .
25:   :Ann        a          :Person ;
26:         :hasFather       :Carl .
27:   :Carl       a          :Man .

```

### 8.2.3 Exercise

Let  $\Gamma$  be the RDFS graph entailments.n3. Show by way of model semantics the following claims:

1.  $\Gamma \models \text{:Father rdfs:subClassOf :Person .}$
2.  $\Gamma \not\models \text{:Ann a :Child .}$
3.  $\Gamma \models \text{:Ann :hasParent :Carl .}$
4.  $\Gamma \models \text{:Carl a :Person .}$
5.  $\Gamma \not\models \text{:Carl :hasChild :Ann .}$

### 8.2.4 Exercise

Let  $\Gamma$  be the RDFS graph entailments.n3. As we have seen in a previous week's exercises, using the standardised RDFS semantics the entailment  $\Gamma \models \text{:hasFather rdfs:domain :Person.}$  does not hold. Does it hold in our simplified semantics?

## 8.3 Querying with reasoning

### 8.3.1 Exercise

Write a program which extends the previous query program with the ability to read one or more models from file and query them with or without RDFS reasoning. The first parameter should be either endpoint or file, indicating if the model to be queried is an endpoint or file(s). If the first parameter given is endpoint, then the program should behave just as the previous program. If the first parameter is file then the program should treat all following parameters given, except the two last parameter, as URIs to RDF files, and collect them to one model, which is to be queried. The second to last parameter is the location of the SPARQL query. The last parameter is either true or false and indicated whether RDFS reasoning should be applied to the collected model prior to reasoning.

Running

```
java your_java_program file schema.rdf individuals.rdf query.rq true
```

should give you the answer of running the query `.rq` on the RDFS combined and inferred model of the files `schema.rdf` and `individuals.rdf`.

Running

```
java your_java_program file schema.rdf individuals.rdf query.rq false
```

should give you the answer of running the query `.rq` on the combined model of the files `schema.rdf` and `individuals.rdf`, but with no reasoning.

### **8.3.2 Exercise**

Run the query “Find everyone that has a mother...” which you have created in an earlier exercise on the Simpsons RDF file with and without reasoning. Compare and explain the two outputs.

### **8.3.3 Exercise**

Run the query “Is Herb the brother of Homer” on the Simpsons RDF file with and without reasoning. Compare and explain the two outputs.

### **8.3.4 Exercise**

Write a SPARQL query which answers the question “Who has Bart a family relationship to?”

Run the query both with and without reasoning and explain the results.

### **8.3.5 Exercise**

Write a SPARQL query that lists all men and women. Run the query both with and without reasoning and explain the results.

### **8.3.6 Exercise**

In the output in the previous exercise, from the query with reasoning enabled, is there someone missing, i.e., is there a person which is not classified as either a man or a woman? Why is that?

Is it possible to write a SPARQL query which lists all persons which are not either a man or a woman? Why / why not?

### **8.3.7 Exercise**

In these exercises the output results for SELECT queries with reasoning enabled almost always returns more results than when reasoning is disabled. For which query/queries is this not the case? Why is it so?