Université de Genève

Technologies du web sémantiques
D400009

TP 3: Ontological Skiing

Author: Sajaendra Thevamanoharan

E-mail: sajaendra.thevamanoharan@etu.unige.ch

Author: Deniz Sungurtekin

E-mail: deniz.sungurtekin@etu.unige.ch

Janvier 2020

UNIVERSITÉ
DE GENÈVE
FACULTÉ DES SCIENCES
Département d'informatique

Introduction

Le but de ce travail pratique est de nous familiariser à l'élaboration de règles en SWRL afin d'améliorer la capacité de notre ontologie à inférer. Dans un premier temps, nous allons mettre à jour notre ontologie avec une classe Route et certaines propriétés nous permettant de simuler la durée du parcours d'une route qui est représentée par une séquence de SkiLift et SkiRun.

Ensuite nous utiliserons les règles SWRL pour permettre à notre "reasoner" d'inférer des propriétés sur nos Routes, puis nous utiliserons des insertions SPARQL pour ajouter ces inférences dans un fichier turtle qui sera lu par GrapheDB. Finalement, nous introduirons la gestions du temps dans notre stations de ski en créant une classe TimeInterval.

Méthodologie

Update your Ski ontology

1. Ajout de la durée

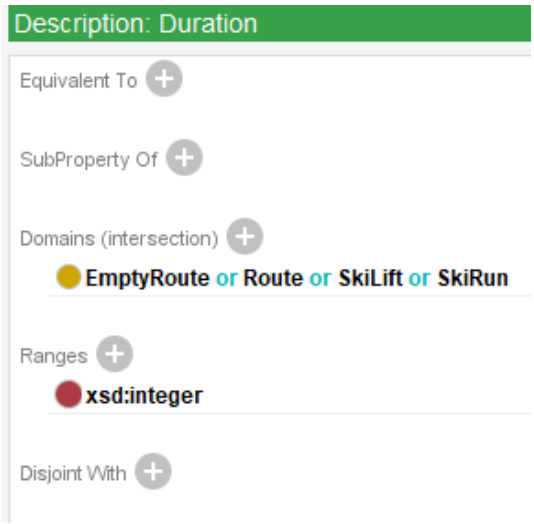

Figure 1: Caption

La durée est lié à une route (Ou une route vide que nous définirons dans la section suivante), un SkiLift ou un SkiRun. C'est une propriété des données puisqu'elle indique par un entier le temps nécessaire pour parcourir ces éléments.

2. Définition d'une route

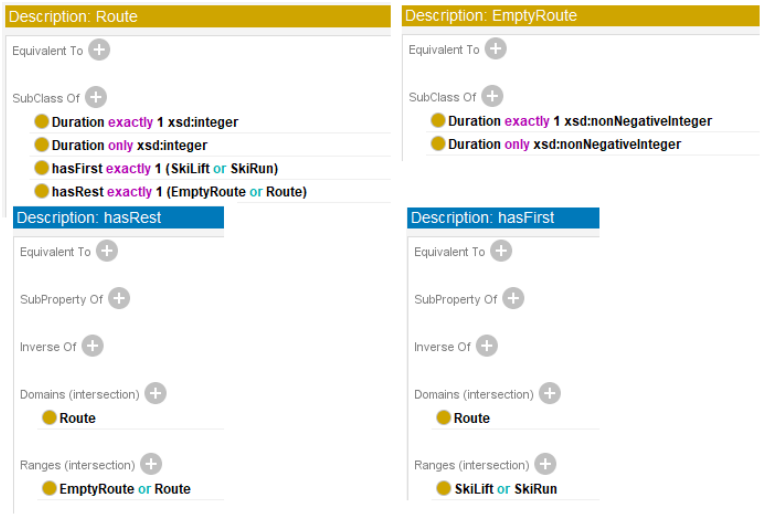

Figure 2: Caption

Afin de définir une route représentée par une séquence de SkiLift ou SkiRun, nous avons besoins de deux classes et de deux propriétés associés. Une route est composée d'un premier élément et d'un reste qui peut être une autre route ou une route vide indiquant la fin d'une séquence. Chaque route possède une certaine durée qui correspond à la somme des durées des SkiRuns et SkiLift qui la compose. Il nous est donc désormais possible de définir une route en indiquant son premier élément qui est un SkiRun ou un Skilift dont nous connaissons la durée et un reste qui renvoie sur une prochaine route possédant lui-même son propre premier élément. Nous faisons cela jusqu'à atteindre la route vide.

3. Inférer les propriétés de la route

Afin d'inférer automatiquement la durée, la difficulté d'une route et l'appartenance d'une Place/SkiLift/SkiRun/Restaurant à une route, nous avons d'abords définis deux nouvelles propriétés, hasMember et difficulty:

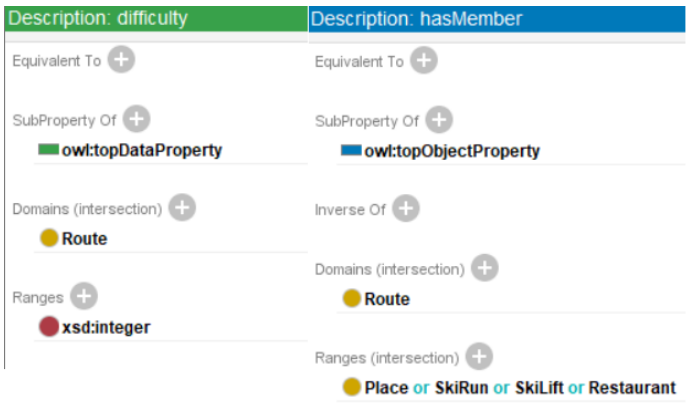

Figure 3: Caption

hasMember indique ce que le skieur peut trouver sur la route et la difficulté d'une route signale la piste la plus difficile se trouvant sur celle-ci. Cette difficulté est représentée par un entier, 0 pour une piste bleu, 1 pour une piste rouge et 2 pour une piste noir.

Grâce à ces nouvelles propriétés, il nous ai possible d'utiliser des règles SWRL pour inférer leurs valeurs. Voici donc les règles que nous avons définis:

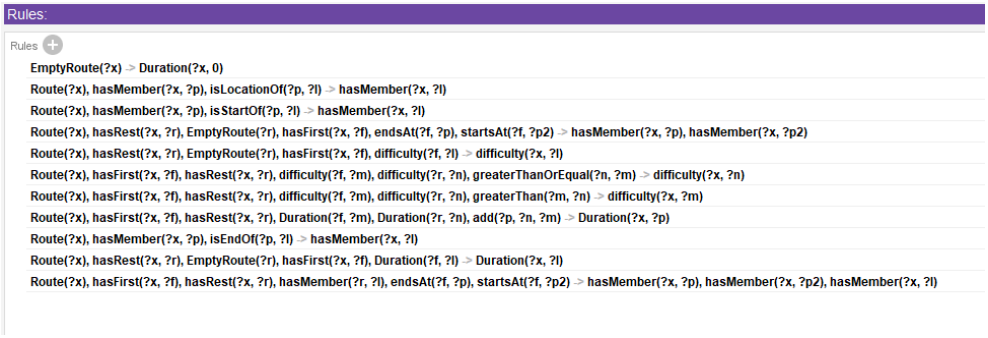

Figure 4: Caption

Voici un exemple de résultat obtenu sur une route constituée uniquement de SkiLift:

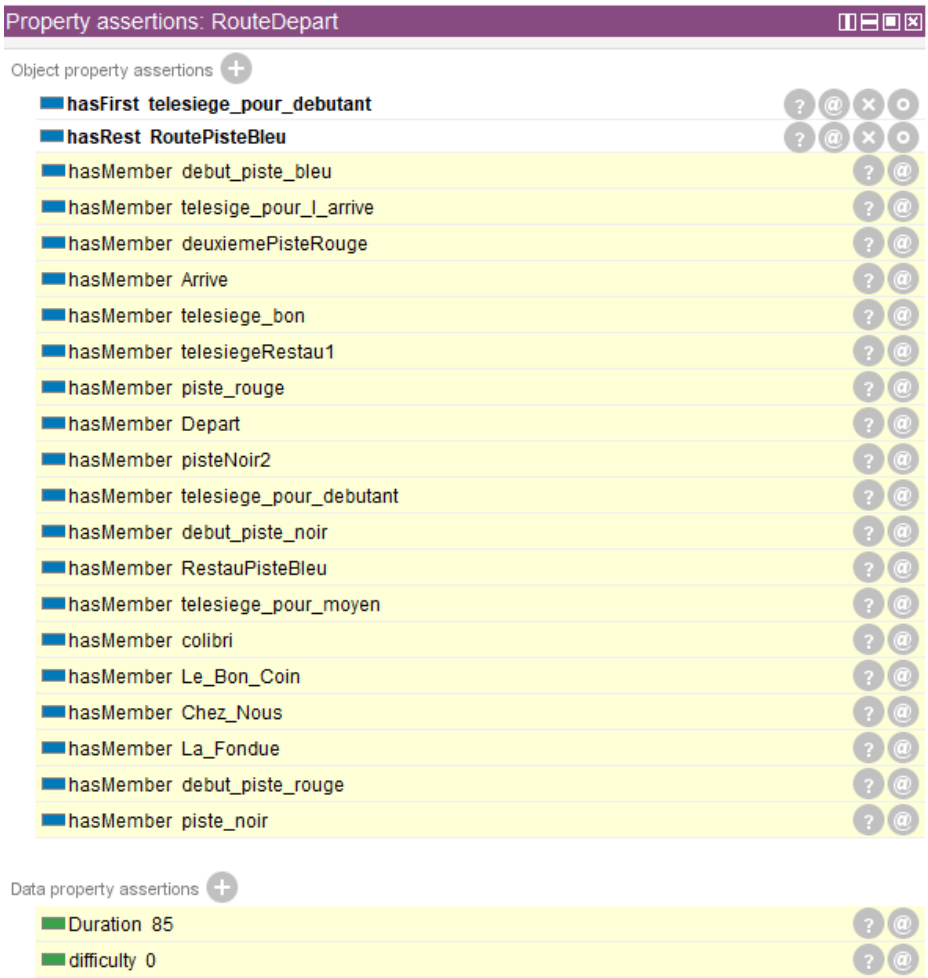

Figure 5: Caption

4. Transférer les inférences sur GrapheDB

Afin de pouvoir utiliser les données inférées par nos règles, il nous ai nécessaire de les ajouter dans le fichier turtle contenant notre ontologie de base via des insertions SPARQL à l'aide d'un script python. Voici un exemple de notre route de départ sous forme de triplet rdf:

```
###  http://www.unigeSki#RouteDepart
<http://www.unigeSki#RouteDepart> rdf:type owl:NamedIndividual ,
                                           <http://www.unigeSki#Route> ;
                                  <http://www.unigeSki#hasFirst> <http://www.unigeSki#telesiege_pour_debutant> ;
                                  <http://www.unigeSki#hasMember> <http://www.unigeSki#Arrive> ,
                                                                  <http://www.unigeSki#Chez_Nous> ,
                                                                  <http://www.unigeSki#Depart> ,
                                                                  <http://www.unigeSki#La_Fondue> ,
                                                                  <http://www.unigeSki#Le_Bon_Coin> ,
                                                                  <http://www.unigeSki#RestauPisteBleu> ,
                                                                  <http://www.unigeSki#colibri> ,
                                                                  <http://www.unigeSki#debut_piste_bleu> ,
                                                                  <http://www.unigeSki#debut_piste_noir> ,
                                                                  <http://www.unigeSki#debut_piste_rouge> ,
                                                                  <http://www.unigeSki#deuxiemePisteRouge> ,
                                                                  <http://www.unigeSki#pisteNoir2> ,
                                                                  <http://www.unigeSki#piste_noir> ,
                                                                  <http://www.unigeSki#piste_rouge> ,
                                                                  <http://www.unigeSki#telesiegeRestau1> ,
                                                                  <http://www.unigeSki#telesiege_bon> ,
                                                                  <http://www.unigeSki#telesiege_pour_debutant> ,
                                                                  <http://www.unigeSki#telesiege_pour_moyen> ,
                                                                  <http://www.unigeSki#telesige_pour_l_arrive> ;
                                  <http://www.unigeSki#hasRest> <http://www.unigeSki#RoutePisteBleu> ;
                                  <http://www.unigeSki#Duration> 85 ;
                                  <http://www.unigeSki#difficulty> 0 .
```

Figure 6: Caption

Après avoir fait cela, il nous ai donc possible d'importer notre ontologie sur GrapheDB avec l'entièreté des propriétés de nos routes.

5. Introduction de la notion de temps

Afin de définir les intervalles de temps ou un SkiLift ou un SkiRun est ouvert, il nous suffit de déclarer une nouvelle Classe TimeInterval possédant une date de début et une date de fin. Il suffit ensuite d'associer une propriété "open" entre un SkiLift/SkiRun et un interval de temps pour indiquer les horaires d'ouvertures de celle-ci.

Voici le modèle utiliser:

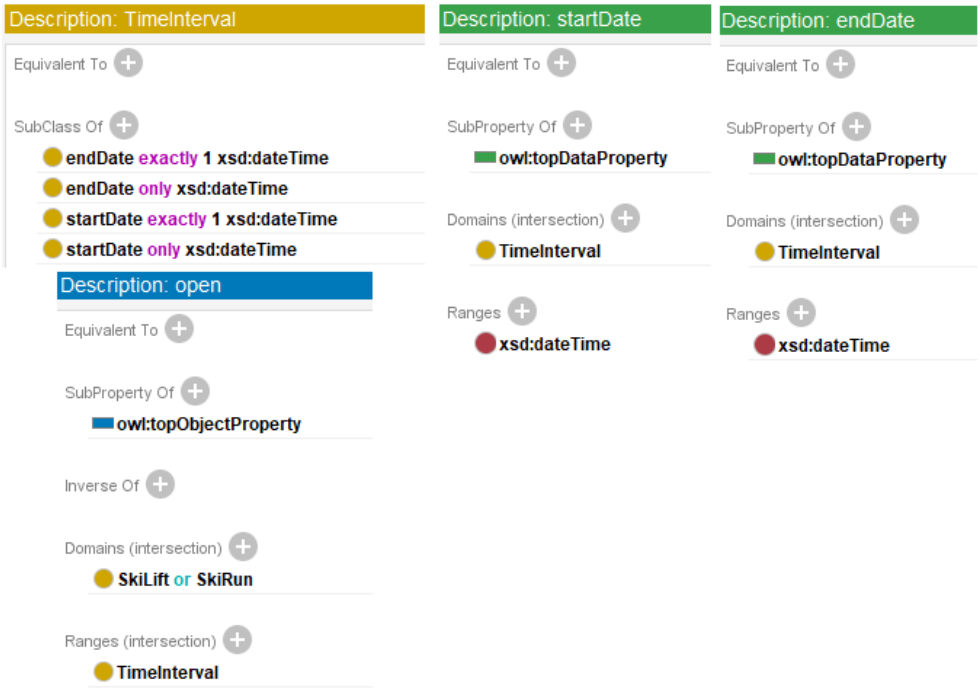

Figure 7: Caption