SIEMENS

Some remarks on Assignments 2+3

Assignment 2

- Is your FOAF-file lean?
  - If not, which triples can you remove to make it lean?
  - If yes, which triples could you add to make it non-lean?

- Which additional triples are entailed by your FOAF file under RDF entailment? (Give 5 triples as example).

- Which additional triples are entailed by your FOAF file **in combination with the FOAFontology** under RDFS entailment? (Give 5 triples as example).

- Which additional triples are entailed by your FOAF file **in combination with the FOAFontology** under D-entailment using the datatype map that includes rdf:XMLLiteral, xsd:integer, xsd:decimal, and xsd:string? (Give 5 triples as example, or argue why no further triples are entailed under this entailment regime).

Assignment 2

- Is your FOAF-file lean?
  - If not, which triples can you remove to make it lean?
  - If yes, which triples could you add to make it non-lean?

Answer:
Simple almost trivial example to make your foaf-file non-lean:

```
:me  a foaf:Person .
[] a foaf:Person .    (added redundant triple)
```

SIEMENS

Assignment 2

Which additional triples are entailed by your FOAF file under RDF entailment? (Give 5 triples as example).

Answer:

1) All the RDF axiomatic triples, e.g.

```
rdf:_1 a rdf:Property .
rdf:_2 a rdf:Property .
   …
```

2) For each property p you used:

p `a rdf:Property .`

e.g.

```
foaf:knows a rdf:Property .
```

SIEMENS

Assignment 2

Which additional triples are entailed by your FOAF file **in combination with the FOAFontology** under RDFS entailment? (Give 5 triples as example).

Answer: If you used

```
A foaf:knows B .
```

E.g. again, all the RDF and RDFS axiomatic triples count plus …You have entailed:

```
A a foaf:Person .
A a foaf:Agent .
A a rdfs:Resource .
foaf:knows a rdfs:Resource .
B a foaf:Person .
B a foaf:Agent .
B a rdfs:Resource .
```

SIEMENS

Assignment 2

Which additional triples are entailed by your FOAF file **in combination with the FOAFontology** under D-entailment using the datatype map that includes rdf:XMLLiteral, xsd:integer, xsd:decimal, and xsd:string? (Give 5 triples as example, or argue why no further triples are entailed under this entailment regime).

Answer:

Unless you have a typed literal with either of those types in your FOAF file, nothing new is entailed beyond what is already entailed under RDFS.

Assignment 3:

- *Datasets in SPARQL queries:* Take your own and at least one FOAF file of your colleagues (or, alternatively/additionally you can also use my FOAF file) and write SPARQL queries over the Dataset created combining these FOAF files (using **FROM** or **FROM NAMED** clauses) which answer the following questions:
  - Which friends (foaf:knows) to the owners of these foaf files have in common?
  - Which persons are only friends of yours but not mentioned as friends by any of the other FOAF files?
- *Querying DBPedia:* Answer the following questions using DBpedia's SPARQL endpoint:
  - The top 3 cities in Austria in terms of population number
  - Cities in Austria the name of which starts with the string "Unter"
  - Come up with an own query in DBPedia that produces duplicates in its results and explain in your own words why it does produce duplicates.
  - Come up with an own query in DBPedia that produces unbound values for some variables in some of its results in and explain in your own words why there are unbound values.

Assignment 3:

Which friends (foaf:knows) to the owners of these foaf files have in common?

Answer:
Assuming the only persons declared to know someone in foaf1.rdf and foaf2.rdf are the owners of the foaf files:

```
SELECT ?P
FROM <foaf1.rdf>
FROM <foaf2.rdf>
WHERE { ?P1 foaf:knows ?P . ?P2 foaf:knows ?P .
                  FILTER( ?P1 != ?P2 )
               }
```

Assignment 3:

Which friends (foaf:knows) to the owners of these foaf files have in common?

Answer:
Assuming the only persons declared to know someone in foaf1.rdf and foaf2.rdf are the owners of the foaf files:

```
SELECT ?P
FROM NAMED <foaf1.rdf>
FROM NAMED <foaf2.rdf>
WHERE {     GRAPH <foaf1.rdf> { [] foaf:knows ?P . }
            GRAPH <foaf2.rdf> { [] foaf:knows ?P . }
      }
```

Assignment 3:

Which persons are only friends of the owner of foaf1.rdf but not mentioned as friends foaf2.rdf?

An

swer (admittedly very ugly… we'll learn something easier today for SPARQL1.1):

```
SELECT ?P
FROM NAMED <foaf1.rdf>
FROM NAMED <foaf2.rdf>
WHERE {     GRAPH <foaf1.rdf> { [] foaf:knows ?P . }
            OPTIONAL {
              GRAPH <foaf2.rdf> { [] foaf:knows ?P2 .
                                  FILTER (?P = ?P2) }
              }
            FILTER !bound(?P2)
      }
```

Which persons are only friends of the owner of foaf1.rdf but not mentioned as friends foaf2.rdf?

This one doesn’t work:

```
SELECT ?P
FROM NAMED <foaf1.rdf>
FROM NAMED <foaf2.rdf>
WHERE {     GRAPH <foaf1.rdf> { [] foaf:knows ?P . }
            OPTIONAL {
              GRAPH <foaf2.rdf> { [] foaf:knows ?P . }
            }
            FILTER !bound(?P)
      }
```

SIEMENS

Assignment 3:

Which persons are only friends of the owner of foaf1.rdf but not mentioned as friends foaf2.rdf?

This one doesn't work either:

```
SELECT ?P
FROM NAMED <foaf1.rdf>
FROM NAMED <foaf2.rdf>
WHERE {     GRAPH <foaf1.rdf> { [] foaf:knows ?P1 . }
            GRAPH <foaf2.rdf> { [] foaf:knows ?P2 . }
            FILTER (?P1 != ?P2)
      }
```

Assignment 3:

The top 3 cities in Austria in terms of population number

Answer: Was more tricky than I thought … since

a) DBpedia endpoint seems to struggle with ORDER BY DESC

b) there's no consistently used category :CitiesInAustria or alike….

Some attempts how I got there… Method:

1) Look at suspect cities (e.g. Vienna, Graz, …) and their RDF data

2) Try to formulate the query with the properties and classes you find there…

```
SELECT DISTINCT ?C ?pop
   WHERE {
   ?C <http://dbpedia.org/ontology/populationTotal> ?pop ;
    <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/
   Austria> .

}
   ORDER BY ?pop
   LIMIT 3
```

SIEMENS

Assignment 3:

The top 3 cities in Austria in terms of population number

Answer: Was more tricky than I thought … since

a) DBpedia endpoint seems to struggle with ORDER BY DESC

b) there's no consistently used category :CitiesInAustria or alike….

Some attempts how I got there… Method:

1) *Look at suspect cities (e.g. Vienna, Graz, …) and their RDF data*
2) *Try to formulate the query with the properties and classes you find there…*

```
SELECT DISTINCT ?C ?pop
   WHERE {
   ?C <http://dbpedia.org/ontology/populationTotal> ?pop ;
    <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/
   Austria> .

}
   ORDER BY DESC ( ?pop )
   LIMIT 3
```

SIEMENS

Assignment 3:

The top 3 cities in Austria in terms of population number

Answer: Was more tricky than I thought … since

a) DBpedia endpoint seems to struggle with ORDER BY DESC

b) there's no consistently used category :CitiesInAustria or alike….

Some attempts how I got there… Method:

1) *Look at suspect cities (e.g. Vienna, Graz, …) and their RDF data*
2) *Try to formulate the query with the properties and classes you find there…*

```
SELECT DISTINCT ?C ?pop
   WHERE {
   ?C <http://dbpedia.org/ontology/populationTotal> ?pop ;
    <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/Austria> .
    [] <http://dbpedia.org/property/city> ?C .
   }
   ORDER BY DESC ( ?pop )
   LIMIT 3
```

SIEMENS

Assignment 3:

Cities in Austria the name of which starts with the string "Unter”

```
SELECT DISTINCT ?L
   WHERE {
   ?C rdfs:label ?L ;
    <http://dbpedia.org/ontology/country>
            <http://dbpedia.org/resource/Austria> .
   FILTER ( regex(str(?L),"^Unter"))
   }
```

Assignment 3:

- Come up with an own query in DBPedia that produces duplicates in its results and explain in your own words why it does produce duplicates.
- Come up with an own query in DBPedia that produces unbound values for some variables in some of its results in and explain in your own words why there are unbound values.

… we had some examples for those in the last lecture…

Assignment 3:

More questions on the assignment?

Individual feedback by the end of the week (sorry was traveling… your deal: no assignment today!)

Disclaimer:

I will not judge the assignments (particular Dbpedia queries) on whether they fully work and return "correct" results, but on whether I can see that you have tried! As you see: sometimes you have to work around limitations of SPARQL endpoints and available Linked Data datasets.