

CO7216 Examination — Solutions

This copy generated 15th March 2018.

Title of paper	CO7216 — Semantic Web
Version	1
Candidates	All candidates
Department	Informatics
Examination Session	MIDSUMMER EXAMINATIONS 2017
Time allowed	Two Hours
Instructions	Please do not remove this paper from the examination hall. This paper contains TWO questions. All questions carry equal weight (50 marks per question). Full marks may be obtained for answers to two questions. All relevant namespaces have been provided in the helper sheet.
Calculators	No
Books/statutes	No
Own Books/statutes/notes	No
Additional Stationery	No
Number of questions	2

Dwelling Scenario

The Dwelling Scenario described below is used in parts of both Questions 1 and 2.

A dwelling is a self-contained unit of accommodation that functions as a home. Houses and flats are the most common types of dwelling in the UK.

In general, houses can be classified into these categories: bungalows, detached, semi-detached and terraced houses. A bungalow is a house with only a ground floor and no stairs; a detached house is a house that does not share party walls with any other dwellings (a party wall is a dividing partition between two adjoining buildings); a semi-detached house is one that is joined to another similar house on only one side; a terraced house is one of a row of similar houses joined together by their party walls. Flats are usually parts of a block with shared communal areas such as landings, stairwells and walkways. A studio flat combines living room, bedroom, and kitchenette into a single room; Halls of residence are purpose-built flats only for students.

House or flats may contain different types of room such as bedroom, kitchen or bathroom. The conventional way of categorising dwellings is to look at the number of bedrooms. For example, a 3-bedroom house has three bedrooms. A bathroom is a room with a toilet, sink and either a shower or bathtub. An en-suite bedroom is a bedroom that has a bathroom directly attached to it. A family bathroom is a bathroom not attached to a bedroom, but with its door opening onto e.g. a corridor.

Question 1.

Answer to question 1

(a) The correct answer is:

(A) SPARQL (B) OWL (C) RDFS (D) RDF (E) XML (F) URI/IRI

[Question type: bookwork- Lecture notes]

(b) A possible answer:

i. A possible solution:

```
<rdf:RDF xmlns="http://www.cs.le.ac.uk/co7216/exam#"
...>
...
<Candidate rdf:ID="Bob">
  <applyFor rdf:resource="#Programmer"/>
  <hasEduBackground rdf:resource="#MSc_Advanced_Software_Engineering"/>
  <hasID rdf:datatype="xsd:string">003</hasID>
  <hasName rdf:datatype="xsd:string">Bob</hasName>
  <hasTechnicalSkill rdf:resource="#CShape"/>
  <hasTechnicalSkill rdf:resource="#HTML"/>
  <hasTechnicalSkill rdf:resource="#java"/>
  <hasTechnicalSkill rdf:resource="#SpringMVC"/>
  <hasPreviousEmployer rdf:datatype="xsd:string">Microsoft</hasName>
  <hasYearOfExperience rdf:datatype="xsd:int">4</hasYearOfExperience>
</Candidate>
...
</rdf:RDF>
```

Note: these are **NOT** the only correct solutions. Please give credit for other sensible answers.

[Question type: application]

ii. A possible solution:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix : <http://www.cs.le.ac.uk/co7216/exam#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

:Skill
a rdfs:Class .

:Framework
a rdfs:Class ;
rdfs:subClassOf :Skill .

:Tool
a rdfs:Class ;
rdfs:subClassOf :Skill .

:Language
a rdfs:Class ;
rdfs:subClassOf :Skill .
```

```

:hasTechnicalSkill
a rdf:Property ;
rdfs:domain :Candidate ;
rdfs:range :Skill .

:hasEduBackground
a rdf:Property ;
rdfs:domain :Candidate ;
rdfs:range :Degree .

:hasYearOfExperience
a rdf:Property ;
rdfs:domain :Candidate ;
rdfs:range xsd:int .

:Bachelor
a rdfs:Class ;
rdfs:subClassOf :Degree .

:hasName
a rdf:Property ;
rdfs:domain :Candidate ;
rdfs:range xsd:string .

:hasPreviousEmployer
a rdf:Property ;
rdfs:domain :Candidate ;
rdfs:range xsd:string .

:Position
a rdfs:Class .
:hasID
a rdf:Property ;
rdfs:domain :Candidate ;
rdfs:range xsd:string .

:Degree
a rdfs:Class .

:Master
a rdfs:Class ;
rdfs:subClassOf :Degree .

:Candidate
a rdfs:Class .

:applyFor
a rdf:Property ;
rdfs:domain :Candidate ;
rdfs:range :Position .

```

[Question type: application]

iii. Note: These are **NOT** the only correct solutions; all prefixes are omitted.

A possible SPARQL query for iii(1):

```

PREFIX rdf: ...
SELECT ?candidate
WHERE {
    ?candidate a :Candidate.
    ?candidate :applyFor :GraphicDesigner.
    ?candidate :hasTechnicalSkill :Photoshop,
    :CSS.
    ?candidate :hasYearOfExperience ?year.
    FILTER(?year>=2)
} LIMIT 2

```

A possible SPARQL query for iii(2):

```

SELECT ?candidate ?previous_employer
WHERE {
    ?candidate a :Candidate.
    ?candidate :hasEduBackground ?degree.
    ?degree a :Master.
    OPTIONAL {
        ?candidate :hasPreviousEmployer ?previous_employer
    }
}

```

A possible SPARQL query for iii(3):

```

SELECT (COUNT( DISTINCT ?skill) as ?count) ?class
WHERE {
    ?candidate :hasTechnicalSkill ?skill.
    ?skill a ?class.
    ?class rdfs:subClassOf :Skill.
}
GROUP BY ?class
ORDER BY ?count

```

[Question type: application])

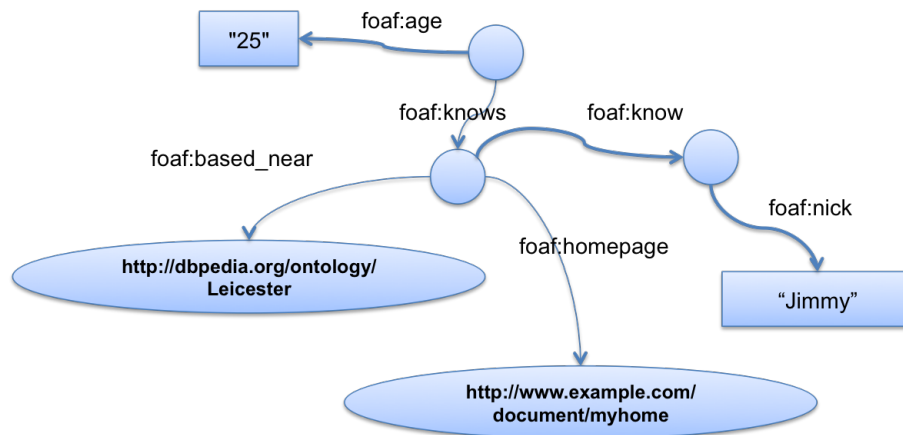
(c) The correct answer is:

(see the diagram on next page)

[Question type: application])

(d) The correct answers are:

- **rdf:ID** sets the subject URI that is unique within this document; **rdf:about** sets the subject URI of a statement, which may be absolute or resolved relative to the BASE of the document.
- **rdfs:domain** specifies the domain of a property **P**. The class of those resources that may appear as subjects in a triple with predicate **P**; **rdfs:range**, specifies the range of a property **P**. The class of those resources that may appear as values (of objects) in a triple with predicate **P**.



- `rdfs:Class` is the class of resources that are RDF classes; `rdf:type` relates a resource to its classes.
- `rdfs:predicate` is used in the reification vocabulary as part of `rdf:Statement`. It is an instance of `rdf:Property` that is used to state the predicate of a triple statement; `rdf:Property` is the class of RDF properties. `rdf:Property` is used to define a property in RDFS.

(e) A possible answer:

SPARQL:

```
PREFIX rdf: ...
SELECT (COUNT( DISTINCT ?instance) as ?count)
WHERE { ?instance rdf:type C}
```

Jena rules:

```
[subClassTransitivity:(?a rdfs:subClassOf ?b),
(?b rdfs:subClassOf ?c)
-> (?a rdfs:subClassOf ?c)]
[instanceSubsumption:(?a rdfs:subClassOf ?b),
(?b rdfs:subClassOf ?c), (?x rdf:type ?a)
-> (?x rdf:type ?c)]
```

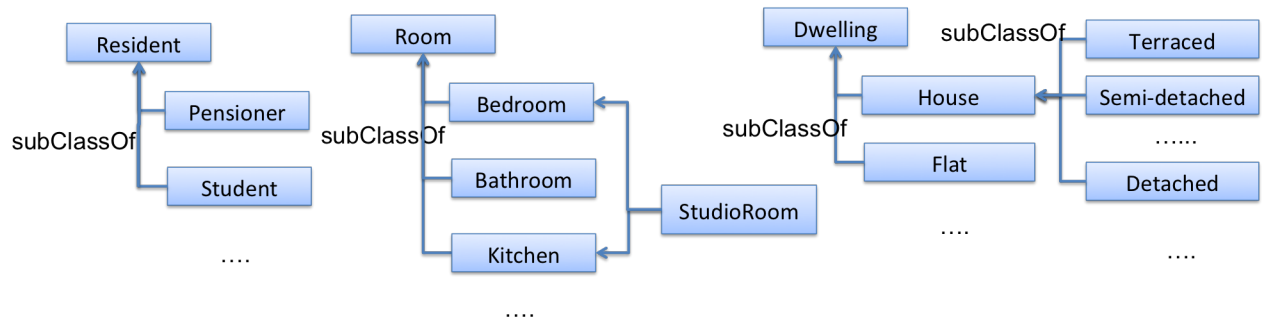
(f) A possible solution:

- A possible answer:

Concepts: House, Flat, Room, Bedroom ...
 Properties: hasRoom (Dwelling/Room), connectsTo (Room/Room),
 shareWallsWith (Dwelling/Dwelling), hasResident (Dwelling/Person) ..
 hasBedroom (Dwelling/Bedroom) is a sub property of hasRoom ..

[Question type: comprehension/application]

Note: These are **NOT** the only correct solutions.



Question 2.

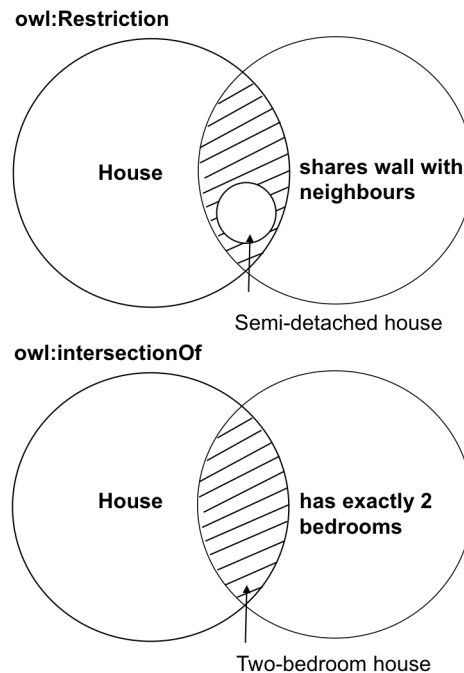
Answer to question 2

(a) A possible answer:

OWL 2 RL is used for applications that require scalable reasoning without sacrificing too much expressive power. You should avoid construct such as `DisjointUnion` or `ReflexiveObjectProperty`. OWL 2 RL supports all axioms of OWL 2 apart from disjoint unions of classes (`DisjointUnion`) and reflexive object property axioms (`ReflexiveObjectProperty`)

[Question type: bookwork]

(b) A possible answer: `rdfs:subClassOf` and `owl:Restriction` merely characterise a `SemiDetachedHouse` is (necessary conditions - must share walls with neighbours), whereas the `owl:intersectionOf` form defines a `TwoBedroomHouse` (necessary and sufficient condition). An example from dwelling scenario:



[Question type: comprehension]

(c) The correct answer is:

- i. **False:** Functional properties can also be data type properties (e.g. `hasAge`).
- ii. **False:** For example an instance of `Person` can property values for `hasMother` and `hasFather`, despite `hasMother` and `hasFather` are disjoint.
- iii. **False:** `owl:inverseOf` is not a class, it is a property.
- iv. **True**
- v. **True**

- vi. **False:** `owl:differentFrom` property links an individual to an individual. An `owl:differentFrom` statement indicates that two URI references refer to different individuals. It does not make all individuals pairwise different such as `owl:AllDifferent`.
- vii. **False** OWL Full is undecidable and currently there is no reasoner that supports reasoning of every feature of OWL Full.

[Question type: comprehension]

- (d) Note that these is **NOT** the only solution as SWRL/Jena rules can be written in different ways. Please give credit for any sensible answers. Possible answers:

(i) in SWRL:

```
hasSibling(?x, ?y)^hasChild(?y, ?z)^Male(?z)-> hasNephew(?x, ?z)
```

or written as a Jena rule:

```
[rules1:(?z rdf:type Male), (?x hasSibling ?y), (?y hasChild ?z)
-> (?x hasNephew ?z)]
```

(ii) in Jena rule syntax:

```
[rules1:(?p rdf:type owl:TransitiveProperty), (?x ?p ?y), (?y ?p ?z)
-> (?x ?p ?z)]
```

(iii) in Jena rule syntax:

```
[rule3: (?s ?p ?o), (?p rdfs:range ?c) -> (?o rdf:type ?c)]
```

[Question type: application]

- (e) A possible answer using RDFa

```
<div>
  <p>Lord <span property="foaf:name" typeof="foaf:Person" property="dbo:starring">
    Richard Attenborough</span>
    played the character of John Hammond in the
    1993 film <span typeof="dbo:Film">Jurassic Park</span>.
  </p>
</div>
```

[Question type: application]

- (f) The correct answer is.

303 URIs

Pros: very flexible, the redirection target can be configured separately for each resource.

Cons: multiple HTTP requests can cause latency.

- Hash URIs

Pros: reduces access latency by reducing the number of necessary HTTP round-trips.

Cons: a large amount of data being unnecessarily transmitted.

[Question type: comprehension]

(g) A possible answer:

```
<!-- http://www.cs.le.ac.uk/co7216/exam/#Ensuite -->

<owl:Class rdf:about="#EnsuiteBedroom">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="#Bedroom"/>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#connectsTo"/>
          <owl:someValuesFrom rdf:resource="#Bathroom"/>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
  <rdfs:subClassOf rdf:resource="#Bedroom"/>
</owl:Class>
```

Note: the XML serialization of the OWL may vary as axioms can be expressed differently in RDF/XML. Partial credits should be given if the student uses `Restriction`, `onProperty`, `someValuesFrom`, `intersectionOf`, `subClassOf` and `equivalentClass` correctly.

[Question type: application]

(h) The correct answers are:

i. A possible answer (Halls of residence) in OWL Manchester syntax:

```
Class: :HallsOfResidence
  EquivalentTo:
    Dwelling
      AND (:hasResident only :Student)
  SubClassOf:
    :Flat
```

ii. A possible answer: (2-Bedroom detached house) in OWL Manchester syntax:

```
Class: :FourBedroomDetachedHouse
  EquivalentTo:
    House
      AND (NOT (:sharePartWallWith some :House))
      AND (:hasBedroom exactly 2 :Bedroom)
```

iii. A possible answer: 4-Bedroom detached house with t least one family bathroom in OWL Manchester syntax:

```

Class: :Bathroom
EquivalentTo:
Room
    AND (:hasFacility some (:Toilet and :Sink and (:Shower or :ShowerTub) ))
DisjointedWith:
Bedroom, Kitchen

Class: :FamilyBathroom
EquivalentTo:
    Bathroom
    AND (NOT (:attachTo some :Bedroom))

Class: :FourBedroomHouseWithFamilyBathroom
EquivalentTo:
    House
    AND (:hasBedroom exactly 4 :Bedroom)
    AND (:hasRoom some :FamilyBathroom)

```

[Question type: bookwork/comprehension/application]

(i) Possible answers:

- Transitive property:
Room sameFloorAs Room;
Property isMoreExpensiveThan Property
...
- Symmetric property:
House shareWallWith House;
Room connectsTo Room
...
- Asymmetric property:
House hasMoreRoomThan House;
Room isLargerThan Room
...
- Disjoint property:
House hasBathroom Room
House hasKitchen Room;
isDetached and isNotDetached
..
- Reflexive property:
Flat locatedInTheSameBlockAs Flat
...
- Irrflexive property:
House isNeighbourOf House
...

- Functional property:
House hasNumberofBathroom xsd:int
...
- InverseFunctional property:
House hasAddress xsd:String
...

Note: These are **NOT** the only correct solutions. Please give credit for any sensible answers and please also consider partial credits for properties that are slightly off-topic.

Appendix A: jobs.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <candidates>
3    <candidate id="001" name="Thomas">
4      <position>Web Designer</position>
5      <years_of_experience>1</years_of_experience>
6      <education>BSc in Computing</education>
7      <technical_skills>
8        <language>JavaScript</language>
9        <framework>jQuery</framework>
10       <tool>Dreamweaver</tool>
11       <language>HTML</language>
12       <language>CSS</language>
13     </technical_skills>
14   </candidate>
15   <candidate id="002" name="Phoebe">
16     <position>Graphic Designer</position>
17     <years_of_experience>2</years_of_experience>
18     <education>MA in Creative Arts and Design</education>
19     <technical_skills>
20       <tool>Photoshop</tool>
21       <language>CSS</language>
22     </technical_skills>
23   </candidate>
24   <candidate id="003" name="Bob">
25     <position>Programmer</position>
26     <years_of_experience>4</years_of_experience>
27     <education>MSc in Advanced Software Engineering</education>
28     <previous_employer>Microsoft</previous_employer>
29     <technical_skills>
30       <language>Java</language>
31       <language>C#</language>
32       <framework>SpringMVC</framework>
33       <language>HTML</language>
34     </technical_skills>
35   </candidate>
36   <candidate id="004" name="Alice">
37     <position>Programmer</position>
38     <years_of_experience>1</years_of_experience>
39     <education>BSc in Computer Science</education>
40     <technical_skills>
41       <language>Java</language>
42       <language>JavaScript</language>
43       <framework>jQuery</framework>
44       <language>HTML</language>
45     </technical_skills>
46   </candidate>
47 </candidates>

```

Appendix B

(a) Namespaces

```
base="http://www.cs.le.ac.uk/co7216/exam/#"  
rdfs="http://www.w3.org/2000/01/rdf-schema#"  
owl="http://www.w3.org/2002/07/owl#"  
xsd="http://www.w3.org/2001/XMLSchema#"  
rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
foaf="http://xmlns.com/foaf/0.1/"  
dc="http://purl.org/dc/elements/1.1/"  
po="http://purl.org/ontology/po"
```

Default namespace:

```
http://www.cs.le.ac.uk/co7216/exam/#
```

(b) Classes and properties for Q2.e.

classes: foaf:Person, dbo:Film; properties: foaf:name, dbo:starring