Pranav Nair
2019130042
TE Comps Batch – C
29th November, 2021

Experiment-7

Aim: To design and implement an expert system, incorporating the match algorithm and the rule language and to answer the queries, in the given problem statement.

Problem statements:

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) :-

    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),
    member(h('New Town',_,F),C),
    D>E,
    D<F,

    % 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_amt(CityName, RainfallAmt):-
    city(C),
    member(h(CityName,_,RainfallAmt),C),
    write(CityName),write(" has received "),write(RainfallAmt),write(" inches"),nl.

query_city_region(CityName, Region):-
    city(C),
    member(h(CityName,Region,_),C),
    write(CityName),write(" is in "),write(Region),write(" region"),nl.
```

Output:

1.
```
query_rain_amt(_,65)
Last Stand has received 65 inches
true
```

```
?- query_rain_amt(_,65)
query_rain_amt('Mile City',_)
Mile City has received 27 inches
true
```

2.
```
?- query_rain_amt('Mile City',_)
query_city_region(_,desert)
Polberg is in desert region
true
```

3.
```
?- query_city_region(_,desert)
query_city_region('Olliopolis',_)
Olliopolis is in valley region
true
```

4.
```
?- query_city_region('Olliopolis',_)
```

Conclusion:
In this experiment, the given problem statement 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. Prolog is very well suited for implementing expert systems due to several reasons: Prolog itself can be regarded as a simple inference engine or theorem prover that derives conclusions from known rules. Very simple expert systems can be implemented by relying on Prolog's built-in search and backtracking mechanisms.

Swish link: https://swish.swi-prolog.org/p/Pranav%20Nair.pl
GitHub: https://github.com/pranav567/AI-ML-Lab/tree/master/experiment-7