Jaiwin Shah
2019130058
TE Comps
Batch C

Aim:

To design and implement an expert system, incorporating the match algorithm and the rule language.

1. It should provide a fact base updating function.
2. It should provide a function that check the rules’ LHS and return which rules were matched.
3. It should support firing RHS according to matches.

Theory:

Expert System:

A naive implementation of an expert system might check each rule against known facts in a knowledge base, firing that rule, if necessary, then moving on to the next rule (and looping back to the first rule when finished). For even moderate sized rules and facts knowledge-bases, this naive approach performs far too slowly. The Rete algorithm provides the basis for a more efficient implementation. A Rete-based expert system builds a network of nodes, where each node (except the root) corresponds to a pattern occurring in the left- hand-side (the condition part) of a rule. The path from the root node to a leaf node defines a complete rule left-hand-side. Each node has a memory of facts which satisfy that pattern. This structure is essentially a generalized trie. As new facts are asserted or modified, they propagate along the network, causing nodes to be annotated when that fact matches that pattern. When a fact or combination of facts causes all of the patterns for a given rule to be satisfied, a leaf node is reached and the corresponding rule is triggered.

Problem Statement

Read the below passage carefully and answer the questions:

Five cities all got more rain than usual this year. The five cities are: Last Stand, Mile City, New Town, Olliopolis, and Polberg. The cities are located in five different areas of the country: the mountains, the forest, the coast, the desert, and in a valley. The rainfall amounts were: 12 inches, 27 inches, 32 inches, 44 inches, and 65 inches.

- The city in the desert got the least rain; the city in the forest got the most rain.
- New Town is in the mountains.
- Last Stand got more rain than Olliopolis.
- Mile City got more rain than Polberg, but less rain than New Town.
- Olliopolis got 44 inches of rain.
- The city in the mountains got 32 inches of rain; the city on the coast got 27 inches of rain.

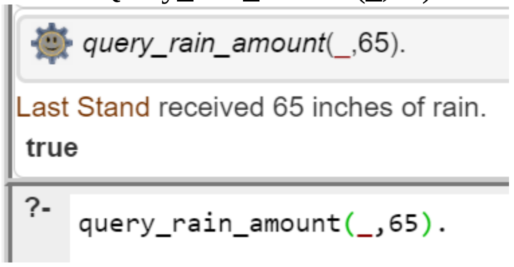
1. Which city got the most rain?
2. How much rain did Mile City get?
3. Which city is in the desert?
4. Where is Olliopolis located?

Code:

```
city(C) :-
% there are 5 cities
length(C, 5),
% city names
member(h('Last Stand', _, _), C),
member(h('Mile City', _, _), C),
member(h('New Town', _, _), C),
member(h('Olliopolis', _, _), C),
member(h('Polberg', _, _), C),
% city areas
member(h(_, mountains, _), C),
member(h(_, forest, _), C),
member(h(_, coast, _), C),
member(h(_, desert, _), C),
member(h(_, valley, _), C),
% rainfall amounts
member(h(_, _, 12), C),
member(h(_, _, 27), C),
member(h(_, _, 32), C),
member(h(_, _, 44), C),
member(h(_, _, 65), C),
% Hints
% The city in the desert got the least rain;
% the city in the forest got the most rain.
member(h(_, desert, 12), C),
member(h(_, forest, 65), C),
% New Town is in the mountains.
member(h('New Town', mountains, _), C),
% Last Stand got more rain than Olliopolis.
member(h('Last Stand', _, A), C),
member(h('Olliopolis', _, B), C),
A > B,
% Mile City got more rain than Polberg, but less rain than New Town.
member(h('Mile City', _, D), C),
member(h('Polberg', _, E), C),
D > E,
member(h('New Town', _, F), C),
F > D,
% Olliopolis got 44 inches of rain.
member(h('Olliopolis', _, 44), C),
% The city in the mountains got 32 inches of rain; the
% city on the coast got 27 inches of rain.
member(h(_, mountains, 32), C),
member(h(_, coast, 27), C).
query_rain_amount(City_Name, Rainfall_Amount) :-
city(C),
member(h(City_Name, _, Rainfall_Amount), C),
write(City_Name), write(" received "),
write(Rainfall_Amount), write(" inches of rain."),
nl.
query_city_region(City_Name, Region) :-
city(C),
member(h(City_Name, Region, _), C),
write(City_Name), write(" is located in the "),
write(Region), nl.
```

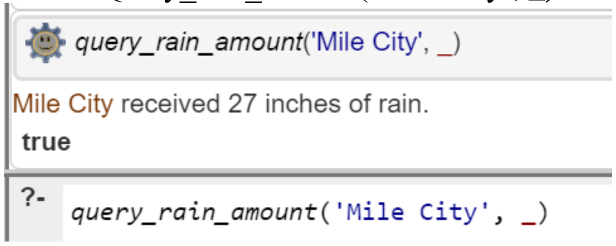
Output:

Query_rain_amount(_,65)



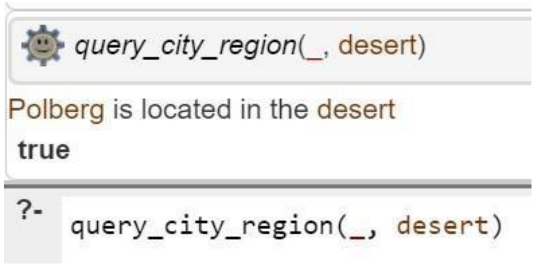
```
query_rain_amount(_,65).
Last Stand received 65 inches of rain.
true

?- query_rain_amount(_,65).
```

- Query_rain_amount('Mile City', _)



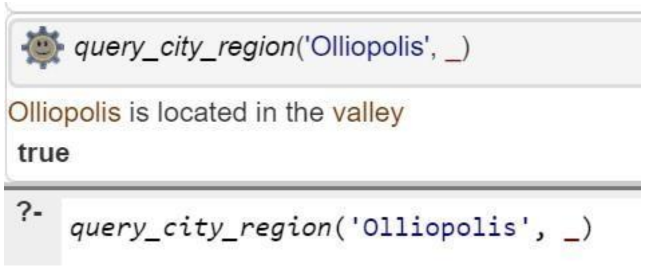
```
query_rain_amount('Mile City', _)
Mile City received 27 inches of rain.
true

?- query_rain_amount('Mile City', _)
```

- Query_city_region(_, desert)



```
query_city_region(_, desert)
Polberg is located in the desert
true
?- query_city_region(_, desert)
```

- Query_city_region(‘Olliopolis’, _)



```
query_city_region('Olliopolis', _)
Olliopolis is located in the valley
true
?- query_city_region('Olliopolis', _)
```

Conclusion:

I learnt about PROLOG, and how to use it to solve logical problems using a declarative programming style. I learned about Expert systems. It is an interactive computer-based decision-making system which uses both facts and heuristics to solve complex decision-making problems. The given question has data about the amount of rainfall in a city and in which region the city lies in. In the above code, first all the names of cities and regions were stored which were mentioned in the problem statement and then the facts are stored. Then using the query in the above code, we can find the name of city from the amount of rainfall and vice-versa, also we can find the region from the city-name and vice-versa. Prolog makes it easier to find answers to these queries once the facts have been stored and using these facts it finds the solution.