

Instituto Tecnológico de Estudios Superiores de Monterrey



Programming Languages

Teacher: Benjamín Valdés Aguirre

“Tourist Path Algorithm in Prolog”

Adrián Torres Hernández

A01173530

ITESM Querétaro

Table of contents

Context of the problem -----	3
Solution -----	4
Results -----	6
Conclusions -----	6
Setup instructions -----	7
References -----	8

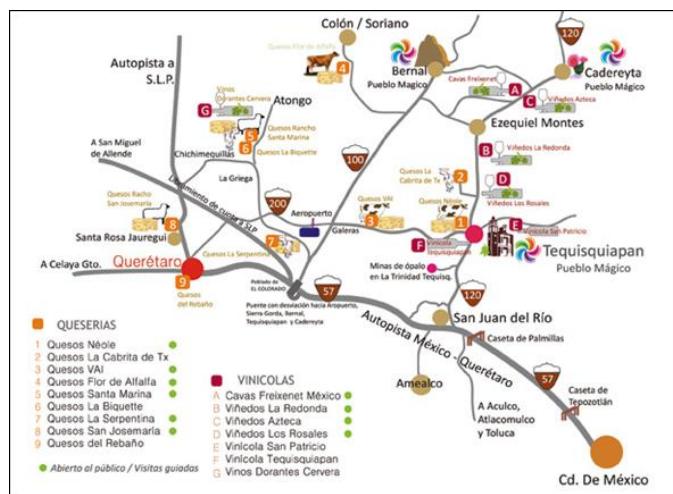
Context of the problem

In a census that was done in 2019, the state of Querétaro was found in 20th place in the number of tourist economic units around México. Furthermore, the lack of tourism has been a problem since 2015. Indeed, an article published in 2016 by “Códice Informativo” mentioned that many people when they go to Querétaro they just visit the center of the city. Querétaro has some other places such as Magical Towns, forests and mountains. Also Querétaro it is the place where I am currently doing my degree so I wanted to present and I was curious about discover some tourist places.

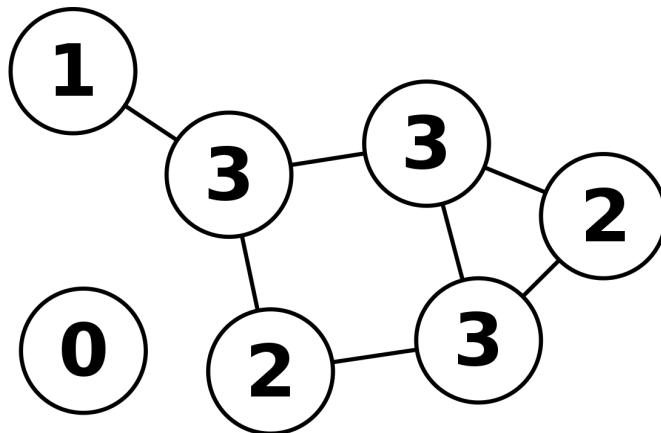


Solution

To aid this problem, I implemented a code that shows all the touristic place that you can visit that are near of the main destination. The code is considering that you can go in two directions on the road. I used the image 1.1 to simplify what places do I have to take into account for the code.



1.1



1.2

I decided to use prolog because as we can see in image 1.2 it is easy to implement and visualize graphs in prolog and I can modulate each touristic place as a node of a graph due to the use of unification and logic programming.

Implementation

First I define the knowledge base base on the map that was previously shown.

```
road(soriano, bernal).  
road(bernal, soriano).  
road(soriano, queretaro).  
road(queretaro, soriano).  
road(bernal, queretaro).  
road(queretaro, bernal).  
road(queretaro, jauregui).  
road(jauregui, queretaro).  
road(bernal, cadereyta).  
road(cadereyta, bernal).  
road(cadereyta, ezequiel).  
road(ezequiel, cadereyta).  
road(ezequiel, tequisquiapan).  
road(tequisquiapan, ezequiel).  
road(tequisquiapan, san_juan).  
road(san_juan, tequisquiapan).  
road(san_juan, amealco).  
road(amealco, san_juan).  
road(san_juan, mexico).  
road(mexico, san_juan).
```

The first part, its to create the predated travel that goes from the origin to destination. In order to return the list of places that you has visited we create the predicate path tha will return the result.

```
travel(Origin, Destination, R) :-  
    empty_stack(Empty_list),  
    stack(Origin, Empty_list, Visited_List),  
    path(Origin, Destination, Visited_List, R).
```

This would we the unification of the result “The base case” that its true when we reach the destination.

```
path(Destination, Destination, R, R).
```

And finally, this is the recursion of path that continue iterating over the roads and determine your possible visited the adjacent tourist places. if you havent visited the tourist place then it will be added to the visiting list.

```

path(Actual, Destination, Visited_List, R) :-  

    road(Actual, Next),  

    not(member_stack(Next, Visited_List)),  

    stack(Next, Visited_List, Updated_list),  

    path(Next, Destination, Updated_list, R), !.

```

Results

Input

I create the testTravel predicate that will call travel with some values and will throw the list of tourist places that you can visit before you reach the destination in each particular case depending on the origin and the destination. Taking into account you can go in two directions in some places.

```

testTravel(X,Y,Z):-  

    travel(bernal, mexico, X), % X = [mexico, san_juan, tequisquiapan, ezequiel, cadereyta, bernal],  

    travel(soriano, jauregui, Y), % Y = [jauregui, queretaro, bernal, soriano],  

    travel(mexico , ezequiel, Z). % Z = [ezequiel, tequisquiapan, san_juan, mexico]

```

Output

```

SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)  

File Edit Settings Run Debug Help  

Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)  

SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.  

Please run ?- license. for legal details.  

For online help and background, visit https://www.swi-prolog.org  

For built-in help, use ?- help(Topic). or ?- apropos(Word).  

?- consult('C:\\\\Users\\\\Adrian Torres\\\\Programming-Languages-Project\\\\Tourist_path.pl').  

true.  

?- testTravel(X,Y,Z).  

X = [mexico, san_juan, tequisquiapan, ezequiel, cadereyta, bernal].  

Y = [jauregui, queretaro, bernal, soriano].  

Z = [ezequiel, tequisquiapan, san_juan, mexico].  

?-

```

As we can see we obtain the corresponding path of tourists places that lead us to the destination staring in a origin.

Conclusions

With prolog the use of backtracking was easy for this particular problem. One of the things that I like about prolog the most was the use of logic programming with is very

useful when you want to implement graphs and nodes for instance the path algorithm that was develop in this project. Finally, the last thing that was fundamental and I like about prolog its that you have to define a base case and you define the knowledge base. These characteristics make me want to choose prolog insted of other languages.

Setup instructions

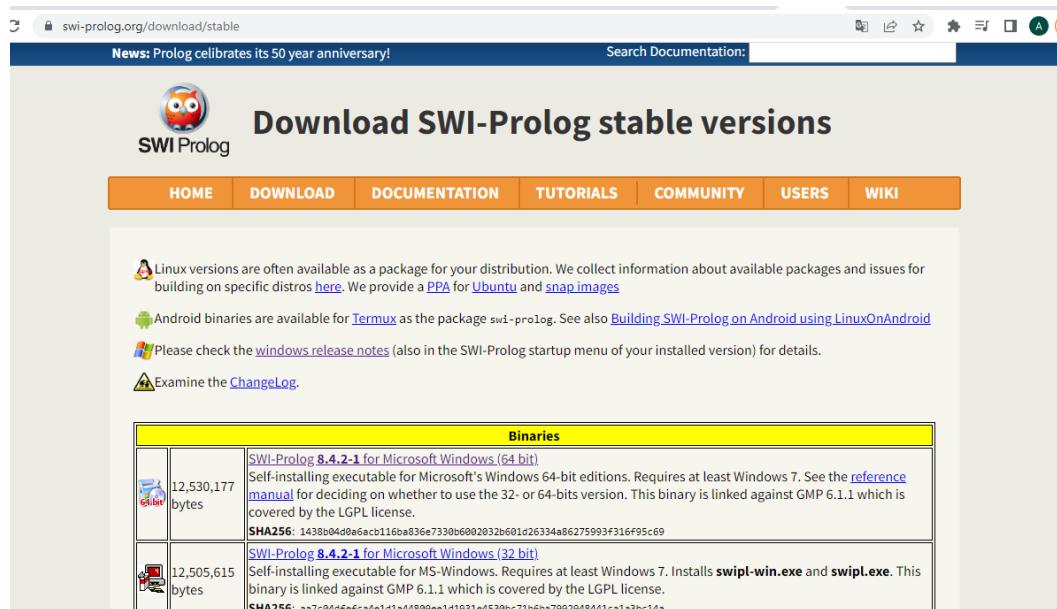
In order to run the following project please do the following instructions.

1. Download the GitHub repository

Enter and download the repository as a ZIP file and then unzip it

2. Install SWI-Prolog

Enter <https://www.swi-prolog.org/download/stable> and download the version that your computer uses.



4.- Run the code

Access the folder where the .pl file is located. and run these commands:

`consult('folder_where_the_file_it_is_located').`

And to see a test run

`testTravel(X,Y,Z).`

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('C:\\\\Users\\\\Adrian Torres\\\\Programming-Languages-Project\\\\Tourist_path.pl').
true.

?- testTravel(X,Y,Z).
X = [mexico, san_juan, tequisquiapan, ezequiel, cadereyta, bernal],
Y = [jauregui, queretaro, bernal, soriano],
Z = [ezequiel, tequisquiapan, san_juan, mexico].
?- 
```

References

- Constantini, A. (2020, 5 octubre). *Censos Económicos 2019: Resultados de Turismo*. Brain Analytics & Innovation. Recuperado 6 de junio de 2022, de <https://brainanalytics.mx/coyuntura/censos-economicos-2019-resultados-de-turismo/>
- Baruah, A. (2020, 8 noviembre). *Prolog - Detecting if path exists between two nodes in a Graph*. Youtube. Recuperado 6 de junio de 2022, de <https://www.youtube.com/watch?v=ym0raWm1WHE>
- *Los retos turísticos de Querétaro*. (2016, 3 noviembre). Códice Informativo. Recuperado 6 de junio de 2022, de <https://codiceinformativo.com/2016/11/los-retos-turisticos-de-queretaro/>
- P., A. G. (2018, 1 diciembre). *La ruta turística enológica en Querétaro y Baja California, México: Un enfoque estratégico*. Scielo. Recuperado 6 de junio de 2022, de https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-235X2018000200122