M2R Exam – Semantic web: from XML to OWL

Duration : 3h
Any document allowed

January 2010

Note : Both parts are independent and will be rated 1/3-2/3. Answers corresponding to each part should be returned on separate sheets. Please, carefully read the questions before answering.

Part 2: Semantic web

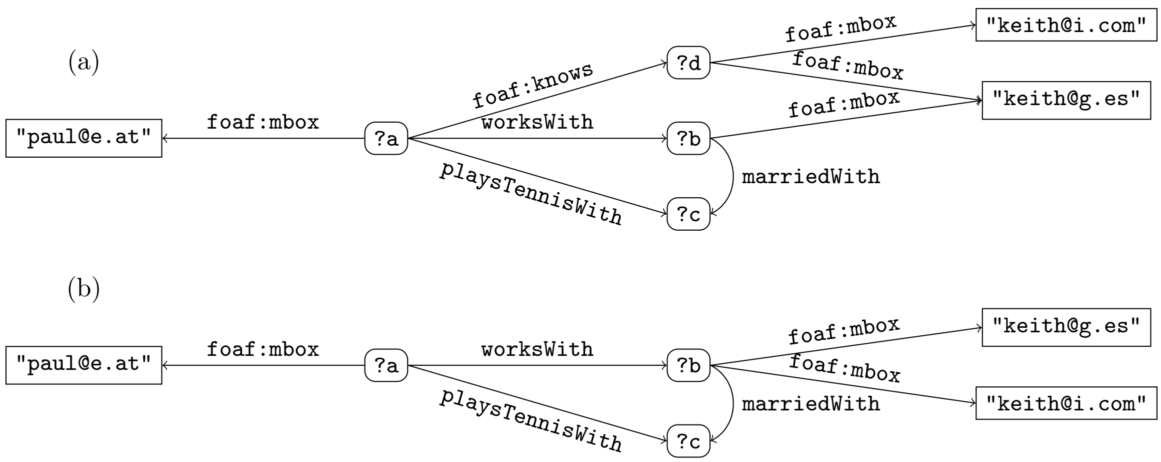


Figure 1: RDF graphs.

RDF and assertions

Consider the two graphs of Figure 1 dealing with the expression of social relationships.

1. Express the graph of Figure 1(b) as a set of triples. You will give the list of litterals, URIRefs and variables (or blanks) in this graph.

⟨?a, foaf:mbox, "paul@e.at"⟩⟨?a, worksWith, ?b⟩
⟨?a, playsTennisWith, ?c⟩⟨?b, marriedWith, ?c⟩
⟨?b, foaf:mbox, "keith@g.es"⟩⟨?b, foaf:mbox, "keith@i.com"⟩

$$\mathcal{L} = \{\texttt{"paul@e.at"}, \texttt{"keith@i.com"}, \texttt{"keith@g.es"}\}$$
$$\mathcal{U} = \{\texttt{worksWith}, \texttt{playsTennisWith}, \texttt{foaf:mbox}, \texttt{marriedWith}\}$$
$$\mathcal{B} = \{\texttt{?a}, \texttt{?b}, \texttt{?c}\}$$

2. What is, informally, the meaning of the graph of Figure 1(b) (tell it in English or French or predicate calculus)?

   An individual whose email address is "paul@e.at" plays tennis with the spouse of a colleague whose email addresses are "keith@g.es" and "keith@i.com". Or, in predicate calculus:

$$\exists a, \exists b, \exists c; worksWith(a,b) \wedge playsTennisWith(a,c) \wedge marriedWith(b,c)$$
$$\wedge\, mbox(a, "paul@e.at") \wedge mbox(b, "keith@g.es") \wedge mbox(b, "keith@i.com")$$

3. Does one of these graphs entail the other? (explain why)

   $a \not\models b$ because it is not possible to deduce that ∃ x; x marriedWith c and x foaf:mbox keith@i.com

   $b \not\models a$ because it is not possible to deduce worksWith from `foaf:knows`.

   Another interesting answer has been given by a student. The following SPARQL query:

```
ASK ?x worksWith ?y. ?y foaf:mbox "keith@i.com".
```

   Succeed for (b) and fails for (a), while

```
ASK ?x foaf:knows ?y.
```

   succeed for (a) and fails for (b). Hence, none of these graphs entail the other.

4. Provide an ontology satisfied by both graphs of Figure 1.

```
<rdfs:ObjectProperty rdf:about="&foaf;knows">
  <rdfs:domain rdf:resource="&foaf;Person" />
  <rdfs:range rdf:resource="&foaf;Person" />
</rdfs:ObjectProperty>

<rdfs:ObjectProperty rdf:about="&foaf;mbox">
  <rdfs:domain rdf:resource="&foaf;Person" />
  <rdfs:range rdf:resource="&xsd;string" />
</rdfs:ObjectProperty>

<owl:ObjectProperty rdf:about="worksWith">
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
  <rdfs:subPropertyOf rdf:resource="&foaf;knows" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="playsTennisWith">
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
  <rdfs:subPropertyOf rdf:resource="#playsSportWith" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="playsSportWith">
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
  <rdfs:subPropertyOf rdf:resource="&foaf;knows" />
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="marriedWith">
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty" />
  <rdfs:subPropertyOf rdf:resource="&foaf;knows" />
</owl:ObjectProperty>
```

DL-Lite ontologies

Consider the ontology O made of the following DL-Lite assertions:

$$\begin{aligned}
\texttt{worksWith} &\sqsubseteq \texttt{foaf:knows}\\
\texttt{playsTennisWith} &\sqsubseteq \texttt{playsSportWith}\\
\texttt{playsSportWith} &\sqsubseteq \texttt{foaf:knows}\\
\texttt{marriedWith} &\sqsubseteq \texttt{foaf:knows}\\
\texttt{Person} &\sqsubseteq \exists\texttt{foaf:mbox}
\end{aligned}$$

5. In which dialect (sublanguage) of DL-Lite is this ontology expressed (DL-Lite$_{core}$, DL-Lite$_{\mathcal{F}}$, DL-Lite$_{\mathcal{R}}$)?

   This is DL-Lite$_{\mathcal{R}}$ because there are assertions on relations.

6. Rewrite O in OWL or RDFS.

```
<owl:ObjectProperty rdf:about="worksWith">
  <rdfs:subPropertyOf rdf:resource="&foaf;knows" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="playsTennisWith">
  <rdfs:subPropertyOf rdf:resource="#playsSportWith" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="playsSportWith">
  <rdfs:subPropertyOf rdf:resource="&foaf;knows" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="marriedWith">
  <rdfs:subPropertyOf rdf:resource="&foaf;knows" />
</owl:ObjectProperty>

<owl:Class rdf:about="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&foaf;mbox" />
      <owl:someValuesFrom rdf:resource="&owl;Thing" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

   or, for the last one:

```
<owl:Class rdf:about="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&foaf;mbox" />
      <owl:minCardinality>1</owl:minCardinality>
```

```
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

7. In which dialect (sublanguage) of RDFS or OWL is your ontology expressed?

   This is OWL-Lite. Although, DL-Lite$_{\mathcal{R}}$ is more expressive than RDFS and non comparable with OWL-Lite, this particular ontology is expressed in OWL-Lite (in which `minCardinality` constraints are expressible with values of 0 or 1).

8. Given the graph of Figure 1(b) to which the axioms of O are added. Compute its closure. What is the difference with the partial closure?

   The closure is:

$$\langle \texttt{?a}, \texttt{foaf:mbox}, \texttt{"paul@e.at"} \rangle \langle \texttt{?a}, \texttt{worksWith}, \texttt{?b} \rangle$$
$$\langle \texttt{?a}, \texttt{playsTennisWith}, \texttt{?c} \rangle \langle \texttt{?b}, \texttt{marriedWith}, \texttt{?c} \rangle$$
$$\langle \texttt{?b}, \texttt{foaf:mbox}, \texttt{"keith@g.es"} \rangle \langle \texttt{?b}, \texttt{foaf:mbox}, \texttt{"keith@i.com"} \rangle$$
$$\langle \texttt{?a}, \texttt{foaf:knows}, \texttt{?c} \rangle \langle \texttt{?a}, \texttt{foaf:knows}, \texttt{?b} \rangle$$
$$\langle \texttt{?a}, \texttt{playsSportWith}, \texttt{?c} \rangle \langle \texttt{?b}, \texttt{foaf:knows}, \texttt{?c} \rangle$$

   plus the axiomatic triples (like $\langle$`foaf:mbox`, `rdf:type`, `rdf:Property`$\rangle$, $\langle$`rdf:type`, `rdf:type`, `rdf:Property`$\rangle$, $\langle$`rdf:_1`, `rdf:type`, `rdf:Property`$\rangle$...). The partial closure is the same minus the triples involving `rdf:_`i because there is no such URI in our graph.

9. Given the graphs of Figure 1 to which the axioms of O are added, does this change something to the answers given to Question 3 if we consider RDFS-entailment? (explain why)

   Now $b, O \models a$ because worksWith $\sqsubseteq$ foaf:knows.

OWL ontologies

We would like to express that an email address cannot correspond to more than one person by the property `foaf:mbox`.

10. How to express this in OWL: with a cardinality constraint, a functional property, an inverse functional property or a symmetric property?

    This is an inverse functional property which states that to an email address corresponds only one image by the inverse of `foaf:mbox`. This is simply expressed by[1]:

```
<owl:InverseFunctionalProperty rdf:about="&foaf;mbox" />
```

    It is possible to use:

```
<owl:Class rdf:about="&owl;Thing">
  <owl:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty>
          <owl:inverseOf rdf:resource="&foaf;mbox" />
        </owl:ObjectProperty>
      </owl:onProperty>
      <owl:maxCardinality>1</owl:maxCardinality>
    </owl:Restriction>
  </owl:subClassOf>
</owl:Class>
```

[1] In principle, in OWL 1, this only works on ObjectProperties and `foaf:mbox` is a DataProperty.

11. Is it possible to express this constraint in DL-Lite? If yes, how?

    This is possible in the DL-Lite$_\mathcal{F}$ language:

    `(functional foaf:mbox`$^{-1}$`)`

12. Consider the graphs of Figure 1 to which this constraint is added, does this change something to the answers given to Question 3? (explain why)

    (a) becomes a super-graph of (b), because `?b` and `?d` becomes the same node because they have the `foaf:mbox "keith@g.es"` in common and this an inverse functional property. The only difference between both graphs is that $\langle$`?a`, `foaf:knows`, `?b`$\rangle$ belongs to (a) and not to (b). As a consequence, $a, O' \models b$ (with O' the ontology with the inverse functional property constraint), but not the other way around for the same reason as before.

SPARQL and queries

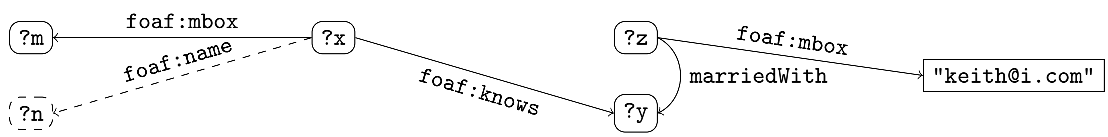
Given the following SPARQL query:

```
SELECT ?m, ?n
PREFIX  foaf: http://xmlns.com/foaf/0.1/
WHERE {
  ?x foaf:mbox ?m.
  ?x foaf:knows ?y.
  ?z foaf:mbox "keith@i.com".
  ?z marriedWith ?y.
  OPTIONAL { ?x foaf:name ?n }
}
```

13. What is, informally, the meaning of this query (tell it in English or French)?

    What are the email address, and if available the name, of people who know the spouse of the person with email address "keith@i.com"?

14. Draw the GRDF graph corresponding to the graph pattern of this query.



15. Evaluate the query on the graphs of Figure 1 and provide the results.

    For both graphs, there is no result because there is no node which is both the object of a `foaf:knows` and a `marriedWith` (for matching `?y`).

16. Evaluate this query on the closure obtained at Question 8 and provide the results.

    (a) still no result because this time there is no node whose `foaf:mbox` is `"keith@i.com"` which is the subject of a `marriedWith` relation, (b) 1 result: $\{\langle$`?m`, `"paul@e.at"`$\rangle$, $\langle$`?n`, `null`$\rangle\}$ because, $\langle$`?a`, `foaf:knows`, `?c`$\rangle$ belongs to the closure of (b).

17. This is not the standard way for answering queries modulo DL-Lite ontologies. What is the proposed method? Rewrite the above query with this method and give its results.

    The usual way to evaluate queries modulo DL-Lite is to rewrite the query as the following:

```
SELECT ?m, ?n
PREFIX  foaf: http://xmlns.com/foaf/0.1/
WHERE {
  ?x foaf:mbox ?m.
  { { ?x foaf:knows ?y. }
    UNION { ?x foaf:playsSportWith ?y. }
    UNION  { ?x foaf:playsTennisWith ?y. }
    UNION { ?x foaf:worksWith ?y. }
    UNION  { ?x marriedWith ?y.}
  }
  ?z foaf:mbox "keith@i.com".
  ?z marriedWith ?y.
  OPTIONAL { ?x foaf:name ?n }
}
```

It can be checked that this query will return the same results as in Question 16.