

CSC-325
Artificial Intelligence
Dr. Adam Wyner

12 March 2021

Lab 4

Last sign-off date is 19 March at 17:00

The practical has four parts, all of which must be completed to complete the lab.

Your work, solution, and code should be your own. If you discuss how to address the problem, use words and narratives.

To get signed off, you should:

- Show your code and solutions to a lecturer/demonstrator;
- Explain your code and solutions to a lecturers/ demonstrators.

Tip: refer to Learn Prolog Now! for further information, particularly sections on lists.

Part I

% SCENARIO:

% We are given the following knowledge base of travel information:

% KB:

byCar(auckland,hamilton).
byCar(hamilton,raglan).
byCar(valmont,saarbruecken).
byCar(valmont,metz).

byTrain(metz,frankfurt).
byTrain(saarbruecken,frankfurt).
byTrain(metz,paris).
byTrain(saarbruecken,paris).

byPlane(frankfurt,bangkok).
byPlane(frankfurt,singapore).
byPlane(paris,losAngeles).
byPlane(bangkok,auckland).
byPlane(losAngeles,auckland).

% PROBLEM:

% Write a predicate travel/2 which determines whether it is possible to travel
% from one place to another by 'chaining together' car, train, and plane journeys.
% For example, your program should answer 'yes' to the query travel(valmont,raglan).
% Take care that your answers are valid.

% QUERIES:

% Can you get from Valmont to Raglan?
%?- travel(valmont,raglan).
%Yes
% Where can you get from Valmont to? Give the first 9 answers.
% Are there any problems getting further results? Describe the problem.

% SOLUTION:

% Put your solution in Prolog here.

Part II

% SCENARIO:

% By using travel/2 from previous part to query the database, you can find out that
% it is possible to go from Valmont to Raglan. What you would then really want to
% know is what is the route to get from Valmont to Raglan.

% KB:

byCar(auckland,hamilton).
byCar(hamilton,raglan).
byCar(valmont,saarbruecken).
byCar(valmont,metz).

byTrain(metz,frankfurt).
byTrain(saarbruecken,frankfurt).
byTrain(metz,paris).
byTrain(saarbruecken,paris).

byPlane(frankfurt,bangkok).
byPlane(frankfurt,singapore).
byPlane(paris,losAngeles).
byPlane(bangkok,auckland).
byPlane(losAngeles,auckland).

% PROBLEM:

% Write a predicate travel/3 which tells you how to go (the path, not the means) from one
place to another.
% Your answers to the queries below should paths, not simply answering 'yes' or 'no'.

% QUERIES:

% Can you travel from Valmont to Paris via Metz?

% What are the routes from Valmont to Los Angeles?

% SOLUTION:

% Put your solution in Prolog here.

Part III

% SCENARIO:

% In the KB, we have direct trains from A to B. But, it is also
% reasonable to assume that one also has a direct train from B to A.

% KB:

```
directTrain(saarbruecken,dudweiler).  
directTrain(forbach,saarbruecken).  
directTrain(freyming,forbach).  
directTrain(stAvold,freyming).  
directTrain(fahlquemont,stAvold).  
directTrain(metz,fahlquemont).  
directTrain(nancy,metz).
```

% PROBLEM:

% Write a predicate to represent that where there is a train connection A to B,
% there is also a train connection from B to A.
% Write a predicate route/3 which gives you a list of towns that are visited by
% taking the train from one town to another. Note that you will need to work with
% list structures and list procedures.
% Hint: you do not want redundant elements in your list, nor do you want 'digressions' not on
% the route.

% QUERIES:

% Here are some queries (expressed in Natural Language) that your solution should
% answer correctly.

% What is the route from Forbach to Metz?

% What cities are connected via the route through Saarbruecken,
% Forbach, Freyming, St Avold, Fahlquemont, and Metz?

% SOLUTION:

% Put your solution in Prolog here.

Part IV

Describe all steps of backward chaining, where \neg is a symbol of negation as failure and starting hypothesis is C

Knowledge base:

Rules: Rule 1. $A \wedge B \Rightarrow C$

Rule 2. $\neg X \Rightarrow A$

Rule 3. $D \wedge F \Rightarrow X$

MO: D, F, B

% SOLUTION:

% Put your solution here.