

TP3 : Ontological Hiking

In this project we had to create the OWL-2 ontology of a tourist area with hiking trails.

1 Queries

We have 8 queries to express with SWRL rules :

1. The places that have at least one restaurant
Rule : `Restaurant(?x), location(?x, ?y) -> query_1(?y)`
Query result : {A, D, I, J, K}
Explanation : We go through every restaurant, we check the location of each Restaurant via `location()` property, and every location is put in the `query_1` class.
2. The places from which one can reach a restaurant by following one trail
Rule : `connected(?x, ?y), query_1(?x) -> query_2(?y)`
Query result : {B, C, D, E, G, I, J, K}
Explanation : We go through every location that has a restaurant, by using the result from `query_1`, and check which locations are connected to the location in `query_1`, and every connected location is put in the `query_2` class.
3. The places from which one can reach a restaurant by following any number of trails
Rule : `connected(?x, ?y), query_2(?x) -> query_3(?x), query_3(?y)`
Rule : `query_3(?x), connected(?x, ?y) -> query_3(?y)`
Query result : {A, B, C, D, E, G, H, I, J, K}
Explanation : We first start by going through the locations that have a Restaurant at distance of 1 trail, then add to `query_3` class that same Location, as well as the locations it is connected to. Finally we do a recursion of every possible Location connected to a Location already in `query_3`.
4. The places that have a hotel and from which a beginner hiker can reach a restaurant by following any number of trails
Rule : `Blue(?x), locationEnd(?x, ?y), locationStart(?x, ?z) -> connectedByBlue(?y, ?z), connectedByBlue(?z, ?y)`
Rule : `Blue(?x), locationStart(?x, ?y), Hotel(?z), location(?z, ?y) -> BeginnerLocationsWithHotel(?y)`
Rule : `Blue(?x), locationEnd(?x, ?y), Hotel(?z), location(?z, ?y) -> BeginnerLocationsWithHotel(?y)`
Rule : `BeginnerLocationsWithHotel(?x), location(?z, ?y), Restaurant(?z), connectedByBlue(?x, ?y) -> query_4(?x)`
Rule : `query_4(?x), BeginnerLocationsWithHotel(?y), connectedByBlue(?x, ?y)`

-> query_4(?y)

Query result : {B, G}

Explanation : First we create the property connectedByBlue() that says 2 locations have a blue trail connecting them. Secondly, we look for every location that has a hotel, and that has a blue trail connected to them. Then, by starting from the locations that have hotels and are connected to blue trails, we check that at the location upon which the blue trail arrives, there is a Restaurant, if there is one, we had this original location to query_4 class. Finally we add a final rule that does the recursion.

5. The restaurants that are on a circuit made of at least two trails

Rule : Restaurant(?x), location(?x,?y), connected(?y,?z), connected(?y,?t) -> query_5(?x)

Query result : {restoA, restoD, restoI1, restoI2, restoJ, restoK}

Explanation : We go through every Restaurant and its location, and we check if that location is connected to 2 different locations, if it is, then the original location is added to query_5 result.

6. The restaurants from which one can reach another restaurant by following a path that only passes through places that have restaurants

Rule : connected(?x,?y), Restaurant(?z), location(?z,?x), Restaurant(?t), location(?t,?y) -> restaurantConnections(?x,?y), restaurantConnections(?y,?x)

Rule : Restaurant(?y), Restaurant(?x), location(?x,?t), location(?y,?z), restaurantConnections(?t,?z) -> query_6(?y), query_6(?x)

Query result : {restoD, restoI1, restoI2, restoJ, restoK}

Explanation : We Start by adding a new property, restaurantConnections() that is visible if 2 locations are connected, and both have restaurants. Then we simply check if 2 different restaurants that are at two different locations, are connected by this new property, if they are, we add them, both to query_6 result.

7. The hotels that can be reached from place p only by good hikers

Rule : Black(?x), locationStart(?x,?y), Hotel(?z), location(?z,?y) -> query_7(?y)

Rule : Black(?x), locationEnd(?x,?y), Hotel(?z), location(?z,?y) -> query_7(?y)

Query result : {H}

Explanation : The only black trail that has a Hotel is H, however, we cant do a rule with a variable in mind, so this answer isn't really the right one.

8. The hotels that can be reached from place p1 by following two different routes of length 2