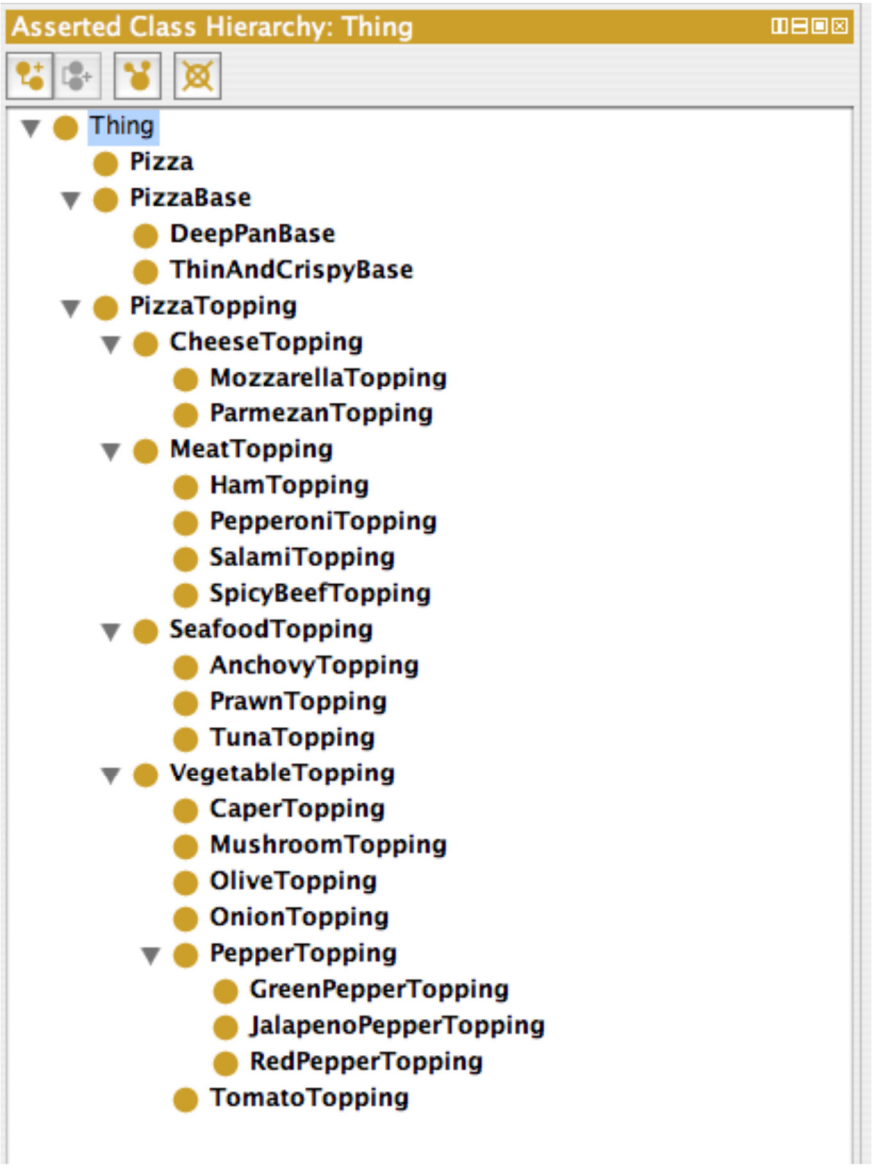
Lab exercise week 4

Complete the following exercises while waiting for your assignment to be marked.

Question 1 (Note: the exercises are based on materials from Protégé tutorial given in week 3).

1. Create the following classes using Protégé (see Figure 4.10 for Protégé Tutorial) – you can also follow instructions in Exercise 7 (page 21).



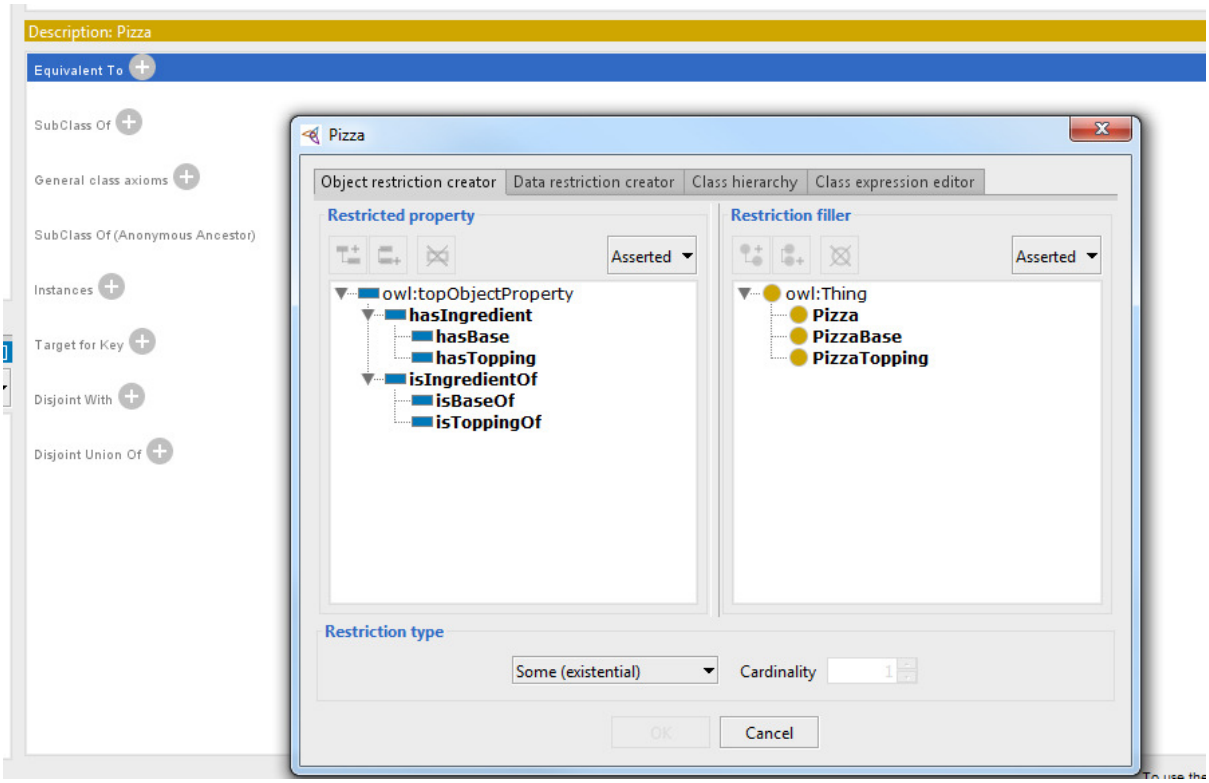
2. Create the following objective property

Name	Sub-property of	Inverse property of	Transitive	Functional	Range	Domain
hasIngredient			X			
hasTopping	hasIngredient				PizzaTopping	Pizza
hasBase	hasIngredient			X	Pizza	PizzaBase
isIngredientOf		hasIngredient	X			
isToppingOf		hasTopping				

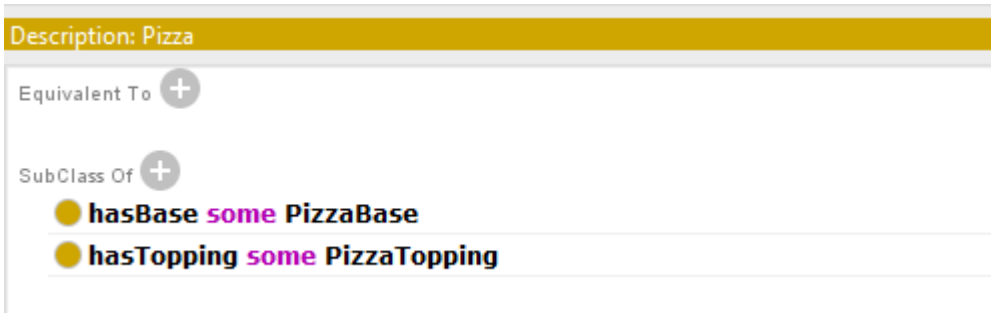
isBaseOf		hasBase			Pizza	PizzaBase

Question: By selecting the object property of hasBase as Functional property, what information can you tell from this information?

3. There are three types of restrictions: Quantifier restrictions, Cardinality restrictions, and hasValue restrictions.
   Quantifier restrictions can be further categorised into existential restrictions and universal restrictions.
   We will add existential restrictions to Pizza class:
   Click on SubClassOf: (see screen shot below) and express "Pizza hasBase some PizzaBase".

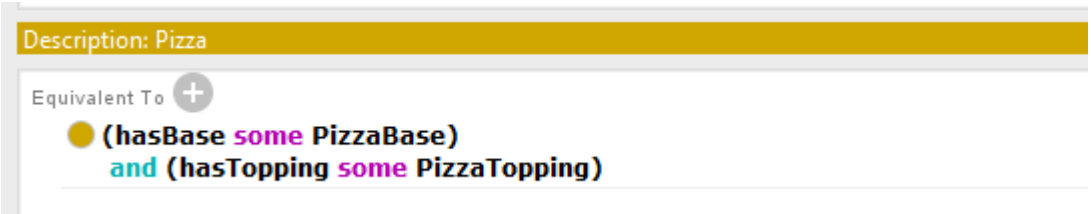


Repeat the step to show "hasTopping some PizzaTopping".



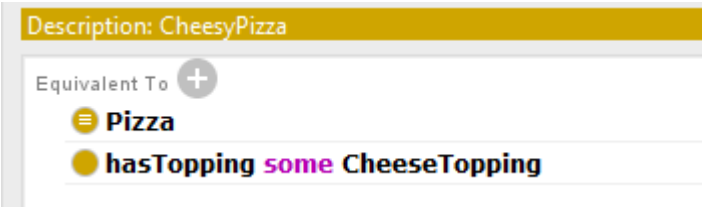
Now choose "Convert to defined class[1]" from the Edit menu. Your screen should look like below:

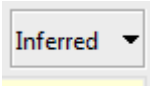
[1] Defined class is used to show necessary and sufficient condition, i.e. the class has a definition and any individual that satisfies the definition will belong to that class. In Protégé it is shown using this symbol 

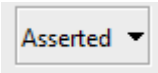


4. Create MozzarellaPizza. What can you see from the Desription window?
   Now assume Mozzarella Pizza has PepperoniTopping, MozzarellaTopping, TomatoTopping and PizzaBase. Repeat step 3 to define MozzarellaPizza. What is shown in the Equivalent To section?

5. Create different kinds of pizzas.
   Refer to Protégé Tutorial document, create different kinds of pizzas follows instructions from page 43 – 48.

6. Create a class called CheesyPizza (subclass of Pizza). Now add the following as shown in the screen shot below:
   

   Click Start Reasoner from the Reasoner menu (ensure one reasoned is selected at the bottom of the menu). What is the output?
   Click 

   (on the top right hand corner of the Class hierarchy window, you will see 

   click the down arrow and you should see Inferred).
   What is the output?

7. What should you input if you want to state ChessyPizza is a pizza that has cheese topping?

8. Create VegetarianPizza (see page 60). Ensure you complete up to Exercise 31 (page 61) to convert to defined class.

9. Check that MargheritaPizza has the description as shown in Figure 4.51 (see page 64) and convert to defined class.

10. Check for SohoPizza (page 65, Exercise 35) and AmericanHotPizza (page 66, Exercise 37).

11. Do Exercise 39 (page 69) to create ValuePartition class.

12. Do Exercise 40 (page 71) to add has Spiciness restrictions on PizzaTopping.

13. Do Exercise 43 (page 74) to create InterestingPizza.

14. Do Exercise 45 (page 75) to create FourCheesePizza.

15. Complete all exercises in Chapters 5, 6 and 7.

Question 2

Given the following example, create the ontology using Protege, run the Reasoner. What inference can you make?

```
<owl:Class rdf:ID="FemalePerson"/>
    <owl:Class rdf:ID="MalePerson"/>
    <owl:ObjectProperty rdf:ID="hasWife">
        <rdfs:subPropertyOf>
            <owl:ObjectProperty rdf:ID="hasSpouse"/>
        </rdfs:subPropertyOf>
        <owl:inverseOf>
            <owl:FunctionalProperty rdf:about="#hasHusband"/>
        </owl:inverseOf>
        <rdfs:range rdf:resource="#FemalePerson"/>
        <rdfs:domain rdf:resource="#MalePerson"/>
        <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    </owl:ObjectProperty>
    <owl:FunctionalProperty rdf:ID="hasHusband">
        <rdfs:subPropertyOf rdf:resource="#hasSpouse"/>
        <rdfs:domain rdf:resource="#FemalePerson"/>
        <rdfs:range rdf:resource="#MalePerson"/>
        <owl:inverseOf rdf:resource="#hasWife"/>
        <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </owl:FunctionalProperty>
    <FemalePerson rdf:ID="Martha">
        <hasHusband>
            <MalePerson rdf:ID="George"/>
        </hasHusband>
    </FemalePerson>
</rdf:RDF>
```

Question 3

Create a new class Person to ontology in Question 1, now add the following information, run the Reasoner. What inference can you make now?

```
<Person rdf:ID="Jane">
    <hasHusband>
        <Person rdf:ID="Harry"/>
    </hasHusband>
</Person>
```

Question 4

Make the class FemalePerson and MalePerson as subclasses of Person. Run the Reasoner. What inference can you make?