OD: Lab 3

Hamid Latif Martínez
Göktuğ Cengiz

NOTE: Sometimes, when copying and pasting code from the PDF to somewhere else, some strange characters (like spaces) appear. This has happened in, at least, one query (first one of B part). Please be aware of this if any of the queries fails when pasted somewhere else.

A. Exploring DBpedia

1. Get the class representing an actor

```
SELECT DISTINCT ?s WHERE {
?s rdf:type owl:Class.
?s rdfs:label ?b. FILTER regex(?b, "actor")
}
```

The class representing an actor is obvious, even if there are 4 results.

2. Find super class of actor

```
SELECT ?r WHERE {
?s rdf:type owl:Class.
?s rdfs:subClassOf ?r
FILTER regex(str(?s), "ontology/Actor")
}
```

3. Find all actors in dataset

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?person
WHERE {
?person rdf:type dbo:Actor .
}
```

4. Get different classes defined as range of properties that have Actor defined as their domain

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?r WHERE {
?p rdfs:domain dbo:Actor .
?p rdfs:range ?r .
}
```

5. Find the superproperty of the goldenRaspberryAward property

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?superproperty
WHERE {
dbo:goldenRaspberryAward rdfs:subPropertyOf ?superproperty .
}
```

6. Properties that have the class Actor as either range or domain

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT * WHERE {
{
?classes rdfs:domain dbo:Actor .
}
UNION {
?classes rdfs:range dbo:Actor .
}
}
```

7. Return all persons that are not actors.

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?person WHERE {
?person rdf:type dbo:Person .
MINUS { ?person rdf:type dbo:Actor }
}
```

B. Analytical queries on top of QBAirbase

1. List the country, station type, latitude, and longitude details of each station.
Note: Limit the query to 25 results, and extract only the string values of the required object and not the whole IRIs.

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT str(?st_type) str(?countryP) ?long ?lat
WHERE {
     ?st property:type ?st_type .
     ?st property:longitudeDegree ?long .
     ?st property:latitudeDegree ?lat .
     {
          ?st schema:inCity ?city .
          ?city schema:locatedIn ?country .
     } UNION {
          ?st schema:inCountry ?country .
     }
     ?country property:country ?countryP
}
GROUP BY ?st_type ?country ?long ?lat
LIMIT 25
```

2. List the 10 highest averages of C6H6 emission and the country and the year on which they were recorded.

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT avg(?CH6) ?year str(?countryP)
WHERE {
     ?obs schema:C6H6 ?CH6 .
     ?obs schema:year ?yearN .
     ?yearN property:yearNum ?year .
     ?obs schema:station ?station .
     {  ?station schema:inCity ?city .
        ?city schema:locatedIn ?country .
     } UNION {
        ?station schema:inCountry ?country .
     }
     ?country property:country ?countryP .
}
ORDER BY DESC(?CH6)
LIMIT 10
```

3. For each city and property type, give the yearly average emission for NO2, SO2, PB,and PM10

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT str(?city) as ?city str(?type) as ?type avg(?NO2) as ?NO2
avg(?SO2) as ?SO2  avg(?PB) as ?PB avg(?PM10) as ?PM10
WHERE {
     ?obs schema:station ?station .
     ?station schema:inCity ?cityM .
     ?cityM property:city ?city .
     ?station property:type ?type .
     {
     ?obs schema:SO2 ?SO2 .
     } UNION {
     ?obs schema:NO2 ?NO2 .
     } UNION {
     ?obs schema:Pb ?PB .
     } UNION {
     ?obs schema:PM10 ?PM10 .
     }
}
GROUP BY str(?city) str(?type)
```

4. Define 3 additional SPARQL queries (and their corresponding interpretation) that you think could be interesting for the domain of analyzing air quality/pollution.

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?station ?long ?lat (avg(?pm25) as ?avgpm25) WHERE {
     ?obs schema:PM2.5 ?pm25 .
     ?obs schema:station ?station .
     ?station property:longitudeDegree ?long .
     ?station property:latitudeDegree ?lat .
     ?station property:type "Traffic"^^xsd:string .
     ?obs schema:year ?year .
     ?year property:yearNum "2010"^^xsd:integer .
     {  ?station schema:inCity ?city .
         ?city schema:locatedIn ?country .
     } UNION {
         ?station schema:inCountry ?country .
     }
     ?obs schema:sensor ?sensor .
     ?sensor property:statisticShortName "Mean"^^xsd:string .
     ?country property:isoCode ?isocode .
      FILTER( ?isocode IN ("ES"^^xsd:string) )
} GROUP BY ?station ?long ?lat
```

Annual mean PM2.5 concentrations observed at traffic stations in 2010 in Spain

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?yearN (avg(?pm10) as ?avgpm10) WHERE {
     ?obs schema:PM10 ?pm10 .
     ?obs schema:station ?station .
     ?station property:type "Background"^^xsd:string .
     ?station property:areaType ?class .
     ?obs schema:year ?year .
     ?year property:yearNum ?yearN .
     ?obs schema:sensor ?sensor .
     ?sensor property:statisticShortName "Mean"^^xsd:string .
     FILTER (?class in ("suburban"^^xsd:string, "urban"^^xsd:string) &&
?yearN >= 2010 && ?yearN <= 2012)
} GROUP BY ?yearN
```

Annual mean PM2.5 concentration observed at suburban background stations between years 2010 and 2012

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?yn (max(?c6h6) as ?maxc6h6) WHERE {
      ?s property:type ?b .
      ?s property:ozoneClassification ?class .
      ?obs schema:station ?s .
```

```
       ?obs schema:C6H6 ?c6h6 .
       ?obs schema:year ?year .
       ?year property:yearNum ?yn .
     ?obs schema:sensor ?sensor .
     ?sensor property:statisticShortName "Max"^^xsd:string .
       FILTER(?b = "Background"^^xsd:string && ?class in
("suburban"^^xsd:string, "urban"^^xsd:string) && ?yn >= 2000 && ?yn <=
2014)
} GROUP BY ?yn
```

Annual mean C6H6 concentration observed at suburban background stations

C. Ontology creation

C1. TBOX definition

1. Depending on how you created the TBOX, you need to provide either the SPARQLqueries you used for creating the TBOX (in case you used SPARQL), or in case you used another tool, the methodology/method you used and the output generated in a graphical form (the lecturer should not install any additional tool to validate this part).

For the TBOX creation, Protégé has been used. It allows to easily create the TBOX in simple steps:

1. Create entities: For this step, we must go to the “Entities” tab, and then go to the “Classes” subtab. After this, the subclasses must be added to the owl:Thing class, forming the entity hierarchy as we wish.

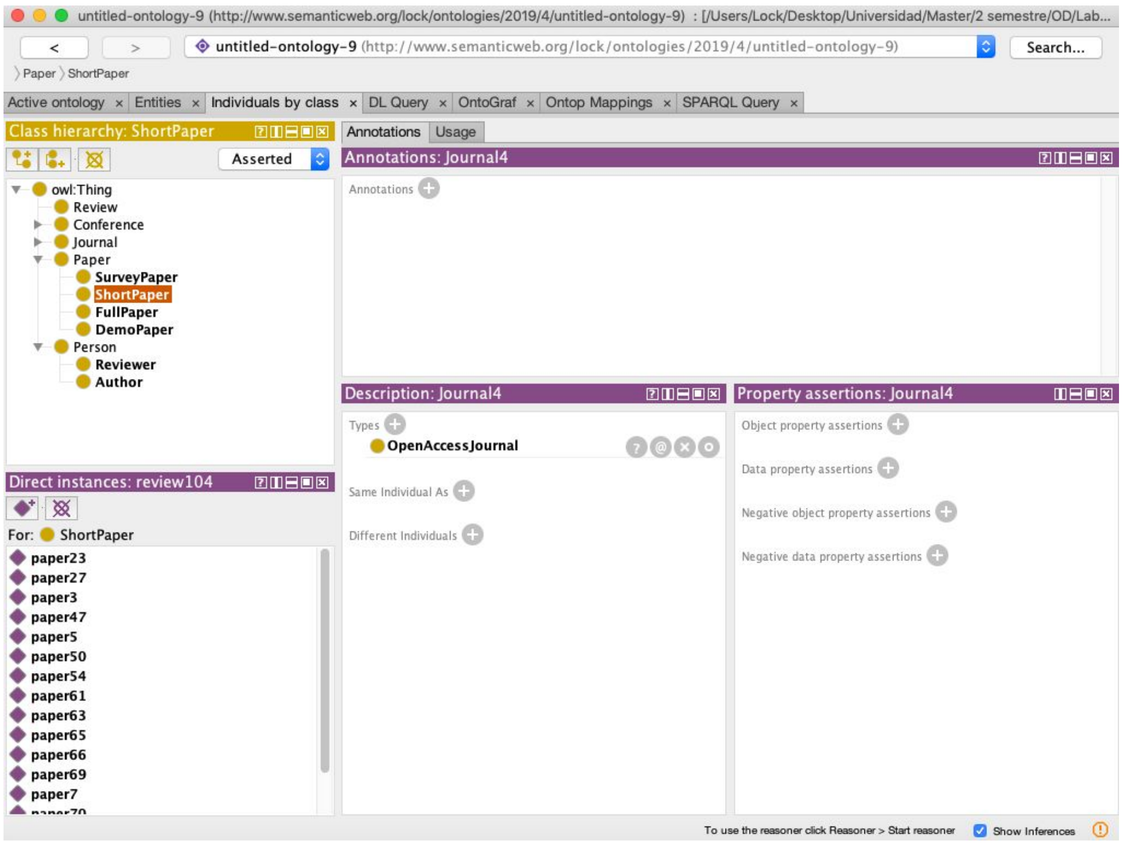


Figure 1: Creation of classes in Protégé

2. Create object properties: In order to create the object properties, we must go to the "Entities" tab, and then to the "Object properties" one. In this tab, the properties must be added as a subproperty of the owl:topObjectProperty one. After having created each of the properties, the domain and the ranges must be specified. To do that, the sections "Domain" and "Ranges". By clicking in the "+" icon, the classes of both the domain and range can be specified.

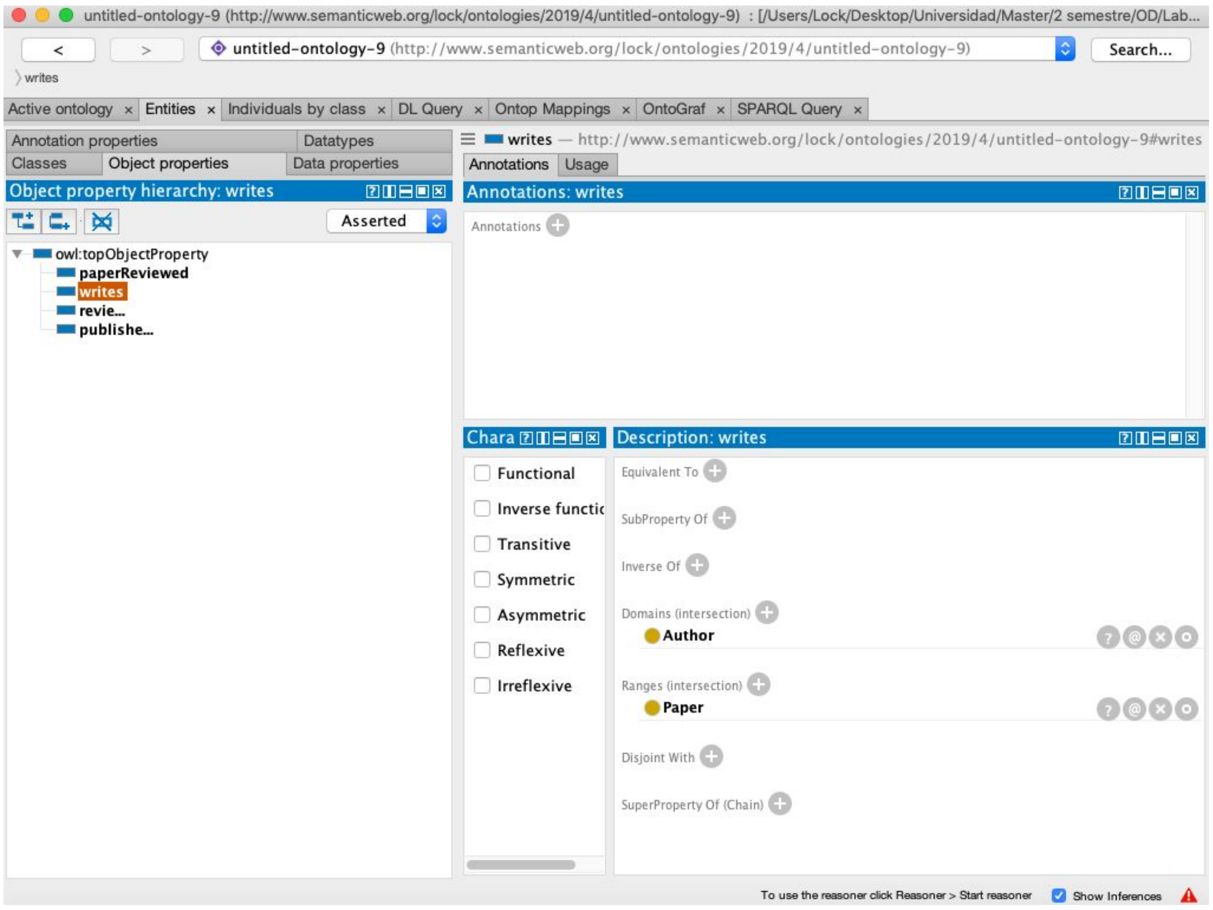


Figure 2: Creation of properties

2. Provide a visual representation of the TBOX.

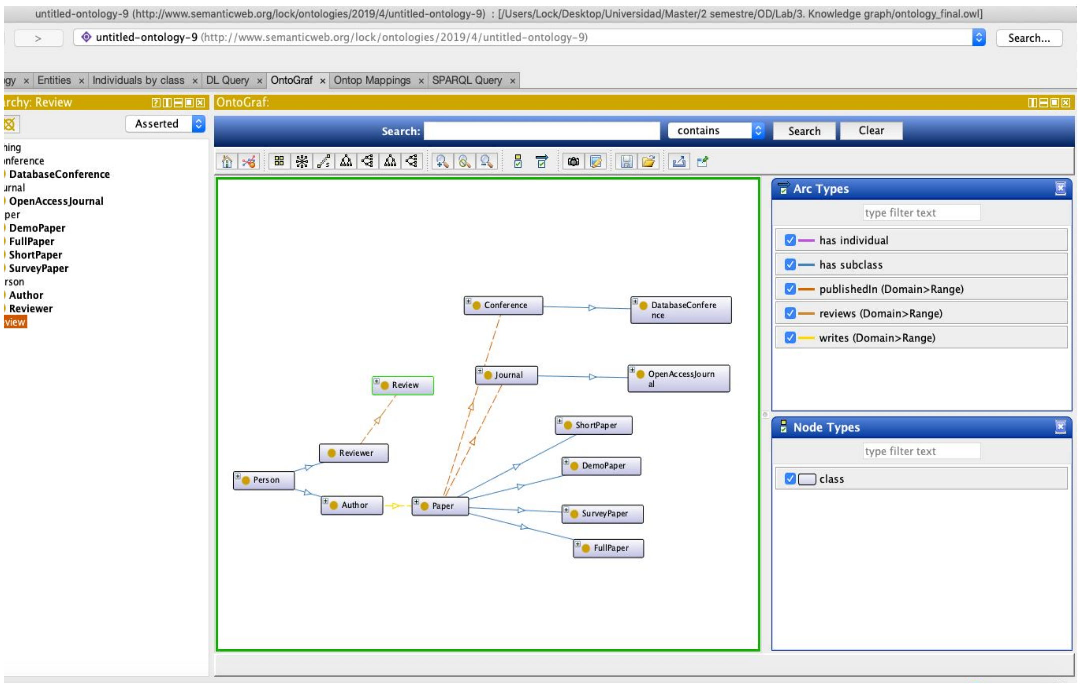
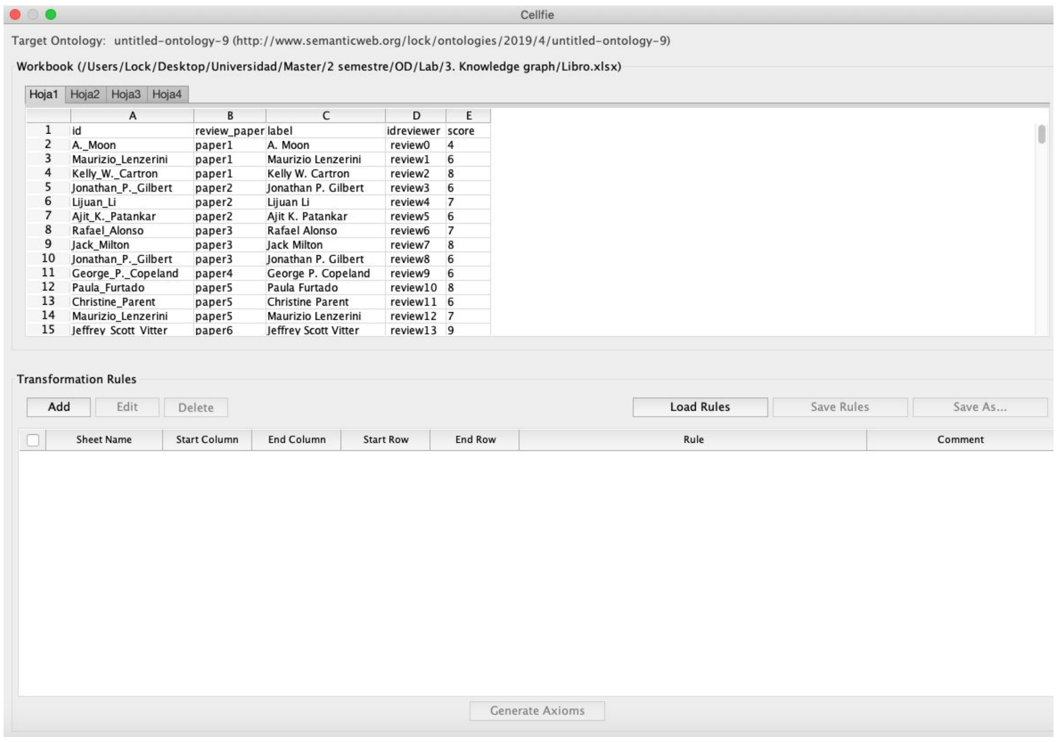


Figure 3: Visual representation of the TBOX

C.2 ABOX definition

1. Explain the method used to define the ABOX

The ABOX has been defined with Cellfie, which is a plugin that Protègè gives to load excel files to create the ABOX. Cellfie can be accessed in the Tools -> Create axioms from Excel workbook option.
Once started, Cellfie will require the route of the Excel file. After being specified, the sheets of the Excel file will be showed.



Target Ontology: untitled-ontology-9 (http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9)

Workbook (/Users/Lock/Desktop/Universidad/Master/2 semestre/OD/Lab/3. Knowledge graph/Libro.xlsx)

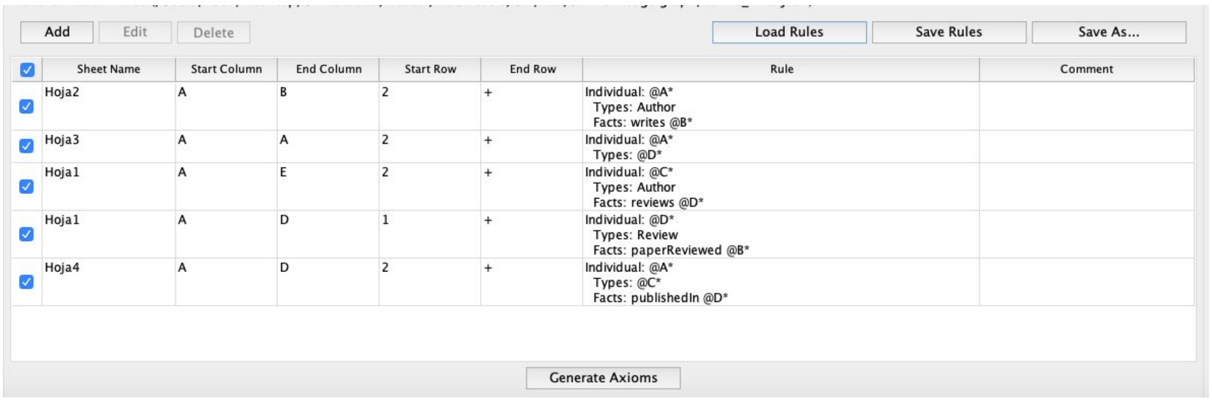
	A	B	C	D	E
1	id	review_paper	label	idreviewer	score
2	A._Moon	paper1	A. Moon	review0	4
3	Maurizio_Lenzerini	paper1	Maurizio Lenzerini	review1	6
4	Kelly_W._Cartron	paper1	Kelly W. Cartron	review2	8
5	Jonathan_P._Gilbert	paper2	Jonathan P. Gilbert	review3	6
6	Lijuan_Li	paper2	Lijuan Li	review4	7
7	Ajit_K._Patankar	paper2	Ajit K. Patankar	review5	6
8	Rafael_Alonso	paper3	Rafael Alonso	review6	7
9	Jack_Milton	paper3	Jack Milton	review7	8
10	Jonathan_P._Gilbert	paper3	Jonathan P. Gilbert	review8	6
11	George_P._Copeland	paper4	George P. Copeland	review9	6
12	Paula_Furtado	paper5	Paula Furtado	review10	8
13	Christine_Parent	paper5	Christine Parent	review11	6
14	Maurizio_Lenzerini	paper5	Maurizio Lenzerini	review12	7
15	Jeffrey_Scott_Vitter	paper6	Jeffrey Scott Vitter	review13	9

Sheet Name	Start Column	End Column	Start Row	End Row	Rule	Comment

Figure 4: Cellfie plugin

Now that the Excel file has been loaded, we must specify the transformation rules, which will determine how the classes will be created. These rules follow the Manchester syntax[1].

[1] https://www.w3.org/TR/owl2-manchester-syntax/



✓	Sheet Name	Start Column	End Column	Start Row	End Row	Rule	Comment
✓	Hoja2	A	B	2	+	Individual: @A* Types: Author Facts: writes @B*	
✓	Hoja3	A	A	2	+	Individual: @A* Types: @D*	
✓	Hoja1	A	E	2	+	Individual: @C* Types: Author Facts: reviews @D*	
✓	Hoja1	A	D	1	+	Individual: @D* Types: Review Facts: paperReviewed @B*	
✓	Hoja4	A	D	2	+	Individual: @A* Types: @C* Facts: publishedIn @D*	

Figure 5: Definition of rules in Cellfie

Now that the transformation rules have been defined, all that remains is to create the ABOX by clicking in the “Generate Axioms” button.

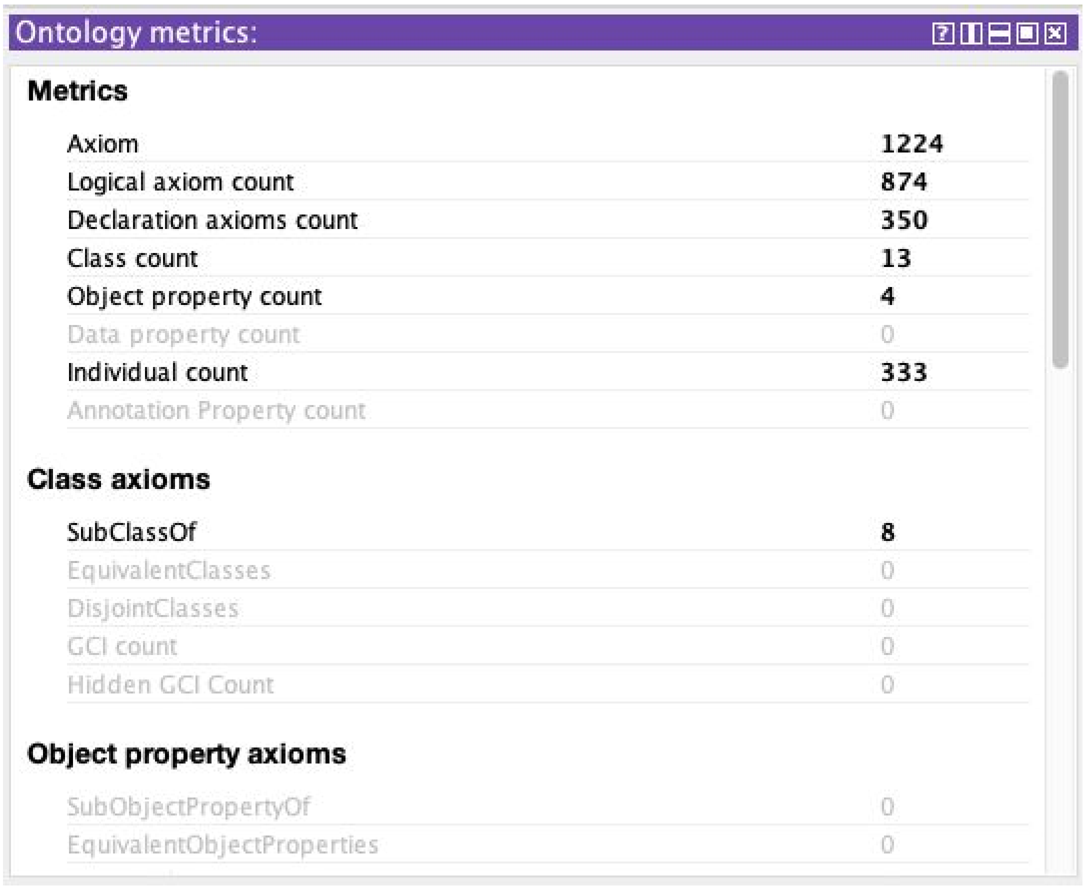
C.3 Linking ABOX to TBOX

1. Provide the SPARQL queries required to create the link between the ABOX and TBOX.

Protégé doesn’t need any extra commands to link the ABOX and the TBOX, so they won’t be used here.

2. Provide a summary table with simple statistics about the RDF graph obtained, e.g.,the number of classes, the number of properties, the number of instances, etc.

Protégé gives a nice summary of the ontology metrics in the main menu (Active ontology tab). In our case, this was the resulting table:


Ontology metrics:

Metrics	
Axiom	1224
Logical axiom count	874
Declaration axioms count	350
Class count	13
Object property count	4
Data property count	0
Individual count	333
Annotation Property count	0
Class axioms	
SubClassOf	8
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
Object property axioms	
SubObjectPropertyOf	0
EquivalentObjectProperties	0

Figure 6: Metrics provided by Protégé

C.4 Queries on top of the Ontology

1. Find all authors:

With TBOX:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT *
      WHERE { ?authors rdf:type od:Author}
```

Without TBOX:

As there is no TBOX, some previous knowledge of the graph must be assumed to be able to obtain the authors. In this case, we know that there's a "writes" property between the Author class and Papers, so we will exploit it:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT ?authors
      WHERE { ?authors od:writes ?papers}
```

The bad thing about this query is that the authors that have not written any paper will not appear.

2. Find all the properties whose domain is Author.

With TBOX:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT DISTINCT ?p WHERE {
?p rdfs:domain od:Author.
}
```

Without TBOX:

In this query we will assume that we know the writes property, which will allow us to query the instances of the Authors (as it is the only class with that property). With these, the properties whose domain is Author can be obtained by linking authors to any other class.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>
```

```
SELECT DISTINCT ?domains WHERE {
?authors od:writes ?papers .
?authors ?domains ?b .
}
```

3. Find all the properties whose domain is Conference or Journal.

With TBOX:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT DISTINCT ?proper WHERE {
{?proper rdfs:domain od:Conference.}
UNION {?proper rdfs:domain od:Journal.}
}
```

Result is empty because there is no property in the domain.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT DISTINCT ?proper WHERE {
{?proper rdfs:range od:Conference.}
UNION {?proper rdfs:range od:Journal.}
}
```

With range, the result is “publishedIn”

Without TBOX:

The property publishedIn, which references a Paper to a Journal or a Conference, can be used to obtain the Journal and Conference classes, which can then be used to obtain the properties whose domain is one of those two.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT DISTINCT ?domain WHERE {
?paper od:publishedIn ?journalconference.
?journalconference ?domain ?b.
}
```

Only rdf:type appears as a result for the same reason as before. In this case, the range can be printed. to see the results:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT DISTINCT ?range WHERE {
?paper od:publishedIn ?journalconference.
?b ?range ?journalconference.
}
```

In this case, publishedIn appears.

4. Find all the things that Authors have created (either Papers or Reviews).

With TBOX:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT DISTINCT ?things WHERE {
{?things rdf:type od:Paper.}
UNION {?things rdf:type od:Review.}
}
```

Without TBOX:

Previous knowledge of the ontology is assumed here, as we know that there exist the writes and the reviews poperties, whose domain is Author and whose range is Paper and Reviewer and Review, respectively. With that assumption, the things that Authors have created can easily be obtained by using these properties.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX od:
<http://www.semanticweb.org/lock/ontologies/2019/4/untitled-ontology-9#>

SELECT DISTINCT ?things WHERE {
{?authors od:writes ?things.}
UNION {?authors od:reviews ?things.}
}
```