QUESTIONS FROM THE COURSES

Day 04: questions from the course on RDFS.

Q4.1 Choose among the following assertions one or more you consider to be true:

- ☐ an ontology is necessarily formalized in first-order logic
- ☐ an ontology may allow inferences on data that uses it
- ☐ conceptual graphs can represent an ontology
- ☐ a shared ontology promotes interoperability
- ☐ description logics can represent an ontology

2, 3, 4, 5

Q4.2 RDFS contains primitives to (several answers possible)…

- ☐ describe classes of resources
- ☐ describe formulas of calculation for values of properties
- ☐ describe types of properties of resources
- ☐ document definitions in natural language
- ☐ sign and authenticate the authors of the definitions of classes and properties

1, 3, 4

Q4.3. What is defined and derived from these definitions?

```
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
@prefix : <http://inria.fr/devices#>
 :Phone rdfs:subClassOf  :Device .
 :Computer rdfs:subClassOf  :Device .
 :Smartphone rdfs:subClassOf :Computer .
 :Smartphone rdfs:subClassOf :Phone .
```

Introduce rdfs to define class.

Phone and computer is a subclass of device. Smartphone is a subclass of phone and computer.

→ Smartphone is a subclass of device.

Q4.4. What is defined and derived from these definitions?

```
@prefix rdfs:  < http://www.w3.org/2000/01/rdf-schema# >
@prefix : <http://inria.fr/member#>
:employeeOf rdfs:subPropertyOf  :proRelationWith .
:hasControlOver rdfs:subPropertyOf :proRelationWith .
:isShareholderOf rdfs:subPropertyOf :hasControlOver .
:isCEOof rdfs:subPropertyOf :employeeOf, :hasControlOver .
```

employeeOf is a sub property of proRelationWith.

hasControlOver is a sub property of proRelationWith.

isShareholderOf is a sub property of hasControlOver.

isCEOof is a sub property of both hasControlOver and employeeOf.

proRelationWith/ :isCEOof

employeeOf/ hasControlOver

isShareholderOf

Q4.5. What can be said about the types of the resources that will be linked by the properties defined below?

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix : <http://inria.fr/humans#>
:driverOf  rdfs:subPropertyOf  :isControling .
:piloteOf  rdfs:subPropertyOf  :isControling .
:isControling  rdfs:domain  :Human ;  rdfs:range :Object .
:driverOf rdfs:range  :Car .
:piloteOf rdfs:domain  :Adult ; rdfs:range  :Plane .
```

:driverOf is a sub property of :isControling.

:piloteOf is a sub property of :isControling .

:isControling has a domain :Human and a range :Object.

:driverOf has a range :Car .

:piloteOf has a domain :Adult and a range :Plane .

Q4.6. What could we add to this schema (several answers are possible)?

```
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
@base  <http://inria.fr/2005/humans.rdfs>
<p1>  a rdf:Property ;  rdfs:label  "age"@fr .
<c1>  a rdfs:Class; rdfs:comment "un être humain"@fr .
```

- ☐ <p1> rdfs:label "prénom"@fr .
- ☐ <c1> rdfs:comment "a human being"@fr .(is not french)
- ☐ <c1> rdfs:label "personne"@fr .
- ☐ <p1> rdfs:label "age"@en .
- ☐ <c1> rdfs:label "woman"@en .
- ☐ <c1> rdfs:label "persona"@es .

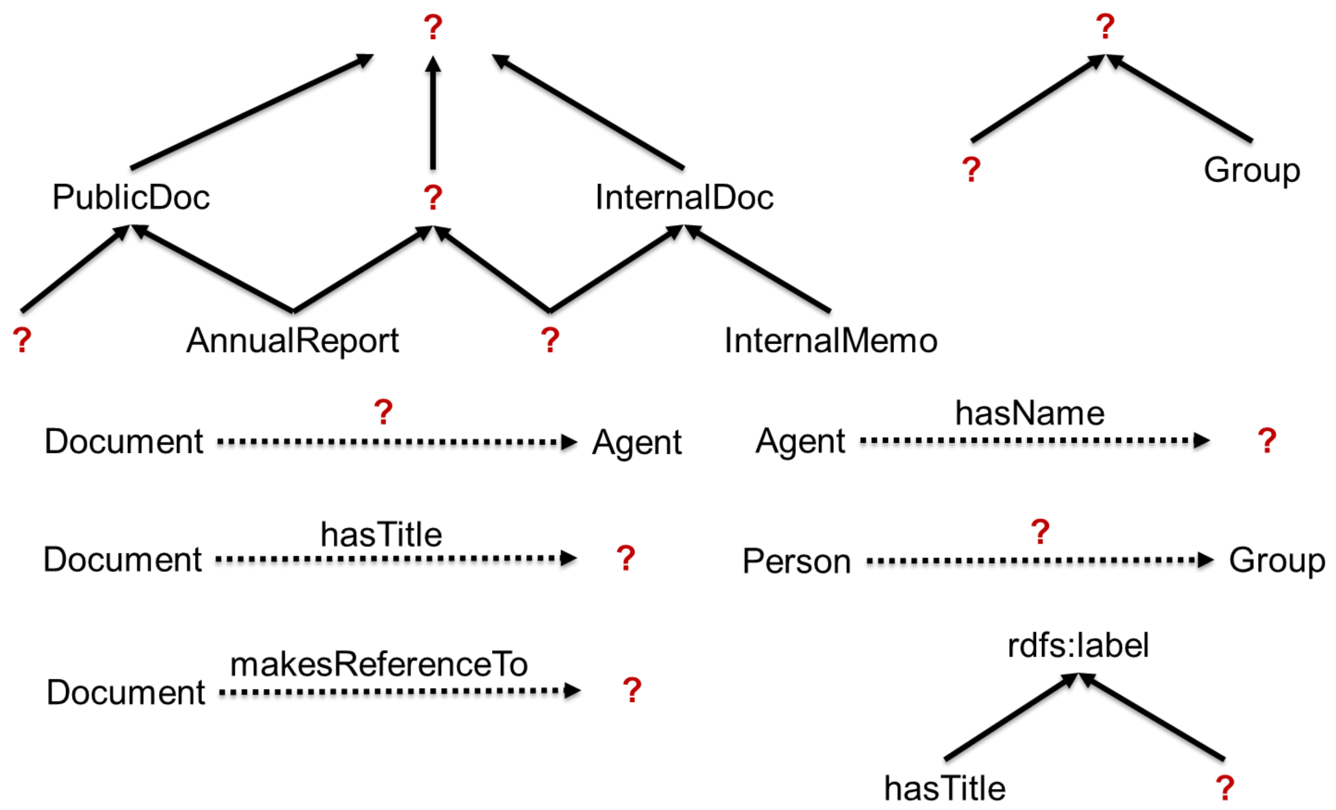
3, 4, 6

Q4.7. (a) Fill the blanks with: `Document, PublicDoc, PressArticle, Report, AnnualReport, InternalDoc, SecretReport, InternalMemo, Agent, Person, Group, hasTitle, hasAuthor, makesReferenceTo, hasName, isMemberOf` + rdf / rdfs primitives.
(b) Write it in RDFS and validate the RDF.

Document Agent

Report Person

PressArticle/ SecretReport



hasAuthor rdfs:Literal

rdfs:Literal isMemberOf

Document hasName

Day 04: questions from the course on OWL.

Q5.1 What can we deduce?

```
ex:Man owl:intersectionOf (ex:Male ex:Human) .
ex:Woman owl:intersectionOf (ex:Female ex:Human) .
ex:Human owl:unionOf (ex:Man ex:Woman) .
ex:Jane a ex:Human .
ex:John a ex:Man .
ex:James a ex:Male .
ex:Jane a ex:Female .
```

Jane is a woman.

John is a male and human.

James is just a male and nothing else we can discribe.

Q5.2 What are we defining and inferring?

```
@prefix ex: <http://example.org/>

ex:GrandFather rdfs:subClassOf [
  a owl:Class ;
  owl:intersectionOf ( ex:Parent ex:Man )
] .

ex:Jim a ex:Man, ex:Parent .
ex:Jack a ex:GrandFather .
```

GrandFather is a class and subclass, which is an intersection of a parent and man.

Jim is a man and a parent, but necessary to be a grandfather. Jack is a grandfather.

Q5.3 What can we deduce?

```
ex:hasSpouse a owl:SymmetricProperty .
ex:hasChild  owl:inverseOf ex:hasParent .
ex:hasParent rdfs:subPropertyOf ex:hasAncestor .
ex:hasAncestor a owl:TransitiveProperty .
ex:Jim  ex:hasChild  ex:Jane .
ex:Jane ex:hasSpouse ex:John .
ex:Jim  ex:hasParent ex:James .
```

Jane has a parent and an ancestor Jim.

John has a spouse Jane.

James has a child Jim. Jim has an ancestor James.

Jane has an ancestor James.

Q5.4 What can we deduce?

```
ex:Human owl:equivalentClass foaf:Person .
foaf:name owl:equivalentProperty ex:name .
ex:JimmyPage a ex:Human ;
           owl:sameAs ex:JamesPatrickPage .
ex:JimmyHendrix owl:differentFrom ex:JimmyPage .
```

Human and person are equivalent.

Name and name are equivalent.

JimmyPage is same as JamesPatrickPage. JimmyPage is a person.

JimmyHendrix is different from JimmyPage

Q5.5 What are we defining and inferring?

```
ex:UnluckyPerson owl:equivalentClass [
  a owl:Class ;
  owl:intersectionOf (
    ex:Person
    [ a owl:Class ; owl:complementOf ex:Lucky ]
  )
] .
```

Unlucky is a class. Person is a class.

To be an unlucky person should be a person opposite from lucky.

Q5.6 What can we deduce?

```
ex:Human rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty ex:hasParent ;
    owl:allValuesFrom ex:Human ] .
ex:Tom a ex:Human .
ex:Tom ex:hasParent ex:James, ex:Jane.
```

James and Jane will become human.

Q5.7 What are we defining and inferring?

```
@prefix ex: <http://example.org/>
ex:PersonList rdfs:subClassOf
  [
    a owl:Restriction ;
    owl:onProperty rdf:first ;
    owl:allValuesFrom ex:Person
  ] ,  [
    a owl:Restriction ;
    owl:onProperty rdf:rest ;
    owl:allValuesFrom ex:PersonList
  ] .

ex:value rdfs:range ex:PersonList .
ex:abc ex:value (ex:a ex:b ex:c) .
```

Each of a, b and c become a person.

Q5.8 What are we defining and inferring?

```
@prefix ex: <http://example.org/>
ex:Human rdfs:subClassOf [
  owl:intersectionOf (
    [
      a owl:Restriction ;
        owl:onProperty ex:hasBiologicalFather ;
        owl:maxCardinality 1
    ] , [
      a owl:Restriction ;
        owl:onProperty ex:hasBiologicalMother ;
        owl:maxCardinality 1
    ] )
] .
ex:Jane a ex:Human ;
        ex:hasBiologicalFather ex:James , ex:Jhon .
```

ex :James owl:sameas ex :Jhon

PRACTICAL SESSIONS

Day 04: Answers to the practical session on RDFS.

Software requirements

- The RDF XML online validation service by W3C: https://www.w3.org/RDF/Validator/
- The RDF online translator: http://rdf-translator.appspot.com/
- The SPARQL Corese engine: https://project.inria.fr/corese/

Create your own schema Family.rdfs

- Write the the RDF schema that you used in the description of Jen in a RDF/XML (or in turtle and then translate it) and save the RDF/XML in a file called "Family.rdfs". Of course, this assumes that the URIs for the classes and properties declared/used must match in both files. You mays have to update the files Jen.rdf and Jen.ttl to use your ontology.

Your schema:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

@base <http://www.unice.fr/voc>.


<#Man> a rdfs:Class ;

rdfs:subClassOf <#Human> .


<#Woman> a rdfs:Class ;

rdfs:subClassOf <#Human> .


<#hasFather> a rdf:Property ;

rdfs:subPropertyOf <#hasParent> ;

rdfs:domain <#Human> ;

rdfs:range <#Man>.
```

<#hasSpouse> a rdf:Property ;

rdfs:domain <#Human> ;

rdfs:subPropertyOf <#familyLink> .

<#hasChild> a rdf:Property ;

rdfs:domain <#Human> ;

rdfs:subPropertyOf <#familyLink> .

<#hasColleague> a rdf:Property ;

rdfs:domain <#Human> .

<#FamilyMember> a rdfs:Class ;

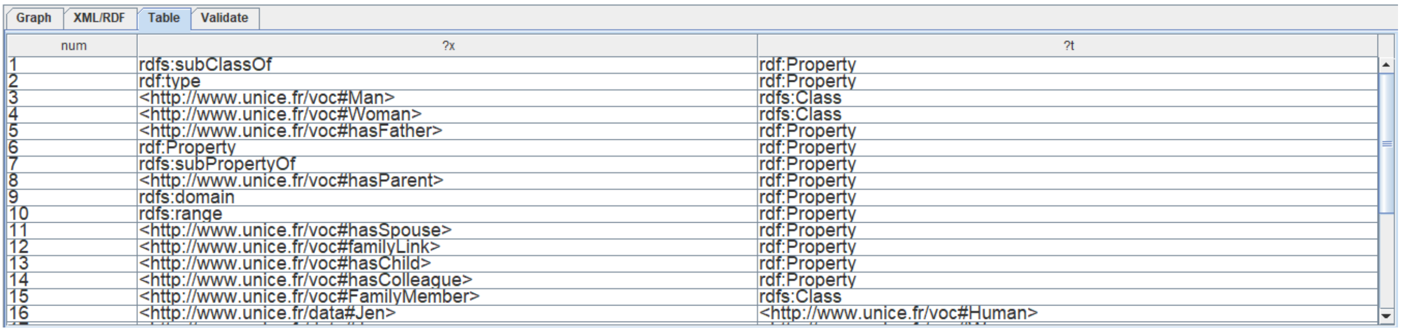
rdf:Property <#familyLink> .

- Check that your RDF schema and RDF files are valid using the W3C's RDF validation service.
- Launch the standalone interface of Corese and load your files Family.rdfs and Jen.rdf
- The interface contains a default SPARQL query:

  `Select ?x ?t where {?x rdf:type ?t}`

  Launch the query and look at the results.

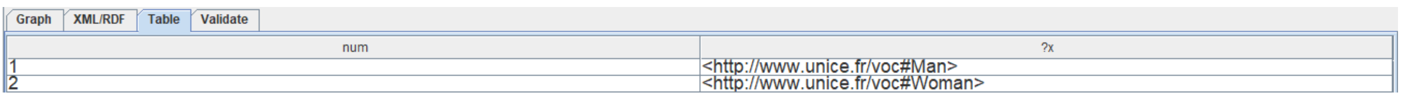
Screenshot:



num	?x	?t
1	rdfs:subClassOf	rdf:Property
2	rdf:type	rdf:Property
3	<http://www.unice.fr/voc#Man>	rdfs:Class
4	<http://www.unice.fr/voc#Woman>	rdfs:Class
5	<http://www.unice.fr/voc#hasFather>	rdf:Property
6	rdf:Property	rdf:Property
7	rdfs:subPropertyOf	rdf:Property
8	<http://www.unice.fr/voc#hasParent>	rdf:Property
9	rdfs:domain	rdf:Property
10	rdfs:range	rdf:Property
11	<http://www.unice.fr/voc#hasSpouse>	rdf:Property
12	<http://www.unice.fr/voc#familyLink>	rdf:Property
13	<http://www.unice.fr/voc#hasChild>	rdf:Property
14	<http://www.unice.fr/voc#hasColleague>	rdf:Property
15	<http://www.unice.fr/voc#FamilyMember>	rdfs:Class
16	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/voc#Human>

- Modify your ontology to declare the classes of Man and Woman as sub classes of Human (don't change the data), reload the schemas and data and search for the humans to see the results

Screenshot:



num	?x
1	<http://www.unice.fr/voc#Man>
2	<http://www.unice.fr/voc#Woman>

Explanation:

We can see thank there is two results, “Man” and “Woman”, with subclass “Human”.

- Modify your ontology to declare the properties hasChild and hasSpouse as sub properties of familyLink (don’t change the data), reload the schemas and data and search for the family links to see the results.

Screenshot:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/data#Seb>
2	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/data#Anny>
3	<http://www.unice.fr/data#Jen>	<http://www.unice.fr/data#Steffen>
4	<http://www.unice.fr/data#Seb>	<http://www.unice.fr/data#Anny>
5	<http://www.unice.fr/data#Seb>	<http://www.unice.fr/data#Steffen>

Explanation:

It shows five links between two people get familyLink.

- Modify your ontology to declare the class FamilyMember and use it to specify the signature of the property familyLink (don’t change the data) then reload the schemas and data and search for the family members.

Screenshot:

Graph | XML/RDF | Table | Validate

num	?x
1	<http://www.unice.fr/data#Jen>
2	<http://www.unice.fr/data#Seb>
3	<http://www.unice.fr/data#Anny>
4	<http://www.unice.fr/data#Steffen>

Explanation:

It shows all the members in the previous five links.

About the human.rdfs schema

1. If you don’t have the human schema file yet, download the RDF schema available at this address and save it as “human.rdfs”: http://wimmics.inria.fr/doc/tutorial/human_2013.rdfs
2. What is the namespace associated with this ontology? How was it associated?

<http://www.inria.fr/2007/09/11/humans.rdfs>

base="http://www.inria.fr/2007/09/11/humans.rdfs"

3. Look at the XML structure of this file and locate different syntactic properties: the different possible uses of the markup (ex: opening tag and closing, single tag), the use of namespaces for qualified names, the use of entities, etc.
4. Locate the use of the terms of the RDF (S) language: Class, Property, label, comment, range, domain, subClassOf, subPropertyOf, etc. To what namespaces are they associated?

XML

5. What are the classes of resources that can have the age property? Explain

There is no.

<rdf:Property rdf:ID="age">

<label xml:lang="en">age</label>

<label xml:lang="fr">âge</label>

<comment xml:lang="en">complete existence duration.</comment>

<comment xml:lang="fr">durée complète d'existence.</comment>

</rdf:Property>

6. Look at the beginning of the file and draw the subgraph of the hierarchy containing the classes `Animal`, `Man` and `Woman`.

Drawing of hierarchy:

Man and Woman is the subclass of Animal.

Query the schema itself

Reset or relaunch the standalone Corese search engine interface and load the file `human`**`.rdfs`** (and only this one).

1. Write a query to find all the classes of the ontology.

query:

Graph | XML/RDF | Table | Validate

num	?x	?p
1	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>	rdf:type
2	<http://www.inria.fr/2007/09/11/humans.rdfs#Male>	rdf:type
3	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>	rdf:type
4	<http://www.inria.fr/2007/09/11/humans.rdfs#Man>	rdf:type
5	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	rdf:type
6	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>	rdf:type
7	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>	rdf:type
8	<http://www.inria.fr/2007/09/11/humans.rdfs#Woman>	rdf:type

2. Write a query to find all the links subClassOf in the ontology.

query:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs#Male>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
2	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
3	<http://www.inria.fr/2007/09/11/humans.rdfs#Man>	<http://www.inria.fr/2007/09/11/humans.rdfs#Male>
4	<http://www.inria.fr/2007/09/11/humans.rdfs#Man>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
5	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
6	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
7	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
8	<http://www.inria.fr/2007/09/11/humans.rdfs#Woman>	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>
9	<http://www.inria.fr/2007/09/11/humans.rdfs#Woman>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>

3. Write a query to find the definitions and translations of "shoe size" (*other* labels and comments in different languages for the resource labeled "shoe size").

query:

prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

```
select * where {

   h:shoesize ?p ?y

}
```

answers:

Graph | XML/RDF | Table | Validate

num	?p	?y
1	rdf:type	rdf:Property
2	rdfs:comment	"express in some way the approximate length of the shoes for a person."@en
3	rdfs:comment	"taille, exprim曚e en points, des chaussures d'une personne."@fr
4	rdfs:domain	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
5	rdfs:label	"shoe size"@en
6	rdfs:label	"size"@en
7	rdfs:label	"pointure"@fr

4. Write a query to find the synonyms in French of the word 'personne' in French (*other* labels in the same language for the same resource/class/property). What are the answers?

query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select ?x ?y where{

?x rdfs:label "personne"@fr ; rdfs:label ?y.

filter (?y != "personne"@fr)

}
```

answers:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	"homme"@fr
2	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	"human"@en
3	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	"human being"@en
4	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	"person"@en
5	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	"humain"@fr
6	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>	"礙tre humain"@fr

5. Write a query to find the different meaning of the term "size" (disambiguation using the different comments attached to different resources/classes/properties having the label "size"). What are the answers?

query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

   ?x rdfs:label "size"@en; rdfs:label ?y

}
```

answers:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	"shirt size"@en
2	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	"size"@en
3	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	"taille"@fr
4	<http://www.inria.fr/2007/09/11/humans.rdfs#shirtsize>	"taille de chemise"@fr
5	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	"size"@en
6	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	"shoe size"@en
7	<http://www.inria.fr/2007/09/11/humans.rdfs#shoesize>	"pointure"@fr
8	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	"size"@en
9	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	"taille"@fr
10	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	"trousers size"@en
11	<http://www.inria.fr/2007/09/11/humans.rdfs#trouserssize>	"taille de pantalon"@fr

6. Write a query to find the properties that use the class Person in their signatures?

query:

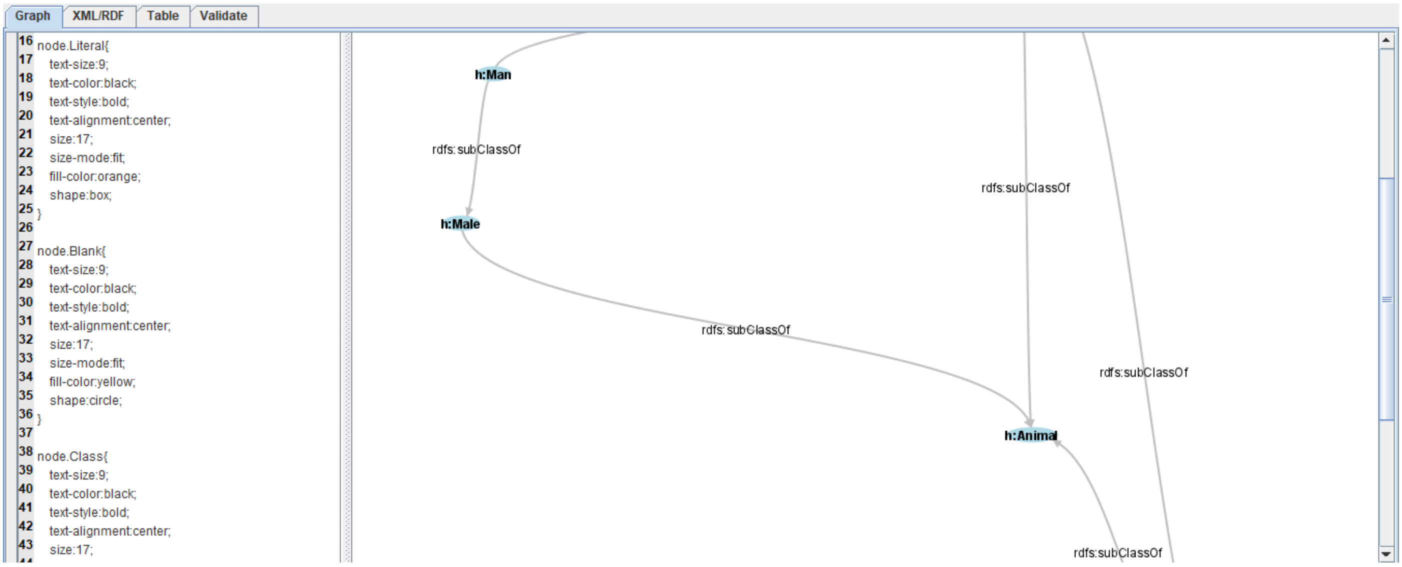
```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

   ?x rdfs:subClassOf h:Person

}
```

7. Rebuild the hierarchy of Classes (CONSTRUCT) considering only the classes in the humans.rdfs schema

query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

construct

{ ?x rdfs:subClassOf ?y .}

where

{ ?x rdfs:subClassOf ?y .}
```

screenshot:



8. To the previous CONSTRUCT add the signatures of the relations.

query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>
```

construct

{ ?x rdfs:subClassOf ?y.

?z1 rdfs:domain ?x.

?z2 rdfs:range ?x. }

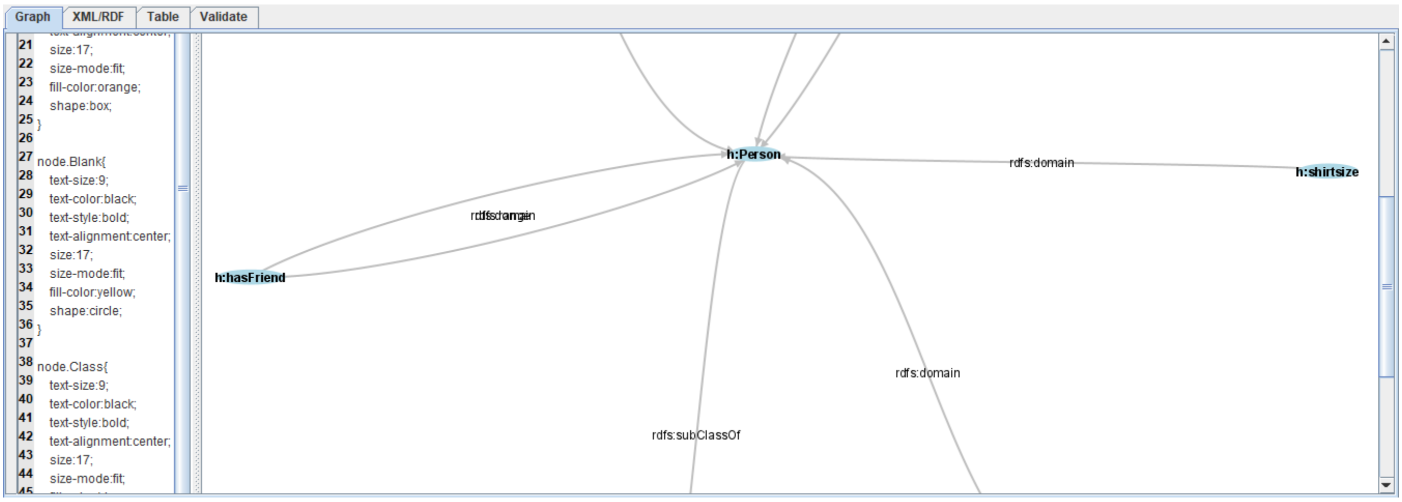
where

{ ?x rdfs:subClassOf ?y.

?z1 rdfs:domain ?x.

?z2 rdfs:range ?x. }

screenshot:



You now know how to query schemas on the semantic Web!

Query data augmented by an RDFS schema

Question 1

1. Reset the Corese engine and load only the annotations (.rdf)
2. Write a query to find the Persons.

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

   ?x ?p h:Person

}
```

Number of results before:

7

3. Load the schema (.rdfs)
4. Rerun the query to find the Persons and explain the result.

New number of results after and your explanation:

28

Question 2

1. Write a query to find Males and their wives. How many answers do you get? Explain this result.

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

  ?x a h:Man; h:hasSpouse ?wife

}
```

Number of results and explanation:

1

2. In the data declare that Lucas has to father Karl. Reset Corese, reload the ontology and the data, and then rerun the query to find Males and their wives. Explain the new result.

Line added in RDF:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

  ?x a h:Male; h:hasSpouse ?wife

}
```

Number of results before and after and explanation:

2

Question 3

1. Write a query to find the Lecturers and their types. How many answers do you get? See how this typing is declared in the data and explain the result.

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>
```

select * where {

?x a h:Lecturer

}

Number of results and your explanation:

2

2. Write a query to find common instances of the classes Person and Male. See how this typing is declared in the data and explain the presence of Jack.

Query:

prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

?x a h:Male, h:Person

}

Your explanation of the result:

Jack is declared a Man. With schema, we know that Man is subclass of Person and Male. That's why Jack is shown in the result query.

Question 4

Write a query to find the hasAncestor relations. Explain the result after checking where this property is used in the data.

Query:

prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

?x h:hasAncestor ?y

}

Your explanation of the result:

In the schema, we know that hasParent is a sub property of hasAncestor. That is why we get the following query.

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>

Question 5

1. Write a query to find the family cores (couples and their children) using a SELECT

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

  ?x h:hasSpouse ?y; h:hasChild ?z

}
```

2. Modify it to display the result with a CONSTRUCT query

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

construct { ?x h:hasSpouse ?y; h:hasChild ?z }

where { ?x h:hasSpouse ?y; h:hasChild ?z }
```

Question 6

1. Declare the olderThan relationship in the schema to indicate between two people which is eldest and construct the arcs between peoples with a SPARQL query

Addition to schema:

```
<http://www.inria.fr/2007/09/11/humans.rdfs#hasAncestor> a rdf:Property ;

    rdfs:label "has for ancestor"@en,

        "a pour ancêtre"@fr ;

    rdfs:comment "relation between an animal and another animal from which it is descended."@en,

        "relation entre un animal et un autre animal duquel il descend."@fr ;

    rdfs:domain <http://www.inria.fr/2007/09/11/humans.rdfs#Animal> ;

    rdfs:range <http://www.inria.fr/2007/09/11/humans.rdfs#Animal> ;

    rdfs:subPropertyOf <http://www.inria.fr/2007/09/11/humans.rdfs#olderThan>    .
```

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

construct { ?x h:olderThan ?y. }
```

where { ?x h:olderThan ?y. }

2. Find a query that generates only the minimum number of links without redundancy with olderThan transitivity.

Query:

<ANSWER HERE/>

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

?x h:olderThan ?y; h:age ?age.

}
```

Question 7

Write a query to find for John the properties which label contains the string "size" and the value of these properties.

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

   <http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>    ?v ?y.

   ?v rdfs:label "size"@en

}
```

Question 8

Use the ontology to document your answers in natural language: write a query to find the types and properties of Laura in French.

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

   {

   <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura> ?p ?y.

   ?y rdfs:label ?z.

   filter ( lang(?z) = "fr").

   } union {
```

```
    <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura> ?p ?y.

    ?p rdfs:label ?z.

    filter ( lang(?z) = "fr").

    }

}
```

Day 04: Answers to the practical session on OWL.

Software requirements

- The RDF XML online validation service by W3C: https://www.w3.org/RDF/Validator/
- The RDF online translator: http://rdf-translator.appspot.com/
- The SPARQL Corese engine: https://project.inria.fr/corese/

A, Query data augmented by an OWL schema

Make a copy of the human.rdfs file, name it humans.owl and use it for the rest of the session. For each of the following statements, specify a SPARQL query that shows that the difference before and after running the OWL inferences: you will find that answers to these queries are different depending on whether you load the ontology humans.rdfs or the humans.owl you modified.

1. Declare that `hasSpouse` is a symmetrical property and do the same for and `hasFriend` .

Code added to the schema:

<http://www.inria.fr/2007/09/11/humans.rdfs#hasSpouse> a rdf:Property, owl:SymmetricProperty ;

<http://www.inria.fr/2007/09/11/humans.rdfs#hasFriend> a rdf:Property, owl:SymmetricProperty ;

Query:

prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

?x ?p ?f.

filter (?p = h:hasSpouse || ?p = h:hasFriend)

}

Result before addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?p	?f
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Alice>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
10	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
11	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#William>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...
12	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs#h...	<http://www.inria.fr/2007/09/11/humans.rdfs-in...

Result after addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?p	?f
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
9	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
10	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
11	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
12	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasFr...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
13	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
14	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
15	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
16	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
17	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
18	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
19	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
20	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
21	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
22	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
23	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
24	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasS...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...

Explanation:

We get extra 12 results after adding owl:SymmetricProperty in the schema. Originally, we only has one direction of relationship, but now getting both side. For example, before we know Harry has spouse Sophie. And now we also know Sophie has spouse Harry.

2. Declare that `hasChild` is the inverse property of the `hasParent` property.

Code added to the schema:

<http://www.inria.fr/2007/09/11/humans.rdfs#hasChild> a rdf:Property ;

owl:inverseOf < http://www.inria.fr/2007/09/11/humans.rdfs#hasParent> ;

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

  ?x ?p ?f.

  filter ( ?p = h:hasChild || ?p = h:hasParent )

}
```

Result before addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?p	?f
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
9	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...
10	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...

Result after addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?p	?f
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
9	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
10	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasCh...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
11	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
12	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
13	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
14	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
15	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
16	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
17	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
18	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
19	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...
20	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs#hasPa...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...

Explanation:

We get extra 10 results after adding owl:inverseOf <#hasParent> in the schema. Originally, we only has one direction of relationship, but now getting both side. For example, before we know Jack has child Harry. And now we also know Harry has parent Jack.

3. Declare `hasAncestor` as transitive property.

Code added to the schema:

<http://www.inria.fr/2007/09/11/humans.rdfs#hasAncestor> a rdf:Property , owl:TransitiveProperty ;

Query:

prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

?x h:hasAncestor ?f.

}

Result before addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?f
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...	<http://www.inria.fr/2007/09/11/humans.rdfs-instance...

Result after addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
10	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>
11	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
12	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
13	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
14	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
15	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
16	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
17	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
18	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>

Explanation:

owl:TransitiveProperty give the result of connection. For example, John has ancestor Harry and Harry has ancestor Gaston. This code makes link between John and Gaston, which means John has ancestor Gaston.

4. Declare the disjunction between `Male` and `Female`. Violate the constraint in the data, check the results and then remove the violation you created.

Code added to the schema:

<http://www.inria.fr/2007/09/11/humans.rdfs#Female> owl:disjointWith <http://www.inria.fr/2007/09/11/humans.rdfs#Male> ;

<http://www.inria.fr/2007/09/11/humans.rdfs#Male> owl:disjointWith <http://www.inria.fr/2007/09/11/humans.rdfs#Female> ;

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

5. Declare that the class `Professor` is the intersection of the class `Lecturer` and `Researcher` class.

Code added to the schema:

<http://www.inria.fr/2007/09/11/humans.rdfs#Professor> a rdfs:Class ;

owl:intersectionOf (<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer> <http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>) .

Query:

prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#> select * where { ?x a h:Lecturer; a ?p }
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#> select * where { ?x a h:Professor . }

Result before addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?p
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Animal>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Female>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Person>
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>

Result after addition to the schema:

num	?x
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Explanation:

With adding owl:intersectionOf, we can select the result by asking who is both lecturer and researcher, but not manually checking.

6. Declare that the `Academic` class is the union of classes `Lecturer` and `Researcher`.

Code added to the schema:

<http://www.inria.fr/2007/09/11/humans.rdfs#Academic> a rdfs:Class ;

owl:unionOf(<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer> <http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>) .

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>
select * where {    ?x a ?p . Filter ( ?p = h:Lecturer || ?p = h:Researcher) }
```

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>
select * where {    ?x a h:Academic . }
```

Result before addition to the schema:

Graph XML/RDF Table Validate

num	?x	?p
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>

Graph XML/RDF Table Validate

num	?x

Result after addition to the schema:

Graph XML/RDF Table Validate

num	?x	?p
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Lecturer>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>	<http://www.inria.fr/2007/09/11/humans.rdfs#Researcher>

Graph XML/RDF Table Validate

num	?x
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Eve>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#David>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>

Explanation:

With adding owl:unionOf, we can select the result by asking who is either lecturer or researcher, but not manually checking.

7. Create a class `Organization` and its sub class `University`. Create a new property `mainEmployer`, with domain `Person` and range `Organization`. Use a restriction to declare that any `Professor` has for main employer a `University`.

Code added to the schema (new property, new classes and new restriction):

```
<http://www.inria.fr/2007/09/11/humans.rdfs#Professor> a rdfs:Class ;
    rdfs:subClassOf [a owl:Restriction ;
        owl:onProperty <http://www.inria.fr/2007/09/11/humans.rdfs#mainEmployer> ;
        owl:allValuesFrom <http://www.inria.fr/2007/09/11/humans.rdfs#University> ] .
```

```
<http://www.inria.fr/2007/09/11/humans.rdfs#Organization> a rdfs:Class ;
    rdfs:subClassOf <http://www.inria.fr/2007/09/11/humans.rdfs#University> .
```

```
<http://www.inria.fr/2007/09/11/humans.rdfs#mainEmployer> a rdfs:Property ;
    rdfs:domain <http://www.inria.fr/2007/09/11/humans.rdfs#Person> ;
    rdfs:range <http://www.inria.fr/2007/09/11/humans.rdfs#Organization> .
```

Code added to the data (just declare the main employer of a Professor):

```
<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer> a :Woman ;

    :hasSpouse <http://www.inria.fr/2007/09/11/humans.rdfs-instances#John> ;

    :mainEmployer <http://www.inria.fr/2007/09/11/humans.rdfs-instances#Org> ;

    :name "Jennifer" .
```

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where { ?x h:mainEmployer ?y. Optional { ?y a ?z } }
```

Result before addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jennifer>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Org>

Result after addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?y	?z
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...	<http://www.inria.fr/2007/09/11/humans.rdfs#Univer...
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instan...	<http://www.inria.fr/2007/09/11/humans.rdfs-instanc...	<http://www.inria.fr/2007/09/11/humans.rdfs#Organi...

Explanation:

We can know more about “Org” is a type of university and organization.

8. Use a restriction to declare that any person must have a parent who is a woman. For this last statement, you need to run the rule engine after loading the ontology and data.

Code added to the schema:

```
<http://www.inria.fr/2007/09/11/humans.rdfs#Person> a rdfs:Class ;

    rdfs:label "human"@en,

        "human being"@en,

        "person"@en,
```

```
            "homme"@fr,

            "humain"@fr,

            "personne"@fr,

            "être humain"@fr ;

        rdfs:comment "a member of the human species"@en,

            "un membre de l'espèce humaine."@fr ;

        rdfs:subClassOf <http://www.inria.fr/2007/09/11/humans.rdfs#Animal> ;

        rdfs:equivalentClass [a owl:Restriction ;

           owl:onProperty <http://www.inria.fr/2007/09/11/humans.rdfs#hasParent> ;

           owl:someValuesFrom <http://www.inria.fr/2007/09/11/humans.rdfs#Woman> ;

           owl:minCardinality "1"^^xsd:int] .
```

Query:

```
prefix h:<http://www.inria.fr/2007/09/11/humans.rdfs#>

select * where {

  ?x h:hasParent ?y

}
```

Result before addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>

Result after addition to the schema:

Graph | XML/RDF | Table | Validate

num	?x	?y
1	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Flora>
2	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Pierre>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>
3	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Sophie>
4	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>
5	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Laura>
6	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Karl>
7	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Lucas>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Catherine>
8	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Mark>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#John>
9	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Harry>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>
10	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Jack>	<http://www.inria.fr/2007/09/11/humans.rdfs-instances#Gaston>

Explanation:

We can see there is a least a mother of each person.

B, Make your own OWL models:

For each one of the following OWL primitives imagine a definition that could use it and provide that definition in OWL using your preferred syntax (RDF/XML or N3/Turtle). For instance a possible definition using owl:TransitiveProperty would be a definition of the Ancestor property. For each primitive in the following list you imagine the definition of a class or property that was not given in the course and you give that definition in English and in OWL.

1	owl:oneOf	<BloodType> owl:oneOf (<A> <B> <O> <AB>) .	There are four kinds of blood types, A, B, O and AB.
2	owl:unionOf	<Drinks> owl:unionOf (<Tea> <Milk>) .	Drinks can be tea, milk or both.
3	owl:intersectionOf	<Milktea> owl: intersectionOf (<Tea> <Milk>) .	Milk tea is the combination of tea and milk.
4	owl:complementOf	<Healthy> owl: complementOf (<unhealthy>) .	One can be either healthy or healthy.
5	owl:disjointWith or owl:AllDisjointClasses or owl:disjointUnionOf	<Red> owl: disjointWith <Blue> .	Red and blue are independent.
6	owl:ObjectProperty	<hasFriend> owl:ObjectProperty (#A, #B) .	A and B are friends.
7	owl:DatatypeProperty	<hasShoesize> a owl:DatatypeProperty	Has shoe size is an integer data.
8	owl:SymmetricProperty or owl:AsymmetricProperty	<hasEmployee> a owl: AsymmetricProperty .	One can has another an employee, but no opposite way.
9	owl:inverseOf	< hasEmployee > owl:inverseOf < hasBoss > .	Has employee and has boss are relative relationship.
10	owl:TransitiveProperty	<isOlderthan> a owl:TransitiveProperty .	Is older than is a chain effect.
11	owl:propertyDisjointWith	<hasEmployee> owl: ObjectProperty (<hasBoss>) .	Has employee and has boss are opposite.
12	owl:ReflexiveProperty or owl:IrreflexiveProperty	<know> a owl:ReflexiveProperty .	One can know someone including him/herself.
13	owl:propertyChainAxiom	<parentsinlaw> a owl:propertyChainAxiom (<spouse> <parents>) .	Parent in law is parent of one's spouse.
14	owl:FunctionalProperty	<roomnumber> a owl: FunctionalProperty.	Room number can track to specific people.
15	owl:InverseFunctionalProperty	<PassportNumber> a owl: InverseFunctionalProperty.	By passport number can track to one and only one.
16	owl:hasKey	<Student> owl:hasKey (<name> <studentID> <grade> <class>) .	Student owns name, studentID, grade and class for identification.
17	owl:allValuesFrom	<student> owl:allValuesFrom <human> .	If one is a student, one must be human.
18	owl:someValuesFrom	<student> owl: someValuesFrom <clubmember> .	Part of students are clubmember.
19	owl:hasValue	<octopus> owl: hasValue <8^^xsd:int >	Octopus has 8 legs.
20	owl:maxCardinality or owl:minCardinality	<family> owl: minCardinality <2^^xsd:int >	There are minimal 2 people in a family.

21	owl:qualifiedCardinality	<spouse> owl: qualifiedCardinality <1^^xsd:int >	One can only get a spouse.