

# Semantic Web

## Assignment 0. Demo Exam

Johannes Härtel

Iryna Dubrovskaya

Institute of Web Science and Technologies

Department of Computer Science

University of Koblenz-Landau

# 1 Multiple Choice (10 Points)

## 1.1 Linked Data Principles (5 points)

Mark correct and incorrect statements. You get 0.5 points for every box correctly checked; -0.5 points otherwise. Negative points do not carry over to the other sub-task.

Correct	Not correct	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Use URIs as names for things.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Include links by using URIs that dereference remote documents.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Do not use relational databases to provide RDF triple persistence.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Use HTTP URIs so those names can be dereferenced.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Use OWL Full as much as possible.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Use the XML serialization to represent RDF data.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Use one vocabulary per data set.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Return useful information upon lookup of those URIs.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Always use DBpedia as a bridge when defining links.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Enable a SPARQL endpoint to make URIs deferenceable.

## 1.2 Ontology engineering methodology (5 points)

Correct	Not correct	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Draw the ontology graphically before defining axioms.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Consider reusing existing ontologies (or parts of them).
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Define the facets of the slots.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Define the properties of the classes (slots).
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Define the classes and the class hierarchy.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Create instances (of classes).
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Define classes based on potential SPARQL queries.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Enumerate important terms in the ontology.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Determine the domain and scope of the ontology.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Define only foundation (or upper level) ontologies.

## 2 XML (10 points)

Consider the following document type definition (DTD):

```
1: <!DOCTYPE a [  
2:     <!ELEMENT a (b*, c)+>  
3:     <!ELEMENT b (c*, d*, e*)>  
4:     <!ELEMENT c (a*, d)+>  
5:     <!ELEMENT d (#PCDATA)>  
6:     <!ELEMENT e EMPTY>  
7: ]>
```

For each of the following XML documents, with respect to the above DTD, check whether the document is *well-formed* and *valid*.

If an XML document is not well-formed or not valid with respect to the above DTD, give all the reasons.

### 2.1 (5 points)

```
1: <a>  
2:     <b>  
3:         <c><d> Valid? </c></d>  
4:         <c><d></d></c>  
5:         <c><d/><d></c>  
6:     <b>  
7: </a>
```

**Answer:**

Malformed: first b never closed; first c closed before inner d; tags of last d reversed

### 2.2 (5 points)

```
1: <a>  
2:     <b><d>Valid</d></b>  
3:     <c>  
4:         <a><c><d>or</d></c></a>  
5:         <d>not</d>  
6:     </c>  
7: </a>
```

**Answer:**

Well-formed and valid

### 3 RDF and RDFS (15 points)

Consider the following RDF/XML document. The document contains information about Vancouver:

---

```

1: <?xml version="1.0" encoding="utf-8" ?>
2: <rdf:RDF
3:   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:   xmlns:xo="http://example.org/ontology#"
5:
6:   <rdf:Description rdf:about="http://example.org/resource/Vancouver">
7:     <rdf:type rdf:resource="http://example.org/ontology#City"/>
8:     <xo:name>Vanvouver</xo:name>
9:     <xo:latitude>49.25</xo:latitude>
10:    <xo:longitude>-123.1</xo:longitude>
11:    <xo:population>631,000</xo:population>
12:    <xo:state rdf:resource="http://example.org/resource/British_Columbia"/>
13:    <xo:mayor>
14:      <xo:Person rdf:about="http://example.org/resource/Gregor_Robertson"
15:        <xo:age>53</xo:age>
16:        <xo:assumedOffice>2008-12-08</xo:assumedOffice>
17:        <xo:party rdf:resource="http://example.org/resource/BC_NDP"/>
18:      </xo:Person>
19:    </xo:mayor>
20:  </rdf:Description>
21: </rdf:RDF>

```

---

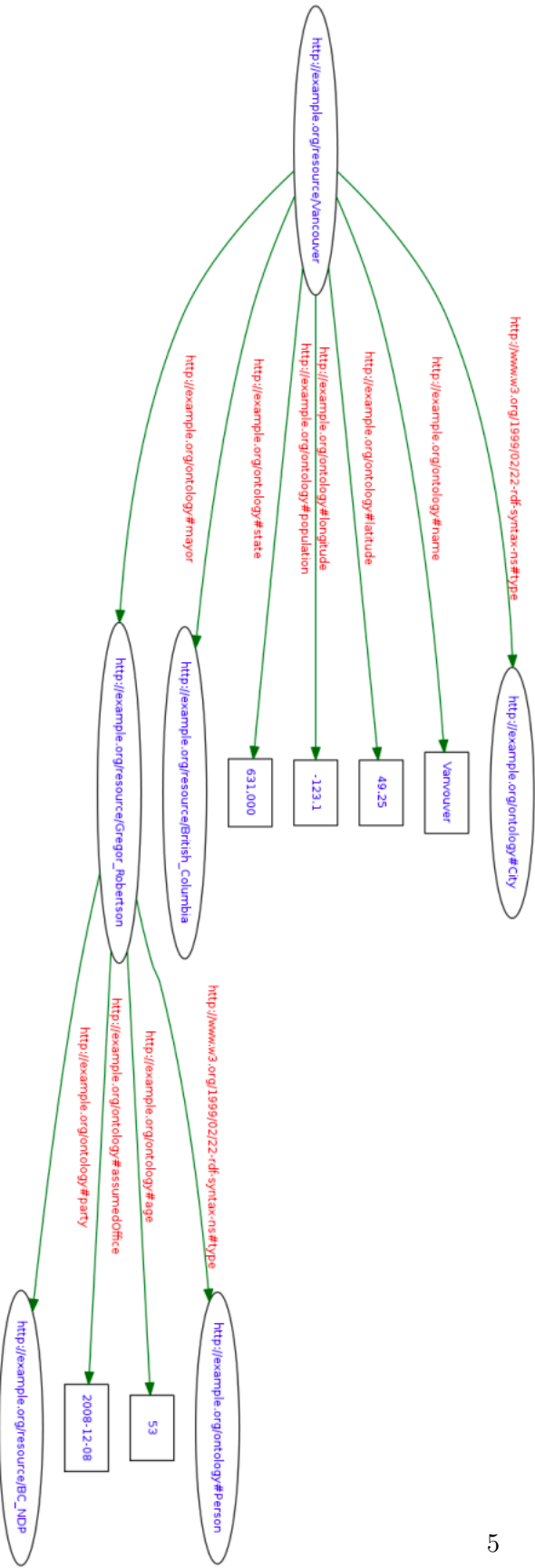
#### 3.1 (12 points)

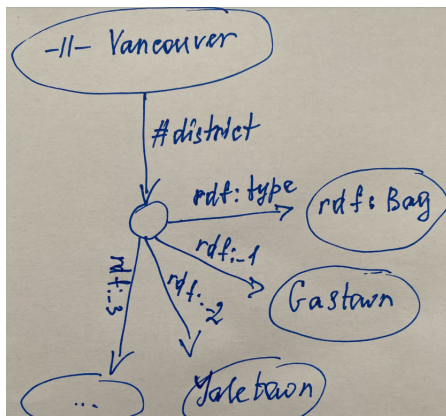
Translate above information into a visual RDF graph representation using the graphical notation presented in the lecture.

#### 3.2 (3 points)

We want to specify districts of Vancouver. Use an RDF container of your choice to group the following 3 districts: `x:Kitsilano`, `x:Gastown` and `x:Yaletown`. Express the information in graphical notation.

**Answer**





## 4 SPARQL (30 points)

Consider the excerpt of an RDF dataset on the next page containing touristic and geographical information.

Study the data structure apparent from the snippet. Based on that structure, formulate SPARQL queries for each of the following tasks.

Hints:

- Specify your queries as precisely as possible.
- View above listing as just an excerpt. Expect there to be other cities, states, countries and their points of interest.
- You can assume the used prefixes to be known. You don't need to declare them.
- Expect numerical literals to be typed accordingly, even without specific types annotations to them.

### 4.1 (4 points)

Specify a query that performs a check for existence for a point-of-interest 'Seattle Great Wheel' in the city of Seattle (x:city2) of type xo:FerrisWheel.

**Answer:**

---

```
1: ASK WHERE {  
2:     x:city2 xo:pointOfInterest ?poi.  
3:     ?poi a xo:FerrisWheel;  
4:         xo:name      Seattle      Great  W h e e l .  
5: }
```

---

### 4.2 (5 points)

Specify a query that returns the population of each German city named 'Frankfurt'. Refer to x:Germany as country.

**Answer:**

---

```
1: SELECT ?p  
2: WHERE {  
3:     ?t xo:name      Frankfurt      ;  
4:     xo:population ?p;
```

---

```
5:         xo:state [ xo:country x:Germany ].
6: }
```

---

#### 4.3 (6 points)

Specify a query that returns the total area of all the states that are part of `x:Germany` or `x:USA`.

**Answer:**

---

```
1: SELECT (sum(?a) as ?total)
2: WHERE {
3:     ?s xo:area ?a.
4:     {?s xo:country x:Germany}
5:     UNION
6:     {?s xo:country x:USA}
7: }
```

---

#### 4.4 (7 points)

Specify a query that returns the name and population of each of the 5 most populous cities (ordered descendingly) within the state of `x:Hesse`.

**Answer:**

---

```
1: SELECT ?name ?p
2: WHERE {
3:     ?t xo:population ?p;
4:         xo:name ?name;
5:         xo:state x:Hesse.
6: }
7: ORDER BY DESC(?p)
8: LIMIT 5
```

---

#### 4.5 (8 points)

Specify a query that returns all the museums within a (geo-spatial) bounding box of 1 degree in width and height around location (lat: 41.5; long: 12.3).



**Answer:**

---

```
1:
2: SELECT ?m
3: WHERE {
4:   ?poi a xo:Museum;
5:         xo:latitude ?lat;
6:         xo:longitude ?long.
7:   FILTER (?lat -41.5 <= 0.5 && ?lat -41.5 >= - 0.5)
8:   FILTER (?long -12.3 <= 0.5 && ?long -12.3 >= - 0.5)
9: }
```

---

```

@prefix x:      <http://example.org/resource/> .
@prefix xo:     <http://example.org/ontology#> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:    <http://www.w3.org/2001/XMLSchema#> .

```

```
# countries and cities
```

```

x:city1      rdf:type      xo:City;
              xo:name       "Frankfurt";
              xo:population "736,000";
              xo:state      x:Hesse.
x:city2      rdf:type      xo:City;
              xo:name       "Seattle";
              xo:population "704,352";
              xo:state      x:Washington.
...

```

```
# states and regions
```

```

x:Hesse      xo:name       "Hesse";
              xo:area       "21,114.94";
              xo:country    x:Germany.
x:Washington xo:name       "Washington";
              xo:area       "184,665.27";
              xo:country    x:USA.
...

```

```
# points of interest
```

```

x:city1      xo:pointOfInterest x:poi11, x:poi12.
x:poi11      rdf:type          xo:Museum;
              xo:name          "Staedel Museum";
              xo:latitude      "50.173818";
              xo:longitude     "9.001271".
x:poi12      rdf:type          xo:Monument;
              xo:name          "Euro-Sculpture";
              xo:latitude      "50.010556";
              xo:longitude     "8.773361".
...
x:city2      xo:pointOfInterest x:poi21, x:poi22.
x:poi21      rdf:type          xo:Tower;
              xo:name          "Space Needle";
              xo:latitude      "47.4934893";
              xo:longitude     "-121.923892".
x:poi22      rdf:type          xo:FerrisWheel;
              xo:name          "Seattle Great Wheel";
              xo:latitude      "47.605239";
              xo:longitude     "-122.54923".
...

```

## 5 Description Logics (15 points)

Consider the following statements:

- i) An administrative unit may be municipality, city district or city and always has some residents.
- ii) In administrative terms municipalities, city districts and cities are to be distinguished from each other (pairwise).
- iii) An administrative official may be mayor, councillor or clerk.
- iv) Necessarily mayors govern over an administrative unit.
- v) A lord mayor is a mayor that governs a city having at least 100,000 residents.
- vi) A lord mayor elected in Hamburg, Bremen or Berlin also is a head-of-state.

### 5.1 (3 points)

Define a signature for the given sentences.

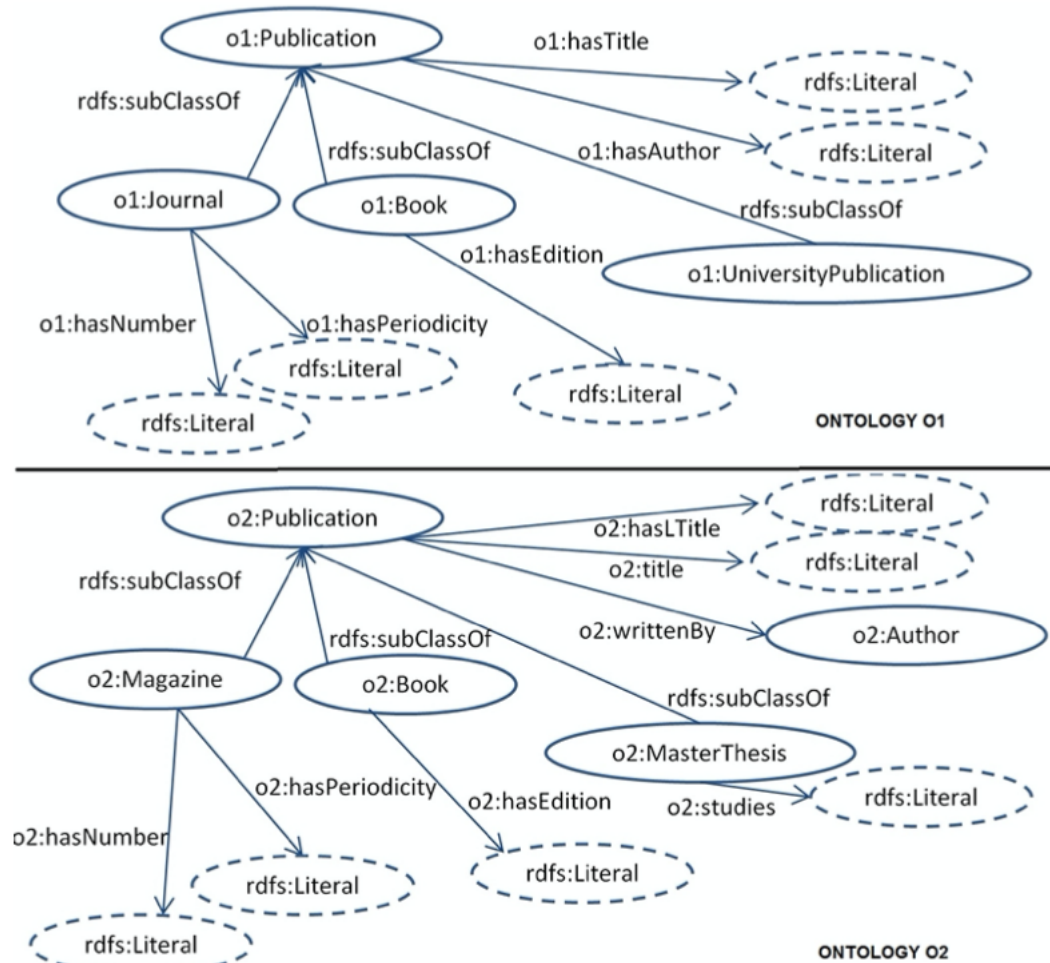
### 5.2 (12 points)

Formalize DL axioms for the following simplified city administration domain using its mathematical notation. Formalize your axioms as precisely as possible, using the implied concepts and roles.

- i)  $AdministrativeUnit \equiv (Municipality \sqcup CityDistrict \sqcup City) \sqcap \exists hasResident. \top$ . [2 Punkte]
- ii)  $Municipality \sqcap CityDistrict \sqsubseteq \perp$ .  $Municipality \sqcap City \sqsubseteq \perp$ .  $City \sqcap CityDistrict \sqsubseteq \perp$ . [3 Punkte]
- iii)  $Official \equiv Mayor \sqcup Councillor \sqcup Clerk$ . [1.5 Punkte]
- iv)  $Mayor \sqsubseteq \exists governs. AdministrativeUnit$ . [1.5 Punkte]
- v)  $LordMayor \equiv Mayor \sqcap \exists governs. (City \sqcap \geq 100000 hasResident. \top)$ . [2 Punkte]
- vi)  $LordMayor \sqcap \exists electedIn. \{Hamburg, Bremen, Berlin\} \sqsubseteq HeadOfState$ . [2 Punkte]

## 6 Ontology Alignment (10 points)

Consider the following two ontologies O1 and O2:



### 6.1 2.5 points)

Using string-based techniques, identify the mapping between the `o1:hasTitle` property and the corresponding element in O2 with minimum Levenshtein distance. Provide the calculations of the Levenshtein distance.

**Answer:**

`o2:hasLTTitle` (lev=1))

**6.2 (2.5 points)**

Using graph-based techniques, identify the mapping between the `o1:Journal` class and the corresponding element in `O2`.

**Answer:**

`o2:Magazine`. It is structurally similar to `o1:Journal`, as all of their children (leaves) are structurally similar.

**6.3 (2.5 points)**

What kind of mapping would you define between `o1:hasAuthor` and `o2:Author`?

- 
- ☐ subsumption.
  - ☐ equivalence.
  - ☐ disjointness.
  - ☐ does not apply.
- 

**Answer:**

"does not apply"

**6.4 (2.5 points)**

Calculate Levenshtein distance between `o1:hasEdition` and `o2:hasPeriodicity`. You can assume that both strings are converted into lowercase.

**Answer:**

lev = 8