

SemWeb Assignment 2

Baudouin Duthoit
Student number : 2540566
VUnet ID : bdt230
Actualized on April 1, 2014

April 1, 2014

Contents

1	Ontology 10 points	2
1.1	1)	2
1.2	2)	2
1.3	3)	2
1.4	4) & 5)	2
1.5	6)	2
1.6	7)	3
2	SPARQL 9 points	4
2.1	1)	4
2.2	2)	4
2.3	3)	4
2.4	4)	4

1 Ontology 10 points

1.1 1)

Chose a domain for your ontology

I chose the furniture field for my ontology.

1.2 2)

Define two classes using restrictions as shown in lecture 3

"Outdoor_furnitures" and "Inside_furnitures" are my two classes. I also define some other classes like "Chair", "Table", and "Outside_furniture" as an equivalent class of "Outdoor_furniture".

1.3 3)

Here are the 3 instances for each class :

- Outdoor_furnitures

- ♡ plastic_table

- ♡ plastic_chair

- ♡ sunshade

- Inside_furnitures

- ♡ bed

- ♡ carpet

- ♡ armchair

1.4 4) & 5)

After loading in Protégé, the classes are in the right order, but even after reasoning, the instances don't go to the right classes. But when I query with Yasgui on my local fuseki server, the instances are right classified. Here is the Protégé's screen-shot.

1.5 6)

I use "owl:AllValuesFrom" and "owl:hasValue". The first one to get all furnitures and then restrict those by the second property which is the resistance against rain or not. See the turtle file beneath.

1.6 7)

An outdoor furniture is necessarily an instance of the furniture class too (owl:onProperty rdf:type; owl:asValue furniture). But to be sufficient too, it needs to resist against rain (owl:onProperty resistTo; owl:asValue db:Rain). That's how I describe outdoor furniture : furniture which resist against rain. To concluded, with the two properties, we get necessary and sufficient restriction to be an outdoor furniture.

2 SPARQL 9 points

2.1 1)

Perform 3 queries on <http://www.dbpedia.org> (6 points) Here are 3 queries I've made on Yasgui : <http://bit.ly/19hycKa> and <http://goo.gl/xuuGPn> and a construct query to finish <http://goo.gl/H3PAoz>

2.2 2)

Here are my queries on my ontology. The instances of my classes are well classified, as expected with reasoning.

The first query try to get all triples where "my:Inside_furniture" is the predicate. The second query show the first ten elements which are from type "?obj". The first triple of the query put some restrictions about "?obj" (here it has only one value).

The third query is a service query that return the first ten triples which have "rdfs:range" as predicate, and the same object and the same subject.

2.3 3)

To try without reasoning, I comment a few lines in the config-inf-tdb.ttl. After that I try a query I have already done on the local server to compare the results. In this screen-shot, it is the same query as the second one. There is only 4 results.

2.4 4)

The third query I made on the local server is a service query. It shows that we can select all data we want from an other data set, with conditions on it. Here the query asks for all the common tiples which as "rdfs:range" as predicate.