# M2R Exam – Semantic web: from XML to OWL
# Semantic web part

Duration : 1h30
Documents allowed – no communication device allowed

January 2015

**Note:** Read all the questions carefully before answering.

## Semantic web part

### RDF Entailment

Consider the graph $G$ of Figure 1.

1. Explain what it means (in English or French, taking into account the meaning of RDFS vocabulary).

   CHUs are hospitals. A gyneco-obsetrics service is a maternity. The Belledonne hospital has a pediatry and a gyneco-obsetrics service. The La tronche CHU has a maternity and a radiology service.

2. Rewrite it as a set of triples.

```
a:CHU rdfs:subClassOf a:Hospital .
a:Gyneco-Obsetrics rdfs:subClassOf a:Maternity .
b:Belledonne rdf:type a:Hospital .
b:H.Michallon rdf:type a:CHU .
b:Bellodone rdfs:label "Belledonne" .
b:H.Michallon rdfs:label "La Tronche" .
b:Belledone a:hasService b:GO .
b:GO rdf:type a:Gyneco-Obsetrics .
b:Belldonne a:hasService b:Pediatry .
b:Pediatry rdf:type a:Pediatry .
b:H.Michallon a:hasService b:Radio .
b:Radio rdf:type a:Radiology .
b:H.Michallon a:hasService b:Maternity .
b:Maternity rdf:type a:Maternity .
```

Given the GRDF graph $P$={ ?x `rdf:type a:CHU` . ?x `a:hasService` ?y . ?y `rdf:type a:Maternity` . }

3. Does $G \models_{RDF} P$ ? (explain how and/or why)

   Yes. Because the instantiation of $P$ with ?x replaced by `b:H.Michallon` and ?y replaced by `b:Maternity` is a subgraph of $G$.

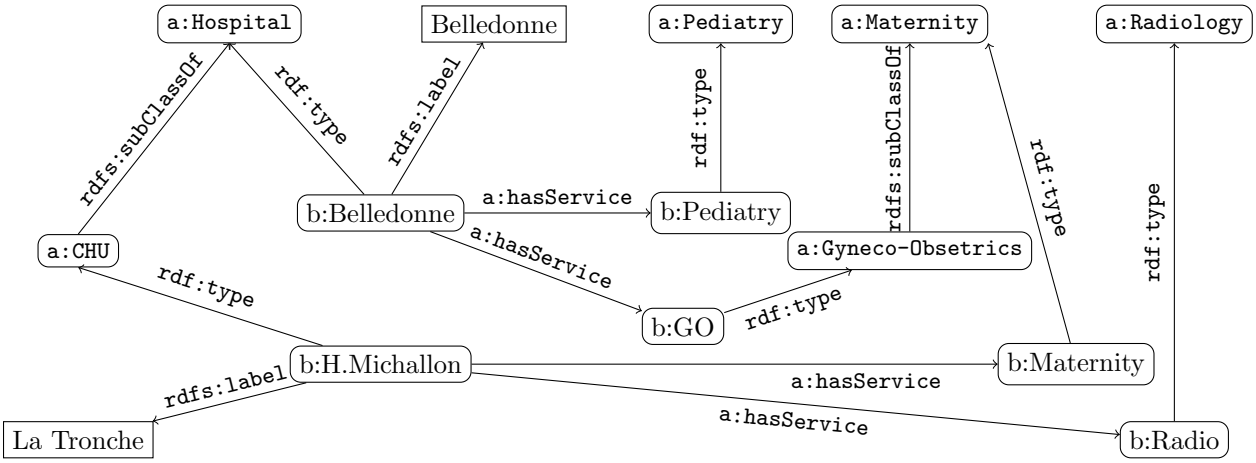


Figure 1: RDF graph $G$.

## RDFS Entailment

Given $P'$={ ?x `rdf:type a:Hospital` . ?x `a:hasService` ?y . ?y `rdf:type a:Maternity` . ?x `a:hasService` ?z . ?z `rdf:type a:Pediatry` .},

4. Does $G \models_{RDF} P'$? (explain how and/or why)

   No. Because there is only one `a:Hospital` in $G$, but there is no triple in $G$ stating that it has a service whose `rdf:type` is `a:Maternity`.

5. Which procedure may be used to derive that $G \models_{RDFS} P'$?

   The relevant procedure would be to compute the partial ter Horst closure $\hat{G}\backslash P'$ and show that $\hat{G}\backslash P' \models P'$.

## SPARQL CONSTRUCT

It is possible to generate RDF graphs from an RDF graph through the use of SPARQL CONSTRUCT. Consider the query $Q$:

CONSTRUCT { ?y `rdf:type` ?x }
WHERE { ?u `rdfs:subClassOf` ?x . ?y `rdf:type` ?u }.

6. Apply $Q$ to the graph $G$ of Figure 1 and give the resulting graph. Let us call the result $Q(G)$, does $G \cup Q(G) \models_{RDF} P'$?

   ```
   b:H.Michallon rdf:type a:Hospital .
   b:GO rdf:type a:Maternity .
   ```

   $G \cup Q(G) \models_{RDF} P'$ because by substituting in $P'$ ?x by `b:Belledonne`, ?y by `b:GO` and ?z by `b:Pediatry`, we obtain a subgraph of $G \cup Q(G)$.

7. For any map $\sigma$, does

   $$\sigma(?u) \texttt{ rdfs:subClassOf } \sigma(?x).\ \sigma(?y) \texttt{ rdf:type } \sigma(?u) \models_{RDFS} \sigma(?y) \texttt{ rdf:type } \sigma(?x)$$

   (explain why)

Yes because for any RDFS model $\langle I_R, I_P, Class, I_{EXT}, I_{CEXT}, Lit, \iota\rangle$ of the premises, $\iota(\sigma(?y)) \in I_{CEXT}(\iota(\sigma(?u)))$ and $I_{CEXT}(\iota(\sigma(?u))) \subseteq I_{CEXT}(\iota(\sigma(?x)))$, hence $\iota(\sigma(?y)) \in I_{CEXT}(\iota(\sigma(?x)))$, which means that $\sigma(?y)$ `rdf:type` $\sigma(?x)$ is satisfied by any model of the premises.

8. Is this related with the rules of the ter Horst closure? (tell if it corresponds more closely to one of these rules).

   In fact this query is a transcription of [RDFS11].

9. Would it be possible to transform all closure rules as SPARQL CONSTRUCTS? Do it or explain why.

   Rules [RDF1] and [RDFS1] could either be rendered by a CONSTRUCT with empty WHERE part or simply added to the graph $G$. Rule [RDF3] is a bit difficult to transcribe the WHERE clause. Otherwise all other rules can be expressed as SPARQL CONSTRUCT as follows:

| | | | |
|---|---|---|---|
| [Q2] | CONSTRUCT | { ?p `type` `prop` } | WHERE { ?s ?p ?o } |
| [Q6] | CONSTRUCT | { ?u `type` ?x } | WHERE { ?u ?a ?v . ?a `dom` ?x } |
| [Q7] | CONSTRUCT | { ?v `type` ?x } | WHERE { ?u ?a ?v . ?a `range` ?x } |
| [Q8a] | CONSTRUCT | { ?x `sp` ?x } | WHERE { ?x `type` `prop` } |
| [Q8b] | CONSTRUCT | { ?x `sp` ?z } | WHERE { ?x `sp` ?y . ?y `sp` ?z } |
| [Q9] | CONSTRUCT | { ?x ?b ?y } | WHERE { ?x ?a ?y . ?a `sp` ?b } |
| [Q10] | CONSTRUCT | { ?x `sc` `res` } | WHERE { ?x `type` `class` } |
| [Q11] | CONSTRUCT | { ?y `type` ?x } | WHERE { ?u `sc` ?x . ?y `type` ?u } |
| [Q12a] | CONSTRUCT | { ?x `sc` ?x } | WHERE { ?x `type` `class` } |
| [Q12b] | CONSTRUCT | { ?x `sc` ?z } | WHERE { ?x `sc` ?y . ?y `sc` ?z } |
| [Q13] | CONSTRUCT | { ?x `sp` `member` } | WHERE { ?x `type` `contMP` } |
| [Q14] | CONSTRUCT | { ?x `sc` `literal` } | WHERE { ?x `type` `datatype` } |

10. What more is needed for computing the closure?

    SPARQL queries are one shot, for computing the closure it is necessary that the result of rules be applied recursively until they do not generate new triples.

    (Q2,Q6,Q7,Q8a,Q8b,Q9,Q10,Q11,Q12a,Q12b,Q13,Q14)$^*(RDF1 \cup RDFS1 \cup G)$