3. RDFS Inference

The purpose of this lab is to build an RDFS inference engine using a rule system.

3.1 Introduction

RDFS reasoning can be realized by a set of rules [1] (See Figure 1). Therefore, RFDS reasoning can be implemented using a rule engine. Each RDF triple `xxx yyy zzz .` can be modeled as a (Datalog) fact.

```
triple(xxx, yyy, zzz)
```

The entailment patterns can be modeled as Datalog rules. For instance, entailment pattern `rdfs2` said that

> If *S* contains triples "aaa rdfs:domain xxx ." and "yyy aaa zzz .", then *S* RDFS entails the triple "yyy rdf:type xxx ."

It can be captured by the following Datalog rule

```
triple(Y, "rdf:type", X) :- triple(A, "rdfs:domain", X),
                            triple(Y, A, Z).
```

Together with the the following facts

```
  triple(":hasTripAdvisorRating", "rdfs:domain", ":TourismObject").
  triple("i:ChickenHut", ":hasTripAdvisorRating", "3.5") .
```

we can compute all the entailed triples using a Datalog engine like DLV[2].

```
$ ~/bin/dlv rdfs-2-example.dlv
DLV [build BEN+ODBC/Dec 17 2012   gcc 4.2.1 (Apple Inc. build 5666) (dot 3)]

{ triple(":hasTripAdvisorRating","rdfs:domain",":TourismObject"),
  triple("i:ChickenHut","rdf:type",":TourismObject"),
  triple("i:ChickenHut",":hasTripAdvisorRating","3.5") }
```

[1] http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/#rdfs_entailment
[2] http://www.dlvsystem.com/dlv/

RDFS entailment patterns.

	If S contains:	then S RDFS entails recognizing D:
rdfs1	any IRI aaa in D	aaa `rdf:type rdfs:Datatype .`
rdfs2	aaa `rdfs:domain` xxx . yyy aaa zzz .	yyy `rdf:type` xxx .
rdfs3	aaa `rdfs:range` xxx . yyy aaa zzz .	zzz `rdf:type` xxx .
rdfs4a	xxx aaa yyy .	xxx `rdf:type rdfs:Resource .`
rdfs4b	xxx aaa yyy.	yyy `rdf:type rdfs:Resource .`
rdfs5	xxx `rdfs:subPropertyOf` yyy . yyy `rdfs:subPropertyOf` zzz .	xxx `rdfs:subPropertyOf` zzz .
rdfs6	xxx `rdf:type rdf:Property .`	xxx `rdfs:subPropertyOf` xxx .
rdfs7	aaa `rdfs:subPropertyOf` bbb . xxx aaa yyy .	xxx bbb yyy .
rdfs8	xxx `rdf:type rdfs:Class .`	xxx `rdfs:subClassOf rdfs:Resource .`
rdfs9	xxx `rdfs:subClassOf` yyy . zzz `rdf:type` xxx .	zzz `rdf:type` yyy .
rdfs10	xxx `rdf:type rdfs:Class .`	xxx `rdfs:subClassOf` xxx .
rdfs11	xxx `rdfs:subClassOf` yyy . yyy `rdfs:subClassOf` zzz .	xxx `rdfs:subClassOf` zzz .
rdfs12	xxx `rdf:type rdfs:ContainerMembershipProperty .`	xxx `rdfs:subPropertyOf rdfs:member .`
rdfs13	xxx `rdf:type rdfs:Datatype .`	xxx `rdfs:subClassOf rdfs:Literal .`

Figure 1: RFDS Entailment Patterns

3.2 Tasks

- Convert the tourism RDFS ontology into a Datalog file (“tourism.dlv”).
- Implement rdfs2, rdfs3, rdfs5, rdfs7, rdfs9, rdfs11 in Datalog (“rdfs.dlv”)
- Test “tourism.dlv” and “rdfs.dlv” using DLV to check whether we can get all expected entailed triples.
- (Optionally) Implement other entailment patterns and create RDFS ontologies for testing.

3.3 Submission

- Two Datalog files “tourism.dlv” and “rdfs.dlv”
- A short report “description.txt”