Examination – Semantic web: from XML to OWL

Duration: 3h
Any document allowed

Janvier 2009

Metanota: This document is the translation of only the Semantic web part of the 2009 exam and its correction.

Nota: Both parts are independent and each of them is ranked on 10 points. Return you answers in two different sheets (or sets of sheets).

Part 2: Semantic web

Answers are, in general, short.

RDF assertions

Consider the four graphs of Figure 1.

1. Are they all valid RDF graphs? Why?

   Graph (b) is not a valid RDF graph because a literal, `"La peste"` is the subject of a relation (`rdf:type`).

2. Express the graph of Figure 1(a) as a set of triples.

   ```
   ?x foaf:name "Albert".
   ?x m:aecrit ?l.
   ?l rdf:type m:Roman.
   ?l dc:title "La peste".
   ```

   You will list the sets of literals, URIRefs and variables (or blanks) found in this graph.

   Literals: "Albert", "La peste"
   URIRefs: `foaf:name`, `m:aecrit`, `m:Roman`, `rdf:type`, `dc:title`
   Variables: `?x`, `?l`

3. Give an informal meaning of this graph or its expression in the predicate calculus

   Some entity named `"Albert"` has written a Novel titled `"La peste"`.

   $$\exists x, \exists l, name(x, \text{“}Albert''') \wedge aecrit(x, l) \wedge Roman(l) \wedge title(l, \text{“}Lapeste'')$$

4. Consider the RDF graphs of Figure 1 which are well-formed, do some of them entail others?

   $$(a) \models (d) \text{ and } (d) \models (a)$$

   Explain why.

   Since (a) is a strict subgraph of (d), then it is obvious that (d) $\models$ (a).
   For the opposite direction, it is sufficient to see that the node labeled by `?y` in (d) can be projected to

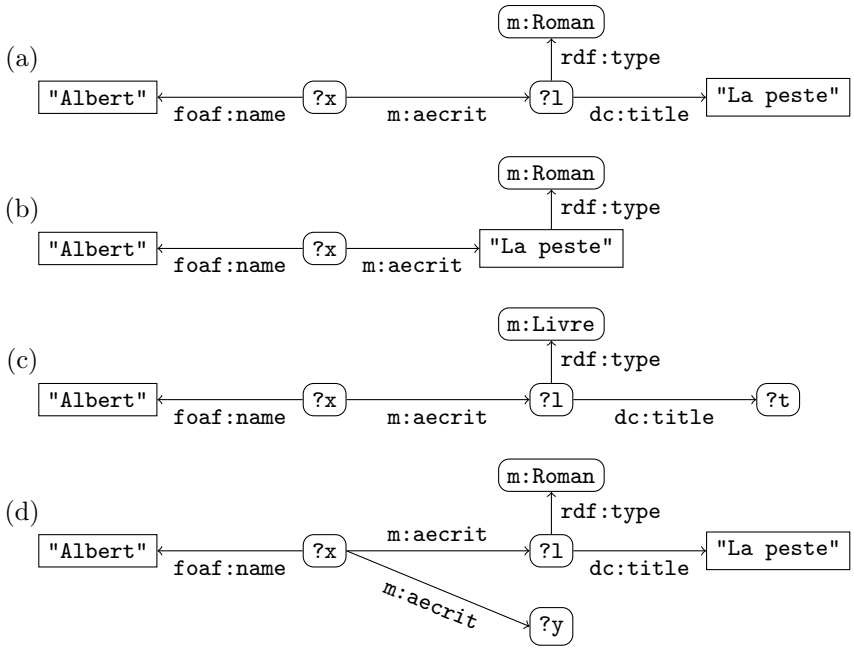

Figure 1: RDF graphs.

the node labeled `?l` *in* (a) *while projecting the other nodes to those nodes bearing the same label. This projection preserves all the edges of the graph and it is complete, then* $(a) \models (d)$.

In predicate calculus, this is:

$$\exists x, \exists l, name(x, \text{"Albert"}) \wedge aecrit(x, l) \wedge Roman(l) \wedge title(l, \text{"La peste"})$$

is equivalent to:

$$\exists x, \exists l, name(x, \text{"Albert"}) \wedge aecrit(x, l) \wedge Roman(l) \wedge title(l, \text{"La peste"}) \wedge \exists y, aecrit(x, y)$$

which is equivalent to:

$$\exists x, \exists l, \exists y, name(x, \text{"Albert"}) \wedge aecrit(x, l) \wedge Roman(l) \wedge title(l, \text{"La peste"}) \wedge aecrit(x, y)$$

OWL ontologies

5. Describe in OWL (RDF or XML/RDF) the ontology containing the following assertions:
   - All authors are persons;
   - A book (`m:Livre`) has exactly one year of publication (`m:annee`);
   - A novel (`m:Roman`) is a book (`m:Livre`) and a book is a work (`m:Oeuvre`);
   - The title (`dc:title`) of a work is a character string (`xsd:string`);
   - The relation "a écrit" (`m:aecrit`) relates an author to a work.

```
<owl:Class rdf:about="#Auteur">
  <owl:subClassOf rdf:resource="#Personne"/>
</owl:Class>
```

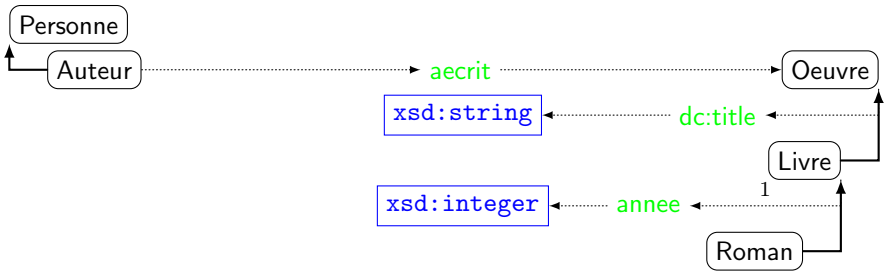

Figure 2: OWL ontologies.

```
<owl:Class rdf:about="#Oeuvre">
  <owl:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="dc:title"/>
      <owl:datatype>&xsd;string</owl:datatype>
    </owl:Restriction>
  </owl:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Livre">
  <owl:subClassOf rdf:resource="#Oeuvre"/>
  <owl:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#annee"/>
      <owl:cardinality>1</owl:cardinality>
    </owl:Restriction>
  </owl:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Roman">
  <owl:subClassOf rdf:resource="#Livre"/>
</owl:Class>

<owl:ObjectProperty rdf:about="#aecrit">
  <rdfs:domain rdf:resource="#Auteur"/>
  <rdfs:range rdf:resource="#Oeuvre"/>
</owl:ObjectProperty>
```

which corresponds to the diagram of Figure 2.

6. If one associates the graph (a) of Figure 1 and the ontology resulting from the previous question, is it possible to deduce the type (`rdf:type`) of `?x`?

   `?x rdf:type m:Auteur`.

   Can you semantically justify how?

   $I(\texttt{aecrit}) \subseteq I(\texttt{Auteur}) \times I(\texttt{Oeuvre})$
   $\langle I(\texttt{?x}), I(\texttt{?l})\rangle \in I(\texttt{aecrit})$
   hence
   $I(\texttt{?x}) \in I(\texttt{Auteur})$
   thus: `?x rdf:type Auteur`

   What else can be deduced?

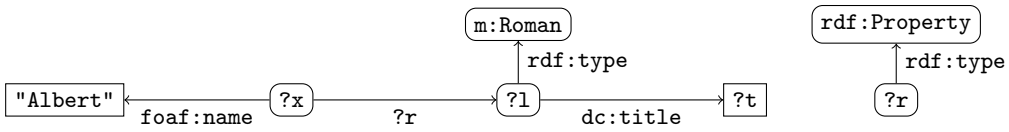

Figure 3: SPARQL graph patterns.

`?l` *has a year of publication:* `?l m:annee ?y`.

7. Does the use of this ontology for defining the graphs of Figure 1 would change something to the answer to Question 4? What?

   Yes: $(a) \models (c)$ because `?l rdf:type m:Roman` $\models$ `?l rdf:type m:Livre`

   and $(d) \models (c)$ *because* $(d) \models (a)$.

SPARQL queries

Given the following SPARQL query:

```
SELECT ?t
PREFIX
  foaf: http://xmlns.com/foaf/0.1/
  m: http://mydomain.com/myExample#
  dc: http://purl.org/dc/elements/1.1/
  rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
WHERE
  ?x foaf:name "Albert".
  ?x ?r ?l.
  ?r rdf:type rdf:Property.
  ?l rdf:type m:Roman.
  ?l dc:title ?t.
```

8. What is the informal meaning of this query?

   Find all titles (`dc:title`) of novels (`m:Roman`) related (`?r`) to an entity (`?x`) whose name (`foaf:name`) is "Albert".

9. Draw the RDF graph corresponding to the graph patterns of the query.

   It is given in Figure 3.

10. What is the difference between such graph patterns and simple RDF graphs?

    These graph patterns are generalised RDF graphs*: they can contain blanks or variables as edge labels (or as properties in RDF triple terms).*

11. Evaluate this query on each of the well-founded graphs of Figure 1 and provide the answer:

    (a) : {⟨`"La peste"`⟩}
    (c) : {} (`?l` *is a book* (`Livre`) *but not a novel* (`Roman`))
    (d) : {⟨`"La peste"`⟩}

12. What must be added to this query for returning the year of publication of the `m:Roman` if it is available?

    An optional clause:

    ```
    SELECT ?t ?y
    ...
    OPTIONAL { ?l m:annee ?y. }
    ```

Semantic peer-to-peer system[1]

Given a peer-to-peer system with peers n_1, n_2 and n_3 all related to each others. Peers n_1 and n_2 share the ontology o that has been defined at Question 5, n_3 uses the ontology o' defining;

```
t:Work rdf:type owl:Class.
t:copyrightHolder rdf:type rdf:Property.
t:copyrightHolder rdfs:domain t:Work.
t:year rdf:type rdf:Property.
t:year rdfs:domain t:Work.
t:year rdfs:range xsd:integer.
```

There exists an alignment A between o and o' containing two correspondences: `t:Work` $\geq$ `m:Livre`, `t:year` $\equiv$ `m:annee`. Assume that n_1 does not contain instances; n_2 contains:

```
http://mm.com#a345 m:aecrit http://isbn.org/2070360423.
http://mm.com#a345 m:aecrit http://isbn.org/2070360024.
http://mm.com#a345 m:aecrit http://isbn.org/2070322882.
http://mm.com#a345 foaf:name "Albert".
http://isbn.org/2070360423 rdf:type m:Roman.
http://isbn.org/2070360423 dc:title "La peste".
http://isbn.org/2070360024 rdf:type m:Roman.
http://isbn.org/2070360024 dc:title "L'étranger".
http://isbn.org/2070322882 rdf:type m:Livre.
http://isbn.org/2070322882 dc:title "Le Mythe de Sisyphe".
...
```

and n_3 contains:

```
http://isbn.org/2070360423 t:year 1947.
http://isbn.org/2070322882 t:year 1942.
...
```

13. Express the alignment A in OWL.

```
<owl:Class rdf:about="m:Livre">
  <owl:subClassOf rdf:resource="t:Work"/>
</owl:Class>

<owl:DataProperty rdf:about="t:year">
  <owl:equivalentProperty rdf:resource="m:annee"/>
</owl:DataProperty>
```

14. The peer n_1 would like to evaluate the following query:

```
SELECT ?t, ?y
PREFIX ...
WHERE
  ?x foaf:name "Albert".
  ?x m:aecrit ?l.
  ?l rdf:type m:Roman.
  ?l dc:title ?t.
  OPTIONAL { ?l m:annee ?y. }
```

How is it possible to answer this query by using n_2, n_3 and A?

It can be evaluated directly in n_1 and n_2. It can be translated with the help of A for being evaluated in n_3. However, the three data base do not contain the same type of information: n_1 contains no data,

[1] This exercise has not been given at the actual exam.

n_2 contains information about authors and titles while n_3 contains information about publication year. Hence, instead of transforming the query for evaluating it on n_3, it is better to split it into two queries: The result of the transformation would be:

```
SELECT ?l, ?t
PREFIX ...
WHERE
  ?x foaf:name "Albert".
  ?x m:aecrit ?l.
  ?l rdf:type m:Roman.
  ?l dc:title ?t.
```

and

```
SELECT ?l, ?y
PREFIX ...
WHERE
OPTIONAL { ?l m:annee ?y }
```

The second query can be transformed thanks to A into:

```
SELECT ?l, ?y
PREFIX ...
WHERE
OPTIONAL { ?l t:year ?y }
```

and their results can be joined. This is not particularly efficient but this should work. It is also possible to first evaluate the former query to retrieve `?l` *and then to issue queries for specific values of the first answer.*

15. Provide the answer on the available data.

    Evaluating directly the query on n_2 would yield:

    $$\{\langle \texttt{"La peste"}, \texttt{null}\rangle,$$
    $$\langle \texttt{"L'étranger"}, \texttt{null}\rangle, \dots\}$$

    while evaluating a transformation of this query against n_3 would yield an empty answer set.

    Using the stragegy provided above would yield the results:

    $$\{\langle \texttt{"La peste"}, 1947\rangle,$$
    $$\langle \texttt{"L'étranger"}, \texttt{null}\rangle, \dots\}$$