Ex1

An OWL ontology contains the following class hierarchy, properties and individuals:

Class hierarchy

- Place
    - Castle
        - HauntedCastle
    - BedAndBreakfast
    - GuestHouse
    - PerchedHut
- Entity
    - Ghost
    - Tree
- Purpose
    - Providing
- Object
    - Accomodation
    - Breakfast
- Country

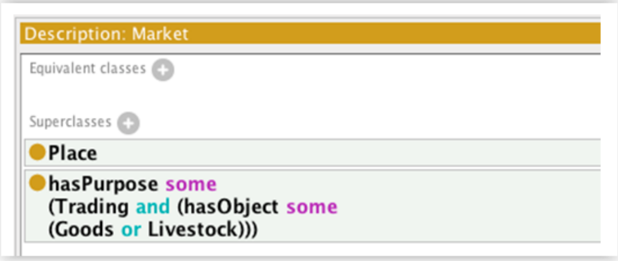
Properties

- locatedIn
- frequentedBy
- hasPurpose
- hasObject

Individuals

- Scotland (of the class Country)

Hint: Here is the description of a Market with a similar vocabulary:



Description: Market

Equivalent classes

Superclasses

Place

hasPurpose some
(Trading and (hasObject some
(Goods or Livestock)))

Write axioms to express the following elements of domain knowledge
Possible solutions in blue:

1. A haunted castle is a castle frequented by ghosts

   HauntedCastle ≡ Castle and frequentedBy min 2 Ghost

   HauntedCastle ⊑ frequentedBy min 2 Ghost

   HauntedCastle ≡ frequentedBy min 2 Ghost because HauntedCastle ⊑ Castle

2. Every castle located in Scotland is frequented by at least 2 ghosts

   Castle and (locatedIn value Scotland) ⊑ frequentedBy min 2 Ghost

3. A bed and breakfast is a place whose purpose is providing accomodation and breakfast and which is located in a guest house

BedAndBreakfast ≡ Place
and (hasPurpose some (Providing and hasObject some Accomodation))
and (hasPurpose some (Providing and hasObject some Breakfast))
and (locatedIn some GuestHouse)

4. A perched hut is a place located in a tree whose purpose is providing accomodation

   PerchedHut ≡ Place and (locatedIn some Tree)
   and (hasPurpose some (Providing and (hasObject some Accomodation)))

Ex2

1. Define the vocabulary for representing *roads* and *road lists*
   NB: Remember that a list is composed of a first element (a *road* for a *road list*) and a *rest* which is also a list (the initial list without the first element)

   Classes
   - Road
   - RoadList
     - EmptyList

   Object properties
   - first
   - rest

2. Using this vocabulary write axiom(s) for defining a *road list*

   RoadList ≡ ((first some Road) and (rest some RoadList)) or EmptyList

3. Define r1, r2 and r3 as specific roads, and axioms for representing:

   - lists containing r2 (in any position)

     RoadListWithR2 ≡ (first value r2) or (rest some RoadListWithR2)

   - lists containing r3 (in any position)

     RoadListWithR3 ≡ (first value r3) or (rest some RoadListWithR3)

   - lists containing r2 and r3 (in any position)

     RoadListWithR2AndR3 ≡ RoadListWithR2 and RoadListWithR3

   - lists containing r2 in first position

     RoadListWithR2First ≡ first value r2

   - lists containing r3 in thirld position.

     RoadListWithR3InThirld ≡ rest some (rest some (first value r3))

   NB: the lists defined in point 3 are also defined as subclasses of *RoadList*.

4. Test yours axioms with the following lists (see RoadLists.owl):

   - a list composed of r1 only

   - a list composed of r3 only

   - a list composed of r2 and r1 (in this order)

   - a list composed of r2 and r3 (in this order)

- a list composed of r1, r2 and r3 (in this order)

Ex3

Using the time ontology described at https://www.w3.org/TR/owl-time/
and available at https://raw.githubusercontent.com/w3c/sdw/gh-pages/time/rdf/time.ttl
define the following proper intervals :

- *Arrive*
- *Arrive_on_time* (for a course and a person)
- *Follow_a_course*
- *Check_email*
- *Check_email_at_right_time* (not during a course)

Hint: a simple way to use the vocabulary defined by the W3C time ontology is to import it into your own ontology. With *Protégé*, go to the *Active ontology* menu then to the *Ontology Imports* and the *Direct imports* tabs.

Arrive_on_time ≡ Arrive and
((time:intervalBefore some Follow_the_course) or (time:intervalMeets some Follow_the_course))

Check_email_at_right_time ≡
Check_email and
((time:intervalAfter some Follow_the_course) or (time:intervalBefore some Follow_the_course) or
(time:intervalMeets some Follow_the_course) or (time:intervalMetBy some Follow_the_course))