Semantic Web

Assignment 5

Dr. Jandson S Ribeiro
jandson@uni-koblenz.de

Isabelle Kuhlmann
iskuhlmann@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission by: June 6, 2021
Tutorial on: June 10, 2021

Please submit your solutions to your group's OLAT folder.
Always list all group members contributing to the solution!
Do not plagiarize from others!

For all the assignment questions that require you to code, make sure to include the code in the answer sheet, along with a separate Python file.

Team Name: Fowler

1. Uthara Ramanandan Kottaypilaprathodi : 220202020
2. Ibtesham Chowdhury : 220201351
3. Adigun Babatunde Ololade : 219203204

1 RDFS 5 points

Consider the following coffee domain specification:

> Coffee is worldwide a popular beverage made of roasted cocoa beans which are seeds from plants of the *Coffea* species. The roasting scale of cocoa beans goes from 1 to 10: level 1 being the least roasted, and 10 being the most roasted possible. There are many different kind of cocoa beans, some popular ones are: Arabica, Brazilian and Colombian. Coffee can be served black or with milk which can be steamed or not. Roasting level of the cocoa beans as well as the ratio of water to milk is relevant to the flavour of the coffee. Coffee can be prepared from many different techniques, for instance it can be filtered, french pressed, or high pressured (aka espresso).

1.1 2.5 points

Write a RDFS vocabulary for the coffee domain above. Your vocabulary must contain at least 5 concepts (i.e. Classes or Properties).

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:ex="http://www.example.org"

Vocabulary:
(ex:Coffee, rdf:type, rdfs:Class)
(ex:CocoaBeans, rdf:type, rdfs:Class)
(ex:Level, rdf:type, rdfs:Class)
(ex:Techniques, rdfs:type, rdfs:Class)
(ex:Black, rdfs:type rdfs:Class)
(ex:Milk, rdfs:type rdfs:Class)
(ex:hasRoastingScale, rdf:type, rdf:Property)
(ex:withTechnique, rdf:type, rdf:Property)
(ex:madeOf, rdf:type, rdfs:Property)
(ex:serve, rdf:type, rdfs:Property)
(ex:ratio, rdf:type, rdfs:Property)

RDFS:
(ex:hasRoastingScale, rdfs:range, ex:Level)
(ex:hasRoastingScale, rdfs:domain, ex:CocoaBeans)
(ex:withTechnique, rdfs:range, rdfs:Techniques)
(ex:withTechnique, rdfs:domain, ex:Coffee)
(ex:BeanKinds, rdfs:subClassOf, ex:CocoaBeans)
(ex:madeOf, rdfs:range, ex:BeanKinds)
(ex:madeOf, rdfs:domain, ex:Coffee)

(ex:serve, rdfs:domain, ex:Coffee)
(ex:Steamed rdfs:subClassOf ex:Milk)
(ex:NonSteamed rdfs:subClassOf ex:Milk)
(ex:ratio, rdfs:range, xsd:string)

1.2 2.5 points

Using the RDFS you specified in the previous step, write an RDF file that encodes the following information from a coffee shop. Your RDF document must use and comply with the RDFS you specified in the previous step.

> A coffee shop serves the following coffees: (1) Brazilian Espresso, espresso made of Brazilian coffee beans (roasting level of 8); (2) Special Arabica, french pressed coffee made of Arabica beans (roasting level of 6); and (3) Colombian Latte, made of Colombian cocoa beans (roasting level of 9) served with non-steamed milk (ratio water to milk of 1:3).

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ex="http://www.example.org"

ex: Brazilian Espresso rdf:type ex: Coffee
ex: Brazilian Espresso ex:madeOf ex: Brazillian
ex: Brazillian ex:hasRoastingScale ex:Level8
ex: Brazilian Espresso ex:withTechnique ex:Espresso

ex: SpecialArabica rdf:type ex: Coffee
ex: SpecialArabica ex:madeOf ex: Arabica
ex: Arabica ex:hasRoastingScale ex:Level6
ex: SpecialArabica ex:withTechnique ex:FrenchPressed

ex: ColombianLatte rdf:type ex: Coffee
ex: ColombianLatte ex:madeOf ex: Colombian
ex: Colombian ex:hasRoastingScale ex:Level9
ex: ColombianLatte ex:served ex:NonSteamed
ex: ColombianLatte ex:ratio 1:3

2 RDFs 5 points

Consider the following RDF/XML document containing information about a student:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xo="http://example.org/ontology#"
    xmlns:x="http://example.org/resource/">

    <rdf:Description
      rdf:about="http://example.org/resource/s1234">
        <rdf:type
          rdf:resource="http://example.org/ontology#BachelorStudent"/>
        <xo:name>Max Mustermann</xo:name>
        <xo:age>21</xo:age>
        <xo:studentID>12345678A</xo:studentID>
        <xo:studyProgram>Computer Science</xo:studyProgram>

        <xo:taughtBy>
            <xo:Professor
              rdf:about="http://example.org/resource/p4321">
                <xo:name>Anna Smith</xo:name>
                <xo:course>Semantic Web</xo:course>
            </xo:Professor>
        </xo:taughtBy>

        <xo:classmate>
            <xo:Person
              rdf:about="http://example.org/resource/s2345">
                <xo:name>Joe Average</xo:name>
            </xo:Person>
        </xo:classmate>
    </rdf:Description>
</rdf:RDF>
```

Specify a RDFS vocabulary for the above RDF.

xo:Person rdf:type rdfs:Class
xo:Professor rdf:type rdfs:Class
xo:BachelorStudent rdf:type rdfs:Class

x:s1234 rdf:type rdfs:Resource
x:p4321 rdf:type rdfs:Resource
x:s2345 rdf:type rdfs:Resource

xo:name rdf:type rdfs:Property
xo:course rdf:type rdfs:Property
xo:studentID rdf:type rdfs:Property
xo:studyProgram rdf:type rdfs:Property

xo:taughtBy rdf:type rdfs:Property
xo:taughtBy, rdfs:domain, ex:Professor
xo:taughtBy, rdfs:range, ex:BachelorStudent

xo:classmate rdf:type rdfs:Property
xo:classmate, rdfs:domain, ex:BachelorStudent
xo:classmate, rdfs:range, ex:BachelorStudent

3 OWL Ontology 5 points

Write an OWL ontology that models the domain described below. Use the FOAF vocabulary to help with the specification.

> Both beer and wine are popular beverages. A beverage has a product name, an expiration date, and is sold at at least one store. A wine may be from a distinct vineyard—for instance, the wine *Riesling Auslese "Theo" Gülser Bienengarten* is from the vineyard *Toni Müller*. A beer has a type, such as "Lager" or "Ale". For example, the beer *Barre Bräu* is of type *Pils*.
>
> Note that a beverage cannot be beer and wine at the same time.

```
Prefix(:=<http://xmlns.com/foaf/0.1/#>)
Prefix(xmlns:owl=<http://www.w3.org/2002/07/owl>)


Ontology( <http://www.example.com/ontology1>


Class(DisjointClasses(:Beer :Wine) partial :Beverages)
ObjectProperty(:hasName domain(owl:Thing) range(xsd:string))
ObjectProperty(:sold domain(owl:Thing) range(:Store))
ObjectProperty(:hasExpDate domain(owl:Thing) range(xsd:date))
ObjectProperty(:fromVineyard domain(:Wine) range(:Vineyard))
ObjectProperty(:type domain(:Beer) range(xsd:string))


Class(:Beverage complete
intersectionOf(
unionOf(:Beer :Wine)
restriction(:sold someValuesFrom(:Store))
:hasName
:hasExpDate))
)
```

4 Correcting Syntax Errors — 5 points

The following excerpt of an OWL ontology in XML contains errors. Correct the mistakes in the syntax in order to make it a valid OWL representation.

```
 1: <?xml version="1.0"?>
 2:
 3: <!DOCTYPE rdf:RDF [
 4:     <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
 5:     <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
 6:     <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
 7:     <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
 8: ]>
 9: <rdf:RDF xmlns="http://www.example.org/ontologies/pizzeria/pizzeria.owl#"
10:      xml:base="http://www.example.org/"
11:      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
12:      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
13:      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
14:      xmlns:owl="http://www.w3.org/2002/07/owl#">
15:
16: <owl:Ontology rdf:about="">
17: <owl:versionInfo>v.1.0</owl:versionInfo>
18: <owl:imports rdf:resource="foaf"/>
19: </owl:Ontology>
20: <owl:Class rdf:about="#Calzone">
21:  <owl:label>Calzone@en</owl:label>
22: </owl:Class>
23: <owl:DatatypeProperty rdf:ID="inventedIn">
24: <rdfs:type rdf:resource="owl:TransitiveProperty" />
25: <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
26: <rdfs:range rdf:resource="#Region" />
27: <rdfs:subClassOf>
28: <owl:Restriction>
29:  <owl:onProperty rdf:resource="#madeFromGrape" />
30: <owl:minCardinality
31: rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
32: </owl:Restriction>
33:  </rdfs:subClassOf>
34: </owl:DatatypeProperty>
35:  <rdf:RDF>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE rdf:RDF [
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
 ]>
<rdf:RDF xmlns ="http://www.example.org/ontologies/pizzeria/pizzeria.owl#"
 xml:base ="http://www.example.org/"
 xmlns:xsd ="&xsd;"
 xmlns:rdfs ="&rdfs;"
 xmlns:rdf ="&rdf;"
 xmlns:owl ="&owl;">

 <owl:Ontology rdf:about ="">
 <owl:versionInfo>v .1.0</owl:versionInfo>
 <owl:imports rdf:resource ="http://xmlns.com/foaf/0.1/"/>
 </owl:Ontology>

 <owl:Class rdf:about ="#Calzone">
 <owl:label> Calzone@en </owl:label>
 </owl:Class>

 <owl:DatatypeProperty rdf:ID ="inventedIn">
        <rdfs:type rdf:resource ="&owl;TransitiveProperty"/>
        <rdfs:domain rdf:resource ="&owl;#Thing"/>
        <rdfs:range rdf:resource ="#Region"/>
        <rdfs:subClassOf>
                <owl:Restriction>
                        <owl:onProperty rdf:resource ="#madeFromGrape"/>
                        <owl:minCardinality
                        rdf:datatype ="&xsd;nonNegativeInteger">1 </owl:minCa
                </owl:Restriction>
        </rdfs:subClassOf>
 </owl:DatatypeProperty>
 </rdf:RDF>
```

Important Notes

Submission

- Solutions have to be submitted to your group's OLAT folder.
- The name of the group and the names of all participating students must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the OLAT folder. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This pdfLaTeX template was adapted by Isabelle Kuhlmann based on the LuaLaTeX version by Lukas Schmelzeisen.

LaTeX

Use `pdflatex assignment_X.tex` to build your PDF.