

Information Integration on the Web: Homework 7

Due on March 6, 2015

Prof. Ambite & Knoblock

Tushar Tiwari

Turtle triples

Problem 1

Queries for 1, 3, 4 and 10

Solution

Question 1

```
BASE <http://dbpedia.org/resource/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
SELECT ?university, ?name
WHERE {
    ?university dbpprop:type <Private_university> .
    ?university dbpedia-owl:state <California> .
    ?university rdfs:label ?name .
}
```

Question 3

```
PREFIX schema: <http://schema.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia: <http://dbpedia.org/resource/>
CONSTRUCT {
    ?university rdf:type schema:CollegeOrUniversity .
    ?university schema:name ?name .
} WHERE {
    ?university rdf:type dbpedia:Private_university .
    ?university rdfs:label ?name .
}
```

Question 4

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX schema: <http://schema.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?university WHERE
{ ?university rdf:type schema:Organization }
```

Question 10

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX schema: <http://schema.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?person WHERE
{ ?person rdf:type schema:Person }
```

Problem 2

Description of tasks

Solution

Question 1

To fetch data from the dbpedia sparql endpoint (<http://dbpedia.org/sparql>) into our memory store a SPARQL Repository is used which is plugged with query1. The Memory store used for this is a regular Memory Store without Forward Chaining.

Question 2

The model that we will be using here will create data like this:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix schema: <http://schema.org/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
dbpedia:University_of_Southern_California a schema:CollegeOrUniversity ;
    schema:name "University of Southern California"@en,
               "University of Southern California"@de, ...
```

Question 3

Used query 3 with the construct query and evaluated it against the local triple store that has data from query 1. Stored the returned graph with the new schema in a new Memory Store (without forward chaining). Deleted the previous memory store with the old schema.

Question 4

Used query 4 and evaluated against the local triple store. Writing results to file. The output was an empty file because the results were empty. This is because there is no record that has `rdf:type schema:Organization`. Also, `schema:Organization` does not exist in the repository yet.

Question 5

Downloaded the NTriples file from schema.org for the schema and loaded it into the local triple store which already consists of information about universities.

Question 6

Evaluated query 4 again. Again the output file was empty. This was because, although there exists a record which describes the subclass relationship between `Organization` and `CollegeOrUniversity`, there is no URI that has `rdf:type Organization` because the relationship is yet to be inferred for records that have `rdf:type CollegeOrUniversity`.

Question 7

Created a new Memory Store with forward chaining enabled. Copied all records from previous memory store to the fc enabled memory store. Now we maintain two repositories: memory store and fc memory store.

Question 8

Evaluated query 4 against the fc memory store. The output file contains all the university URIs. This is because the forward chaining inferencer adds new records to the memory store by inferring the subclass relationships of CollegeOrUniversity and unifying the dbpedia:university entries with this subclass rule. Additional rdf:types are added for each CollegeOrUniversity rdf:type record namely EducationalOrganization, Organization and Thing.

Question 9

First synced the memory store and fc memory store such that they both contain the same records. Added the below record to both the repositories:

```
@prefix schema: <http://schema.org/> .  
@prefix dbpedia: <http://dbpedia.org/resource/> .  
dbpedia:University_of_Southern_California schema:alumni dbpedia:C._L._Max_Nikias .
```

Question 10

Evaluated query 10 against both the memory stores. The memory store without fc gave an empty file and the fc memory store gave the URI of dbpedia:C._L._Max_Nikias. The reason for this is without the forward chaining inferencer it cannot infer that the range of schema:alumni is rdf:type Person. Both the rule and the dbpedia:C._L._Max_Nikias entry exist as separate records with no relationship. In the fc memory store the two records are unified and hence the new record added to the memory store is:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix schema: <http://schema.org/> .  
@prefix dbpedia: <http://dbpedia.org/resource/> .  
dbpedia:C._L._Max_Nikias rdf:type schema:Person .
```