This lab sheet and all other materials can be found on GitHub here: https://github.com/QMUL-ECS735P/lab-week-5/

OWL, Ontologies, and Protege

> **Session objectives:**
>
> - Get familiar with Protege
> - Visualise class hierarchies
> - Build a simple ontology

1. Introduction

In this lab we'll be using Protege, a program for creating and exploring ontologies. The lab machines already have protege installed, to run it simply open a terminal session and type:

```
protege
```

and it will start up. If you want to use your own computer, you can download Protege here: http://protegewiki.stanford.edu/wiki/Install_Protege5

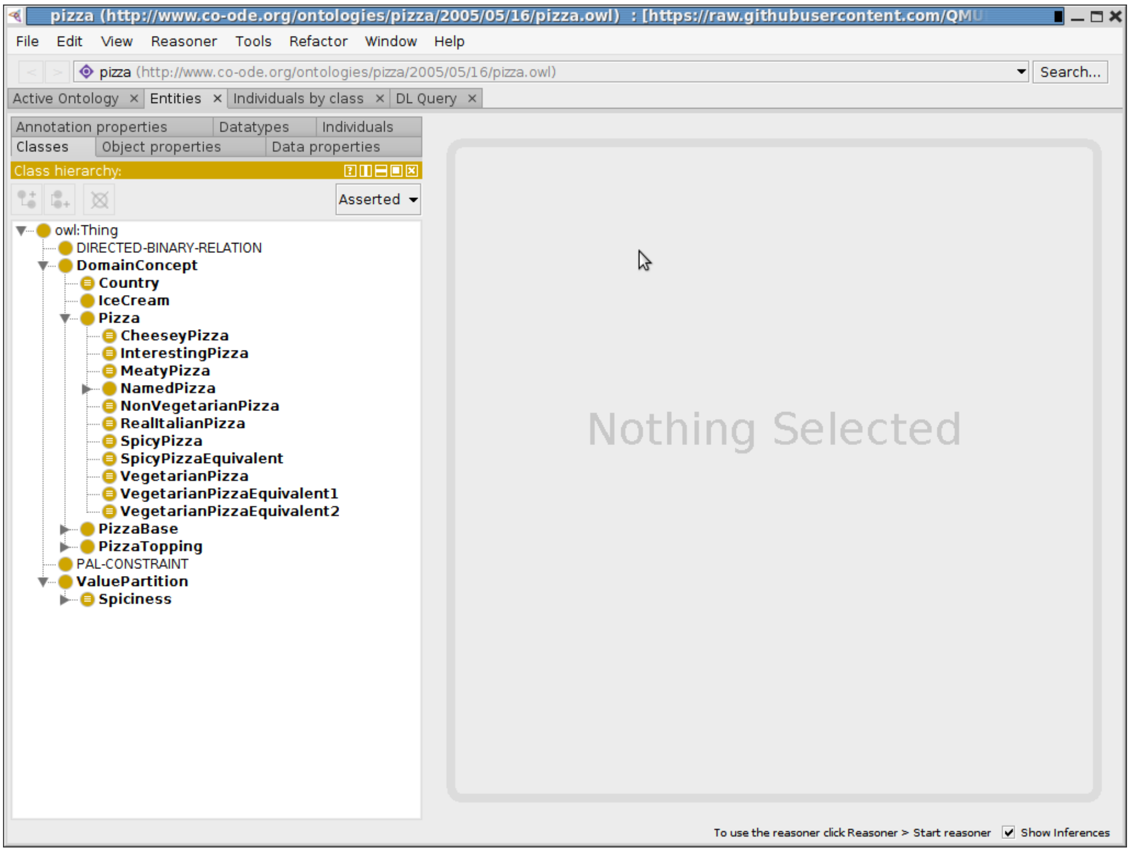
The Protege wiki and University of Manchester tutorials would be useful to keep on hand, refer to them both when you get stuck:

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

2. Examining OWL ontologies with Protege

You should have a `pizza.owl` file included in the lab materials; open Protege and then open this pizza ontology. You may also open Protege and use "Open from URL..." and paste in this URL: https://www.raw.githubusercontent.com/QMUL-ECS735P/lab-week-5/master/pizza.owl

Click the `Entities` tab and explore the class hierarchy. You should see something like this on the left pane:

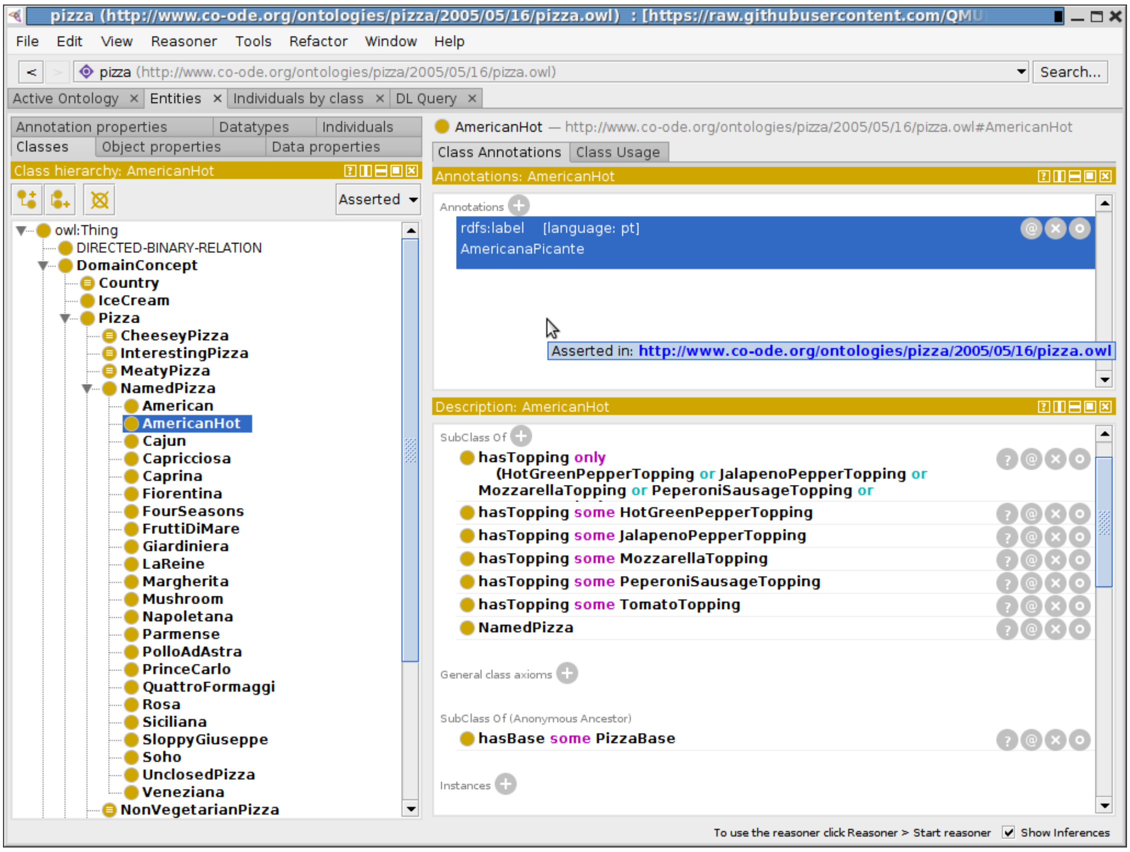


```
owl:Thing
  DIRECTED-BINARY-RELATION
  DomainConcept
    Country
    IceCream
    Pizza
      CheeseyPizza
      InterestingPizza
      MeatyPizza
      NamedPizza
      NonVegetarianPizza
      RealItalianPizza
      SpicyPizza
      SpicyPizzaEquivalent
      VegetarianPizza
      VegetarianPizzaEquivalent1
      VegetarianPizzaEquivalent2
    PizzaBase
    PizzaTopping
  PAL-CONSTRAINT
  ValuePartition
    Spiciness
```

Nothing Selected

To use the reasoner click Reasoner > Start reasoner ☑ Show Inferences

Select a specific class, such as AmericanHot, what can we learn about it?



```
owl:Thing
  DIRECTED-BINARY-RELATION
  DomainConcept
    Country
    IceCream
    Pizza
      CheeseyPizza
      InterestingPizza
      MeatyPizza
      NamedPizza
        American
        AmericanHot
        Cajun
        Capricciosa
        Caprina
        Fiorentina
        FourSeasons
        FruttiDiMare
        Giardiniera
        LaReine
        Margherita
        Mushroom
        Napoletana
        Parmense
        PolloAdAstra
        PrinceCarlo
        QuattroFormaggi
        Rosa
        Siciliana
        SloppyGiuseppe
        Soho
        UnclosedPizza
        Veneziana
      NonVegetarianPizza
```

AmericanHot — http://www.co-ode.org/ontologies/pizza/2005/05/16/pizza.owl#AmericanHot

Annotations

rdfs:label [language: pt]
AmericanaPicante

Asserted in: http://www.co-ode.org/ontologies/pizza/2005/05/16/pizza.owl

SubClass Of

- hasTopping only (HotGreenPepperTopping or JalapenoPepperTopping or MozzarellaTopping or PeperoniSausageTopping or
- hasTopping some HotGreenPepperTopping
- hasTopping some JalapenoPepperTopping
- hasTopping some MozzarellaTopping
- hasTopping some PeperoniSausageTopping
- hasTopping some TomatoTopping
- NamedPizza

General class axioms

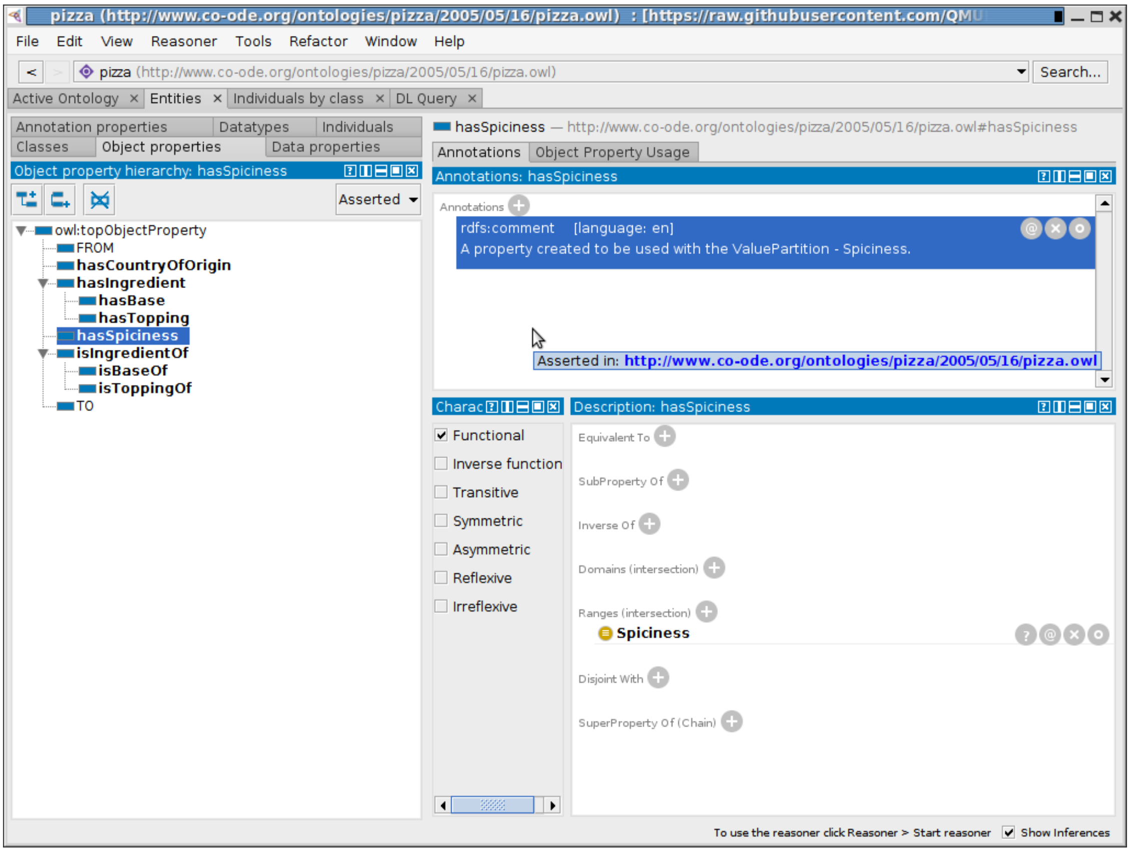
SubClass Of (Anonymous Ancestor)

- hasBase some PizzaBase

Instances

To use the reasoner click Reasoner > Start reasoner ☑ Show Inferences

Along with classes, we also have object properties. Navigate to the `Object properties` tab to see all the properties defined in this ontology.


pizza (http://www.co-ode.org/ontologies/pizza/2005/05/16/pizza.owl) : [https://raw.githubusercontent.com/QMU
File Edit View Reasoner Tools Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2005/05/16/pizza.owl)
Search...
Active Ontology Entities Individuals by class DL Query
Annotation properties Datatypes Individuals
Classes Object properties Data properties
Object property hierarchy: hasSpiciness
Asserted
owl:topObjectProperty
FROM
hasCountryOfOrigin
hasIngredient
hasBase
hasTopping
hasSpiciness
isIngredientOf
isBaseOf
isToppingOf
TO
hasSpiciness — http://www.co-ode.org/ontologies/pizza/2005/05/16/pizza.owl#hasSpiciness
Annotations Object Property Usage
Annotations: hasSpiciness
Annotations
Charac
Description: hasSpiciness
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Ranges (intersection)
Spiciness
Disjoint With
SuperProperty Of (Chain)
To use the reasoner click Reasoner > Start reasoner
Show Inferences

rdfs:comment [language: en]
A property created to be used with the ValuePartition - Spiciness.

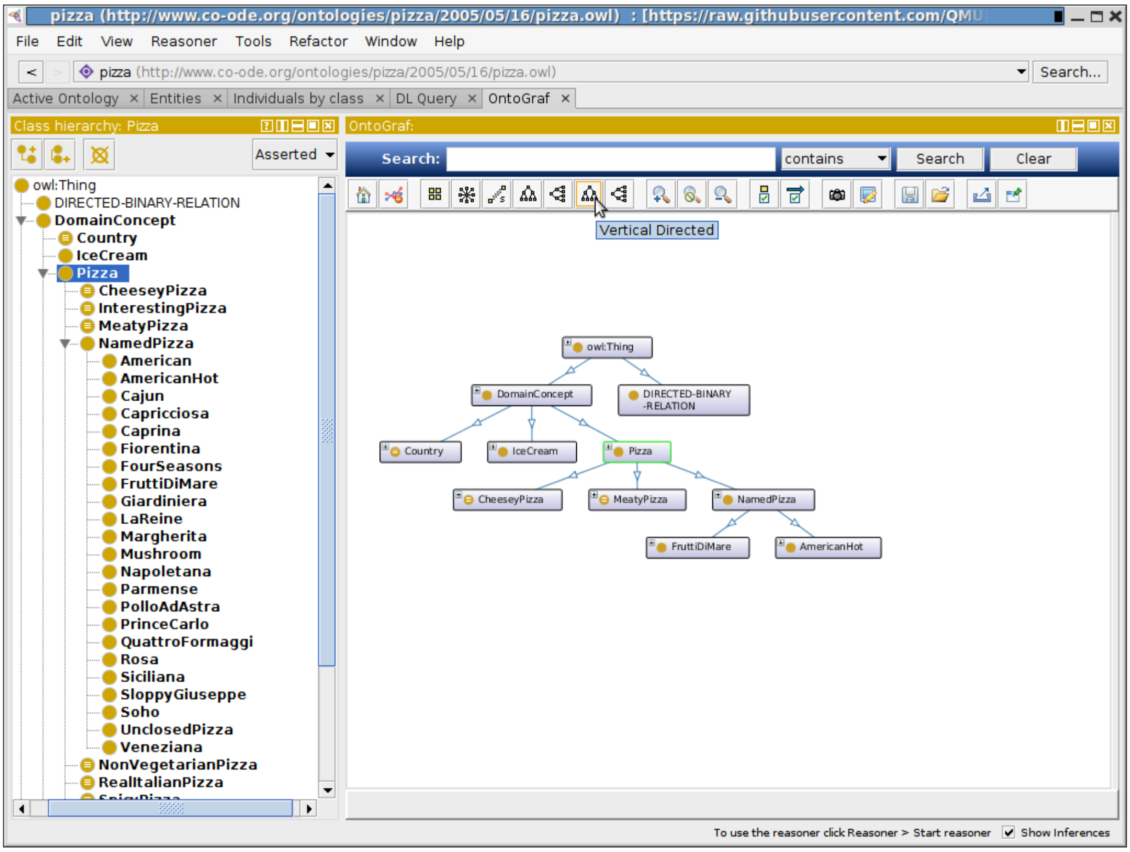
Asserted in: http://www.co-ode.org/ontologies/pizza/2005/05/16/pizza.owl

☑ Functional
☐ Inverse function
☐ Transitive
☐ Symmetric
☐ Asymmetric
☐ Reflexive
☐ Irreflexive

3. Visualising class hierarchies

Navigate to `Window -> Tabs -> OntoGraf` and then click the now visible `OntoGraf` tab. Start selecting some classes from the class hierarchy to place them into the OntoGraf display.

Along the top of the OntoGraf display are a number of options to change how the classes are laid out in the display. Below is an example of the "Vertical Directed" display. Experiment with a few different displays so you can see how the class hierarchy is ordered.



4. Creating ontologies with Protege

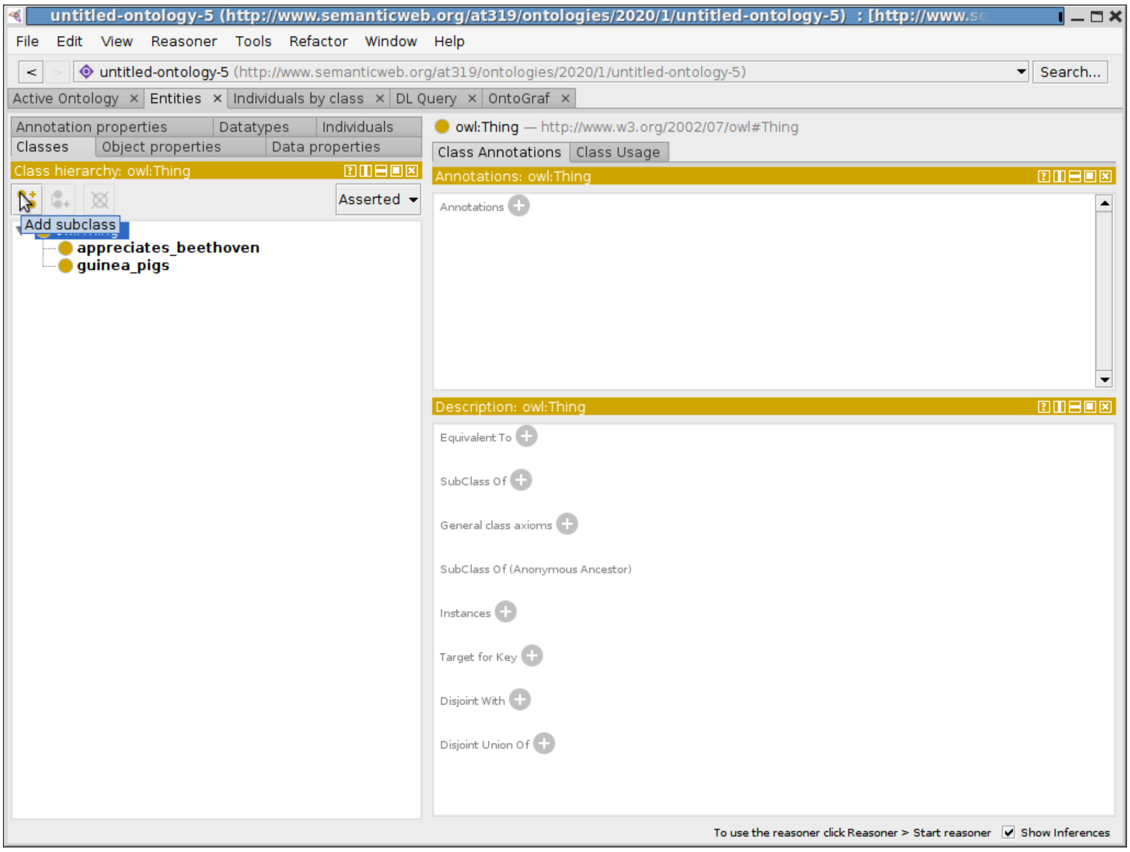
Your task is to create an ontology that formalises the following statements:

1. No one who appreciates Beethoven fails to keep silent when Moonlight Sonata is being played.
2. Guinea pigs are ignorant of music.
3. No one who is ignorant of music ever keeps silent when Moonlight Sonata is being played.
4. Therefore, guinea pigs don't appreciate Beethoven.

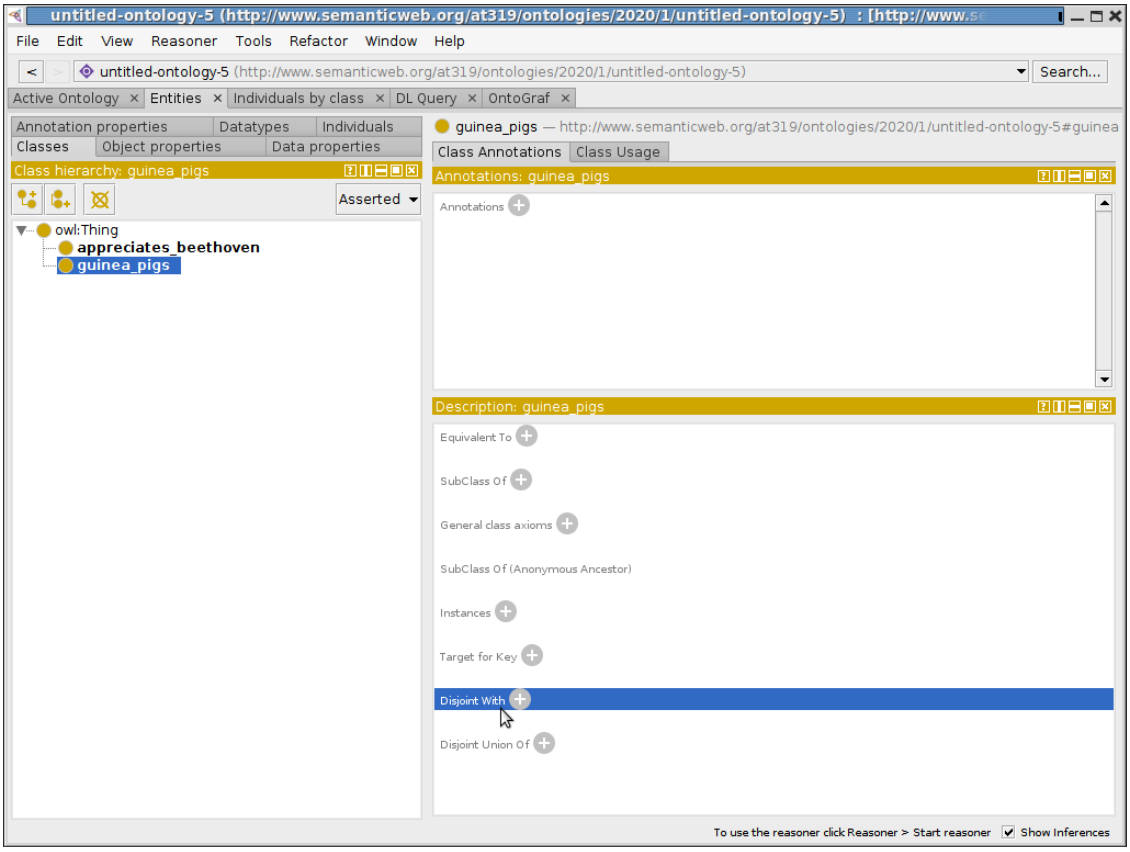
Take a moment to understand each of these statements, then identifiy the classes and object properties we'll need to formalise these statements in an ontology.

Create a new ontology by navigating to `File -> New` (or pressting `Ctrl+N`) and navigate to the `Entities` tab like before. To get you started, create two new classes:

- `appreciates_beethoven`
- `guinea_pigs` by clicking the `add subclass` button.


untitled-ontology-5 (http://www.semanticweb.org/at319/ontologies/2020/1/untitled-ontology-5) : [http://www.se
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-5 (http://www.semanticweb.org/at319/ontologies/2020/1/untitled-ontology-5)
Search...
Active Ontology × Entities × Individuals by class × DL Query × OntoGraf ×
Annotation properties Datatypes Individuals
Classes Object properties Data properties
Class hierarchy: owl:Thing
Asserted
Add subclass
appreciates_beethoven
guinea_pigs
owl:Thing — http://www.w3.org/2002/07/owl#Thing
Class Annotations Class Usage
Annotations: owl:Thing
Annotations
Description: owl:Thing
Equivalent To
SubClass Of
General class axioms
SubClass Of (Anonymous Ancestor)
Instances
Target for Key
Disjoint With
Disjoint Union Of
To use the reasoner click Reasoner > Start reasoner
☑ Show Inferences

We need to construct our ontology such we can prove `guinea_pigs` and `appreciates_beethoven` are **disjoint**.


untitled-ontology-5 (http://www.semanticweb.org/at319/ontologies/2020/1/untitled-ontology-5) : [http://www.se
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-5 (http://www.semanticweb.org/at319/ontologies/2020/1/untitled-ontology-5)
Search...
Active Ontology × Entities × Individuals by class × DL Query × OntoGraf ×
Annotation properties Datatypes Individuals
Classes Object properties Data properties
Class hierarchy: guinea_pigs
Asserted
owl:Thing
appreciates_beethoven
guinea_pigs
guinea_pigs — http://www.semanticweb.org/at319/ontologies/2020/1/untitled-ontology-5#guinea
Class Annotations Class Usage
Annotations: guinea_pigs
Annotations
Description: guinea_pigs
Equivalent To
SubClass Of
General class axioms
SubClass Of (Anonymous Ancestor)
Instances
Target for Key
Disjoint With
Disjoint Union Of
To use the reasoner click Reasoner > Start reasoner
☑ Show Inferences

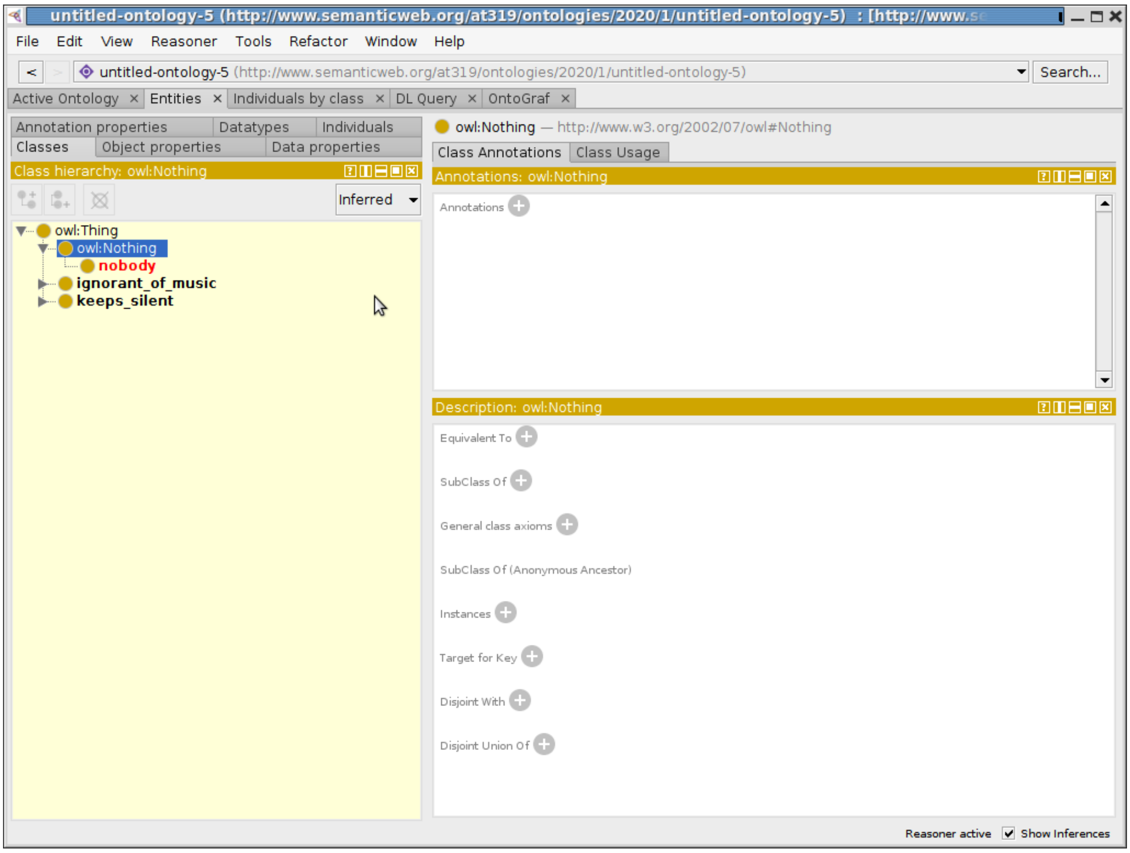
5. Using a reasoner in Protege

Protege can use several powerful OWL reasoners to determine whether our ontology is valid. Go to the `Reasoner` menu and select the HermiT reasoner if it is not already selected, then go back to the menu and start the reasoner.

We want to use the reasoner to prove that `guinea_pigs` and `appreciates_beethoven` are disjoint. To do this, create a new class `nobody` and define it as a `SubClass Of` both `guinea_pigs` and `appreciates_beethoven`.

Press `Ctrl+R` to re-run our reasoner, and then in the `Classes` tab click on the drop-down menu on the right and select "Inferred".

We should see our `nobody` class inferred as a subclass of the *bottom* concept (that's `owl:Nothing`).



untitled-ontology-5 (http://www.semanticweb.org/at319/ontologies/2020/1/untitled-ontology-5) : [http://www.se
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-5 (http://www.semanticweb.org/at319/ontologies/2020/1/untitled-ontology-5)
Search...
Active Ontology
Entities
Individuals by class
DL Query
OntoGraf
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Class hierarchy: owl:Nothing
Inferred
owl:Thing
owl:Nothing
nobody
ignorant_of_music
keeps_silent
owl:Nothing — http://www.w3.org/2002/07/owl#Nothing
Class Annotations
Class Usage
Annotations: owl:Nothing
Annotations
Description: owl:Nothing
Equivalent To
SubClass Of
General class axioms
SubClass Of (Anonymous Ancestor)
Instances
Target for Key
Disjoint With
Disjoint Union Of
Reasoner active
☑ Show Inferences