RDFS and reasoning

Read

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

1 From the lecture

a) Why can we not implement the model-theoretic semantic directly?

b) What 3 types of reasoning does RDFS support?

c) Give examples of some axoimatic triples in RDFS?

d) What are some limitations of RDFS reasoning?

e) What is forward chaining?

f) In which cases is it best to use forward chaining?

2 Entailment

In these exercises we will learn about entailment and decide the logical consequences of RDFS statements.

Let `entailments.n3` be the file listed below, where `rdf` and `rdfs` are the usual namespaces.

```
1   :Person     a               rdfs:Class .
2   :Man        a               rdfs:Class ;
3       rdfs:subClassOf     :Person .
4   :Parent     a               rdfs:Class ;
5       rdfs:subClassOf     :Person .
6   :Father     a               rdfs:Class ;
7       rdfs:subClassOf     :Parent ;
8       rdfs:subClassOf     :Man .
9   :Child      a               rdfs:Class ;
10      rdfs:subClassOf     :Person .
11  :hasParent a                rdf:Property ;
12      rdfs:domain         :Person ;
```

```
13        rdfs:range          :Parent .
14  :hasFather a                    rdf:Property ;
15      rdfs:subPropertyOf :hasParent ;
16      rdfs:range          :Father .
17  :isChildOf a                    rdf:Property ;
18      rdfs:domain         :Child ;
19      rdfs:range          :Parent .
20  :Ann       a                    :Person ;
21      :hasFather          :Carl .
22  :Carl      a                    :Man .
```

:CODE: :END:

2.1   Exercise

Is `entailments.n3` syntactically correct RDF(S)?

2.2   Exercise

Assuming the RDFS statements in `entailments.n3` are syntactically correct, are they semantically correct, i.e., do they give an accurate description of "the real world"?

2.3   Exercise

Explain what it means for one set of statements to entail a (different) set of statements.

3   Manual entailment calculation

In the following exercises decide if `entailment.n3` entails the statement(s) given and explain why/why not? If the answer is "yes, the statement(s) is entailed by `entailments.n3`", then use the simple entailment rules (`se1`, `se2`) and the rdfs entailment rules (`rdfs1`, ..., `rdfs12`) found at RDFS entailment rules to prove your answer. If the answer is "no", then explain, informally or formally, why this is so.

There are quite a few of these exercises, but many of them are quite easy so they should be quick to do. If they are too easy, then skip to the last ones, which are perhaps a bit harder.

3.1   Exercise

First, to get the an overview of the statements in `entailments.n3`, draw a diagram.

3.2   Exercise

```
:Father rdfs:subClassOf :Person .
```

3.3   Exercise

```
:Man rdfs:subClassOf :Person .
```

3.4   Exercise

```
:Carl a :Person .
```

3.5   Exercise

```
:Carl a :Parent .
```

3.6   Exercise

```
:Carl :hasChild :Ann .
```

3.7   Exercise

```
:Carl a :Man .
```

3.8   Exercise

```
:Carl a :Father .
```

3.9   Exercise

```
:Child rdf:type rdfs:Resource .
```

3.10   Exercise

```
:Ann a :Child .
```

3.11   Exercise

```
:Ann :isChildOf :Carl .
```

3.12   Exercise

```
:Ann :hasParent :Carl .
```

3.13   Exercise

```
:Ann :hasParent _:x .
```

3.14 Exercise

```
:Ann :hasParent [ rdf:type :Person ] .
```

3.15 Exercise

```
:hasFather rdfs:domain :Person .
```

3.16 Exercise

```
rdfs:range rdf:type rdfs:Resource .
```

3.17 Exercise

```
:hasFather rdfs:range :Father .
```

3.18 Exercise

```
:Father rdfs:subClassOf [ rdfs:subClassOf :Person ] .
```