

## ***Semantic Web Lab Assignment 5***

**Goal:** *Be able to create restrictions using Protégé OWL and perform SPARQL queries with reasoning.*

### **1. Protégé OWL Restrictions and Reasoning**

Work through the remainder of Chapter 4 of the Protégé Owl Tutorial v1.3 provided with Lab 4 (also at <http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/>). Recall that you worked through Section 4.7 in Lab 4.

Continue performing the instructions of Chapter 4 through 4.19 of the Tutorial using hints below since the tutorial is a bit different than the tool. Additionally, be ready to answer the provided questions in a class discussion (they will not be graded).

NOTE: If the tutorial says to select “Superclasses”, it is now named “SubClass Of”.

#### **Hints:**

##### **Exercise 16:**

**Step 2:** Select the plus next to “SubClassOf”, then...

**Step 3:** ...select the tab “Object Restriction Creator”.

##### **Exercise 17:**

**Step 1:** Select “hasBase” from the Restricted Property side of the view (“hasBase” is a subproperty of “hasIngredient”).

**Step 2:** Select “Some (existential)” in the “Restriction Type” area of the view.

**Step 3:** Select “PizzaBase” from the “Restriction Filler” side of the view.

**Exercises 19 & following:** Modify steps as exercise 16 and 17 above.

**Exercise 25:** Therefore, select the “HermiT 1.3.8.3” reasoner, then, to run it, select “Start Reasoner”. If you need to run it again, select “Stop Reasoner”, then repeat. Also, to run again, select the “Reasoner->Synchronize reasoner”, if present.

**Results** – the bad class shows up as *Equivalent* to the “Nothing” class.

##### **Exercise 26 & 27:**

**Step 2 & 3:** Select the edit function (far right of “Disjoint With”) rather than a row. Hold down the CTRL key and left select “Vegetable Topping” to remove the disjoint with statement.

**Step 4:** Select “Reasoner->None”, then select “Reasoner->Start Reasoner” to run it. You may use “Reasoner->Synchronize reasoner” if present.

**Exercise 30:** Select “Reasoner->None”, then select “Reasoner->Start Reasoner” to run it. You may use “Reasoner->Synchronize reasoner” if present.

**Exercise 31:**

**Step 5:** After you select “Only” near bottom of editor, you can select multiple items by holding down the CTRL key and selecting “CheeseTopping” and “VegetableTopping”. Or, instead of using the “Object Restriction Creator” tab like before, select the “Class Expression Editor” tab. You need to end up with the following:

**hasTopping only (CheeseTopping or VegetableTopping)**

NOTE: As you start typing words, pressing the tab key will try to complete words for you. You need to provide the parentheses.

**Exercise 35:** Like above, after selecting “Only” in the “Object Restriction Creator”, you can select multiple items by holding down the CTRL key and selecting.

**Exercise 36 & 37:** If the instructions don’t work, do it like Exercise 35.

**Exercise 39:**

**Step 6:** Select “Add Equivalent Classes”. In the popup, select the “Class Expression Editor” tab and type “Hot or Medium or Mild”. Then, select OK.

**Exercise 41, Step 3:** This is similar to **Exercise 39, Step 6** above.

NOTE: At the end of the tutorial, Protégé should only show “ProbeInconsistentTopping” as equivalent to “Nothing” in the Class Hierarchy (Inferred) view.

1 – Consider The Following:

The symbol “ $\equiv$ ” appears to the left of some classes in the class hierarchy view. What is protégé trying to tell you?

---

For more details of Manchester syntax (i.e., the current protégé symbology) see:

[http://protegewiki.stanford.edu/wiki/Manchester\\_OWL\\_Syntax](http://protegewiki.stanford.edu/wiki/Manchester_OWL_Syntax)

## 2 – Consider The Following:

In Section 4.16, you make a cardinality restriction and then use the “Edit->Convert to Defined Class” capability. How is the RDF associated with the “InterestingPizza” class different based on the section you put it in? What does the difference do? The “Window->Views->Ontology Views->RDF/XML Rendering” may be helpful.

---

## **2. Reasoning with a New Restriction**

- a. Create a new ontology in Protégé with namespace “utdswc:” that is stored in RDF/XML format. Save your project in a file named “lab5.owl” in an empty directory named Lab5\_2\_<YourID> where <YourID> should be replaced with your Net ID.

Below, the namespace “utdswc:” stands for “http://utdallas.semweb/class#”.

Copy the vCard ontology (the “vCard.owl” file provided with this lab) to the directory, add it as an import to your ontology (see instructions provided in Lab 4), and provide an ontology prefix of “vc” for the associated namespace. The “vCard.owl” file must be in the same directory as your “lab5.owl” file and must be imported as a direct import.

Your ontology now imports (i.e., includes) a vCard ontology created by “nwalsh”. Browse the classes and properties to get some understanding of his approach.

Add a subclass to the “vc:Name” class named “utdswc:Person”. Then, make “vc:Name” a subclass of “utdswc:Person”. Protégé will change the class browser to show they are equivalent and add a higher level entry for “utdswc:Person”. Look at the RDF/XML using “Window->Views->Ontology Views->RDF/XML Rendering” to see how it is represented.

You have effectively made the classes equivalent by making them subclasses of each other.

Make note of the inherited conditions of “utdswc:Person”—the constraints on names, prefixes, etc. Be ready to discuss in class.

b. Use the Protégé tool to create the following hierarchy and relationships:

Category	Triples to Create	Notes
Add Classes	utdswc:Man rdfs:subClassOf utdswc:Person . utdswc:Woman rdfs:subClassOf utdswc:Person . utdswc:Man owl:disjointWith utdswc:Woman .  utdswc:UnmarriedWoman rdfs:subClassOf utdswc:Woman . utdswc:MarriedWoman rdfs:subClassOf utdswc:Woman . utdswc:UnmarriedWoman owl:disjointWith utdswc:MarriedWoman .	Add disjoint classes Man and Woman as subclass of Person. Then, add disjoint classes MarriedWoman and UnmarriedWoman as subclasses of Woman.
Add Properties	utdswc:maidenName rdf:type owl:DatatypeProperty . utdswc:maidenName rdfs:domain utdswc:MarriedWoman . utdswc:maidenName rdfs:range http://www.w3.org/2001/XMLSchema#string .	Add property maidenName with domain MarriedWoman and range string.

c. Create a VCard for **Helen** by adding an individual of type “vc:Name” with identifier “Helen”. Then state Helen’s name by inserting the following triples:

```
(utdswc:Helen vc:fn "Helen Mirren")
(utdswc:Helen vc:given-name "Helen")
(utdswc:Helen vc:family-name "Mirren")
(utdswc:Helen utdswc:maidenName "Mironova")
```

We are using “vc:Name” instead of “vc:Vcard” since the related VCard name properties have domain “vc:Name”. Usually, you would create a “vc:VCard” individual and link through property “vc:name” to a blank node of type “vc:Name”. Then, you would link all the name components to the blank node.

You may exit protégé saving your Lab5.owl file.

d. Reason with an RDFS-Plus implementation (in Gruff)

In the following, you will capture screens of all information about **Helen** before and after reasoning with RDFS-Plus. You will paste your two screen captures into a presentation file (you may use LibreOffice Impress, MS Powerpoint, or Mac Keynote). Save/export your file as a “.pdf” (lab5.pdf) in the same directory as your “lab5.owl” file. The presentation file will only have two pages maximum.

Load your “lab5.owl” file into Gruff using a new repository with a name of your choice that is not in the same directory as the owl file.

Display all links to/from **Helen** by

- 1) right clicking on her node to add any missing connections and
- 2) positioning the nodes so any links *to* her come from above and any links *from* her leave below.

Organize the display nicely, capture the screen, and paste as the first slide in your presentation file.

Run Gruff's RDFS Reasoner by selecting "File->Miscellaneous->Apply RDFS++ Reasoner".

Again, display all links to/from **Helen** as above. Organize, capture the screen, and paste it as a second slide to your presentation file.

Save your presentation file as a PDF with the file name Lab5\_2\_<YourID>.pdf where <YourID> should be replaced with your Net ID.

### 3. Extra Credit (10 points):

If you want some extra credit, figure out if there is a way to get Protégé to create the same inferences about **Helen** as we performed in Gruff. Provide a documentation file (lab5\_3\_<YourID>.doc, .rtf, or similar) with numbered instructions on how to achieve the result. Document the before and after reasoning results in another presentation file and save/export as a PDF (lab5\_3\_<YourID>.pdf). <YourID> should be replaced with your Net ID.

Presuming you are starting at the same point as (2.d) above, but using Protégé instead. You may consider other plug-ins. Document all steps from starting Protégé, loading files, and selecting program functions to produce the same results as shown in Gruff—the before and after reasoning results.

***Please compress your entire directory containing the owl files, protégé project files, and Powerpoint file made in step 2 into one zip file named Lab5\_2\_<YourID>.zip where <YourID> should be replaced with your Net ID. Submit the zip file on eLearning.***

**Grading (100 points + 10 extra credit):**

- 100    Nothing submitted
- 5    Incorrect directory or filenames
- 25    The vCard.owl file not included
- 10    Lacking vCard import
- 10    vCard import from web rather than local file
- 10    Incorrect “utdswc:” namespace
- 10    Incorrect “vc:” namespace
- 10 each    Missing any required “utdswc:” or “vc:” property definition
- 5 each    Missing “utdswc:” or “vc:” domain definitions
- 5    Powerpoint has more or less than 2 images
- 10 to -30    Powerpoint shows more or less content than the before-and-after-reasoning should show (incorrect answers)

***Extra Credit:***

- +10    Instructions and presentation that yield the desired results