Technologies du web sémantique Automne 2022

C. Métral, G. Falquet  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Projet #2

Ontological Skiing

The goal of this project is to create an OWL ontology

* to precisely represent the rules of a domain
* as a support for the automatic inference of implicit facts

## 1. Modelling an OWL-2 ontology

1. Build an ontology, with the Protégé editor, that represents knowledge about a ski resort. Your ontology must contain classes, properties and axioms that correspond to the following UML diagram.



1. Extend your ontology by defining additional classes, properties, and axioms to represent the following facts
   1. a Blue\_Run is an ski run that is easy
   2. a Restaurant is an amenity of type “restaurant”
   3. a Quad is chair lift with a capacity of 4
   4. a Beginner\_Lift is a ski lift whose end (top) is at or nearby the start of an easy ski run
   5. a Food\_Place is a place that is nearby an amenity that serves food (restaurant, fast-food, ...)
   6. a Lift\_Restaurant is a Restaurant whose location is nearby the start or end of a ski run
   7. a Piste\_Restaurant is a Restaurant whose location is nearby a ski run (i.e. it is nearby a place that is in the geometry of a ski run)
   8. a Food\_2Steps\_Away place is a place from which one can reach a Food\_Place by taking two interconnected ski ways. Two ways are interconnected if the end place of the first way is the starting place of the second way or is nearby this starting place.
   9. a Food\_nSteps\_Away place is a place from which one can reach a Food\_Place by taking any number of interconnected ski ways
2. Run an OWL reasoner to check the consistency of the ontology

## 2. Creating a knowledge base

1. Export the ontology in RDF/Turtle and import it in a GraphDB repository with ruleset OWL-RL or OWL-Max
2. Create an ABox from the data available in the file resort.json and load it into your repository. Check that the resoner correctly found the instancs of the classes you defined.

## 3. Combining SPARQL querying with OWL inferences

Some queries may be hard or impossible to express in SPARQL (see examples below) and others are impossible to express in OWL. This part of the project aims to use OWL inferences to “help” the SPARQL query engine. The idea is to define classes whose members are inferred by the OWL reasoner and then to query these classes to solve queries. For example, to find the places from which one can reach a restaurant by taking a ski lift, one can define the class

PlaceWithLiftToRestaurant ≡

Place and isStartOf some

(SkiLift and (endsAt some (isLocationOf some Restaurant)))

and to find the place from which one can reach a restaurant by taking any number of ski lifts one can define

PlaceWithLiftSequenceToRestaurant ≡

PlaceWithLiftToRestaurant or

isStartOf some

(SkiLift and (endsAt some PlaceWithLiftSequenceToRestaurant))

1. Define classes, and related axioms, to answer the following questions (the instances of the class form the answer to the question)

* Find the places that can be reached, by taking a ski lift or a ski run, from the place(s) in *Start* (a subclass of *Place*)
* Find all the places that can be reached, by taking one or more ski lifts and ski runs, from the place(s) in *Start*
* Find the places that can be reached by average skiers from the *Start* places
* Find the places that can be reached from the *Start* places by passing only through places that have a restaurant

1. Import your class definitions in your GraphDB repository and execute SPARQL queries that demonstrate how they work.