

This lab sheet and all other materials can be found on GitHub here:

<https://github.com/QMUL-ECS735P/lab-week-6/>

OWL, Ontologies, and Protege pt.2

Session objectives:

- Craft OWL ontologies by hand
- Continue getting familiar with Protege
- Craft ontologies based on natural language statements

1. Introduction

In this lab we'll be continuing with Protege. As with last week, the lab machines already have Protege installed. To run it, simply open a terminal session and type:

```
protege
```

If you want to use your own computer, you can download Protege here:

http://protegewiki.stanford.edu/wiki/Install_Protege5

The Protege wiki and University of Manchester tutorials would be useful to keep on hand, **refer to them both when you get stuck**:

- http://protegewiki.stanford.edu/wiki/Main_Page
- <http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/>

There is some additional documentation you should keep handy while working through the lab today; these are various pages from the OWL 2 docs:

- OWL2 primer: <http://www.w3.org/TR/owl2-primer/>
- OWL2 functional-style syntax: <http://www.w3.org/TR/owl2-syntax/>
- OWL2 to RDF mapping: <http://www.w3.org/TR/owl2-mapping-to-rdf/>

2. Creating OWL ontologies

Your first task is to design an ontology using OWL 2 that can expression the statements below. To do this, you need to write your ontology using Turtle RDF syntax in a text editor of your choice (do *not* use Protege for this task). The statements are as follows:

- Arthur is a brother of Mary
- Arthur is the only brother of Mary (hint: you can say that the class of brothers of Mary is equivalent to the class container only Arthur)
- The only brother of Mary is the father of Harry
- The only brother of Mary is the farther of the only sister of William

Start by identifying the **classess** you need, then identify the **properties**, then finally the **instances**. Remember to make use of OWL, RDFS, and FOAF ontologies.

Example: Looking at the first statement, we could create the following ontology:

```
ex:Male a owl:Class ;
    rdfs:subClassOf foaf:Person ;
    rdfs:label "A male person" .

ex:BrothersOfMary a owl:Class ;
    rdfs:subClassOf ex:Male .

ex:hasBrother a owl:ObjectProperty ;
    rdfs:domain foaf:Person ;
    rdfs:range ex:Male .
```

Using this ontology, we can begin to make statements about Arthur and Mary:

```
:arthur a ex:Male .

:mary a ex:Person .

:mary ex:hasBrother :arthur .
```

Note the conventions used here. Class names are always capitalised, while property names start with lower case letters.

This is not the only way to create this ontology. Consider valid alternatives, and continue with the ontology design. You should craft an ontology that allows you to write down RDF statements for all four statements.

We're now going to *rewrite* the ontology we just came up with using OWL 2's funtional-style syntax. It should be equivalent to the ontology you just crafted in Turtle RDF.

As an example, the functional-style declaration of the `Male` class could be:

```
Declaration(Class(ex:Maybe))
```

```
SubClassOf(ex:Male foaf:Person)
```

Refer to the OWL 2 to RDF mapping documentation to help you complete this task:

<http://www.w3.org/TR/owl2-mapping-to-rdf/>

Finally, we're going to rewrite the ontology *again* but using Protege this time. This should be quick and easy as we should have a very good understanding of the ontology we've created.

When you're finished save ontology twice, one with Turtle formatting and once with OWL Functional syntax. Navigate to **File** → **SaveAs** to open the prompt.

Inspect the two files in a text editor and compare them with the ontologies you created manually in the previous two steps. What differences are there?

3. Creating Ontologies from Natural Language

Now, similar to last week, you must construct an ontology in Protege to prove the following argument is valid:

- The only animals in this house are cats
- Every animal that loves to gaze at the moon is suitable as a pet
- When I hate an animal, I avoid it
- No animals are carnivorous unless they prowl at night
- No cat fails to catch mice
- No animals ever like me, except for those in this house
- Kangaroos are not suitable for pets
- Only carnivorous animals kill mice
- I hate animals that do not like me
- Animals that prowl at night always love to gaze at the moon
- Therefore, I always avoid kangaroos

As with before, start by collecting the necessary classes, properties, and individuals. Think back to last week and use the reasoner in Protege to ensure your ontology is consistent.