**Knowledge and Data: Practical Assignment 4**

**Modelling in OWL**

* YOUR NAME: Sebastião Manuel Inácio Rosalino
* YOUR VUNetID: sxx209

*(If you do not provide your NAME and VUNetID we will not accept your submission.)*

For this assignment you will be engineering and reasoning over your very own OWL ontology.

You are free to choose the domain (subject) of the ontology you are going to build (e.g. on nutritional value, recipes, supermarkets, food safety, health, restaurants, planes, trains and automobiles, developing countries, modern slavery, political parties, refugees, you name it...). Just be creative and choose a domain we have not seen in class yet.

We expect extensive answers for this assignment: give a full account of what you did, such that a peer would be able to reproduce your ontology. This means that you must explicitly state the new axioms (in some Human readable syntax, such as Manchester Syntax, so no ttl) and that we expect you to motivate your choices (usually 1-3 lines).

**IMPORTANT: You will have to hand in your ontology (as ttl) together with the notebook**

**Learning objectives**

At the end of this exercise you should be able to build and ontology and to reason over it:

1. You will be able to engineer an OWL ontology
2. You will be able to conceptualize a (small) domain
3. You will be able use conditions and property characteristics
4. You will be able to use a reasoner to infer implicit knowledge
5. You will be able to work in Protégé

**Preliminaries**

There are several tools which can be used to create and edit RDF and OWL files (in addition to your favourite text-based editor). For this assignment we urge you to use the open-source tool [Protégé](https://protege.stanford.edu/), which is a stand-alone editor that is very much tailored towards OWL ontology editing.

To install Protégé on your system, please take a look at the [installation instructions](https://protegeproject.github.io/protege/installation/).

Protégé is a complex tool with many options, only few of which we will need for this assignment. There are various resources available to get you started:

* Watch a short [Protégé Screencast Tutorial](https://vimeo.com/183829740) (created by Rinke Hoekstra)
* Check out the [Practical Guide To Building OWL Ontologies Using Protégé 5](https://www.researchgate.net/publication/351037551_A_Practical_Guide_to_Building_OWL_Ontologies_Using_Protege_55_and_Plugins) that uses the [Pizza ontology](https://protege.stanford.edu/ontologies/pizza/pizza.owl) to describe how to create ontologies using Protégé.
* Check the [Assignment 4 document](https://docs.google.com/document/d/1Dw2winjfr2TJq3r1q6ZRpud9Qn--tq4ioWDqiBdrrzI) containing Tips & Recommendations on how to create ontologies.

**Task 1 (1 Point) : Creating an empty ontology**

Create a new empty ontology in Protégé.

Be sure to:

* Choose your own unique ontology IRI, its namespace, and a prefix
* Add metadata in the form of *rdfs:label*, *rdfs:comment*, and *rdfs:isDefinedBy* annotations.  
  Use *rdfs:comment* to describe the domain and target audience of your ontology.

Write down the IRI, its prefix, and the metadata that you have added in the textfield below:

IRI: <http://www.semanticweb.org/sebas/ontologies/knowledge_and_data/assignment_4>

Namespace: ...  
Prefix: ...

Metadata: The data there's always there. Built-in protege things. For example, OWL:Thing. Add your properties. RDF Types, RDFS Syntax. Everything from RDF/RDFS semantics I've used. Properties of own entologies. Example Human label Human comment human being of Homosapiens. Use rdfs.label on every classes. Use comments. Labels for every class. Use everything from the Annotations

**Task 2a (1 Point) : Populating your ontology**

Populate your ontology such that it contains

* six classes
* three data properties
* four object properties

For **each** class, create **two** example instances:

* assert one instance as a member of this class (i.e. using *rdf:type*).
* leave the second instance without any type.

Your ontology should now have 12 instances: six instances with a certain *rdf:type*, and six instances without any *rdf:type*.

List and describe the 6 classes you created in the textfield below, together with their instances. Don't forget to motivate your choices.

Class Name:

Description:

Instances:

Class Name:

Description:

Instances:

Class Name:

Description:

Instances:

Class Name:

Description:

Instances:

Class Name:

Description:

Instances

Class Name:

Description:

Instances

**Task 2b (1 Point) : Asserting properties**

For *each* instance:

* assert at least one data property (e.g. ex:instanceA *ex:hasFullName* "Some Full Name").
* assert at least one object property, relating instances to each other (e.g. ex:instanceA *ex:attendsCourse* ex:instanceB)

List and describe 3 statements from your ontology, containing 3 different data property assertions:

Your answer here

List and describe 4 statements from your ontology, containing 4 different object property assertions:

Your answer here

**The reasoner**

The questions following this point make use of Protégé reasoning capabilities, which are available via plugins but which are disabled by default. Install (if necessary) and start the *Pellet* reasoner before you continue with the next question.

Refer to page 15 of the [Protégé guide](https://www.researchgate.net/publication/351037551_A_Practical_Guide_to_Building_OWL_Ontologies_Using_Protege_55_and_Plugins) for instructions on how to install and run the reasoner.

**Task 3 (0 Points): Reasoning on a basic ontology**

All assertions that were addded up til now were explicit. Yet, it is certainly possible that your ontology also contains one or more *implicit* assertions, that have emerged from the interactions between the added explicit assertions.

Run the reasoner on your yet-basic ontology and write down the inferences occurred (if any) below:

Your answer here

**Task 4a (1 Point): Necessary and sufficient conditions**

Select two classes from your ontology.  
For **both** classes:

* add necessary and sufficient conditions (e.g. engineers are people who hold an engineering degree, and any person holding an engineering degree is an engineer)
* infer class membership of *at least* one instance using each condition

List and describe the four conditions that you have added (i.e. axioms) in the textfield below. Do not forget to motivate your choices.

Your answer here

**Task 4b (1 Point): Reasoning over conditions**

Run the reasoner once again (after having added the two conditions).

Write down the different steps of the reasoning process (ie, what happens when you run the reasoner). Also write down and explain the resulting inferences below.

Your answer here

**Task 5a (1 Point): Property characteristics**

Add **four** different property characteristics.

Examples are

* transitivity
* symmetricity
* functionality
* (ir)reflexivity
* property chain
* disjoint properties
* etc

All four different characteristics can be asserted on one property (or on four different properties).

List and describe the four property characteristics you defined (i.e. axioms) in the textfield below. Do not forget to motivate your choices.

Your answser here

**Task 5b (1 Point): Reasoning over properties**

Run the reasoner once again (after having added the four properties).

Write down the steps of the reasoning process (ie, what happens when you run the reasoner). Also write down and explain the resulting inferences below.

Your answer here

**Task 6 (1-2 Point): Saving your ontology**

Go over your ontology to ensure that it is consistent and that it meets all requirements as asked through the various questions. If both criteria are met you will obtain an additional point (making a total of 2 points for this question). Note that there are several [online validators](http://mowl-power.cs.man.ac.uk:8080/validator/) that can help you check your ontology for consistency.

Next, export/save your ontology to a file using Turtle as serialization format. Use *save as* to ensure that later modifications won't end up in this file.

**Submit this file together with your notebook**

**Task 7 (1 Point): An inconsist ontology**

Add one or more axioms to your ontology in such a way that it becomes inconsistent. Note that the added axioms themselves *must* be consistent; adding a single inconsistent axiom is not allowed.

**IMPORTANT: do not submit this version**

List and describe the axioms that you added. Motivate your choices and explain why the ontology became inconsistent.

Save the good version first

Your answer here

**Submitting your answers**

To submit your answers for this assignment, create a zip-file containing both this notebook and your *consistent* ontology (saved during task 6). Name this zip-file **assignment\_4\_VUnetID.zip** (where VUnetID is of course to be replaced by your personal VUnetID, eg **rss220**), and submit it via Canvas.