CSB 310: Artificial Intelligence

Lab Assignment 2

Submitted By:

Name: **Umang Kumar**

Roll No: **201210051**

Branch: **CSE**

Semester: **5th Sem**

Submitted To: ***Dr. Chandra Prakash***

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

Department of Computer Science and Engineering

2022

PART A: Exposition Problems

1. Classics Example of Logical Programming

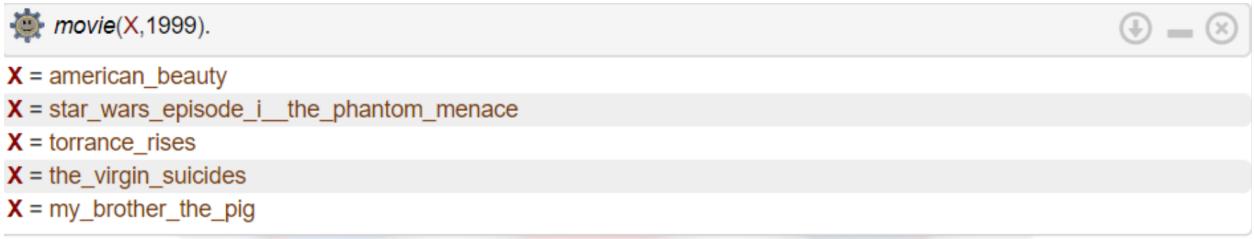
- **Movie database** : provides a couple of thousands of facts about movies for you to query.

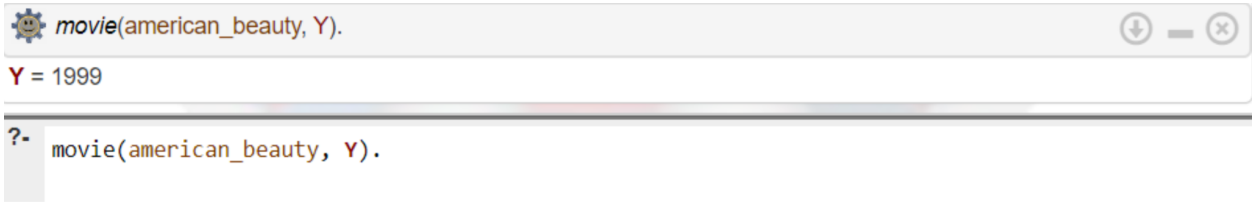
Code:

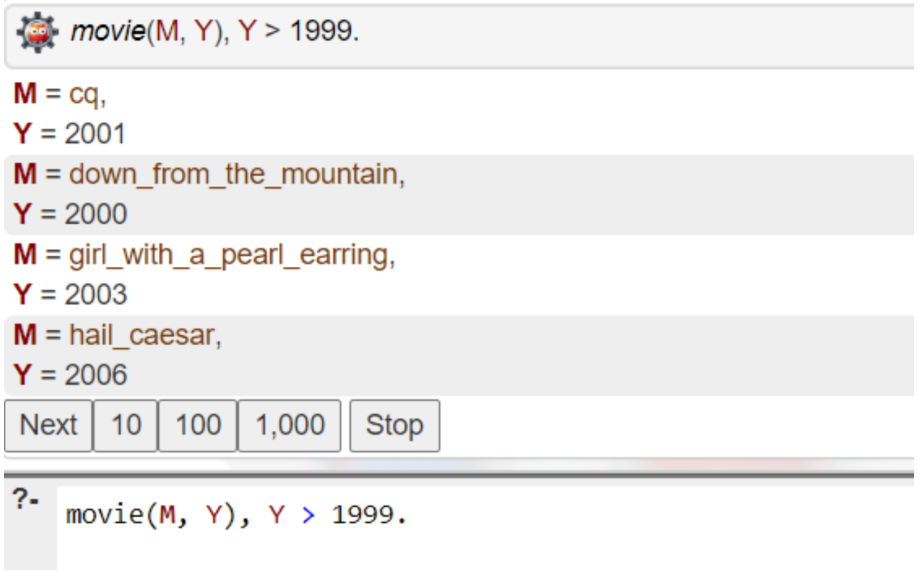
examples | movies

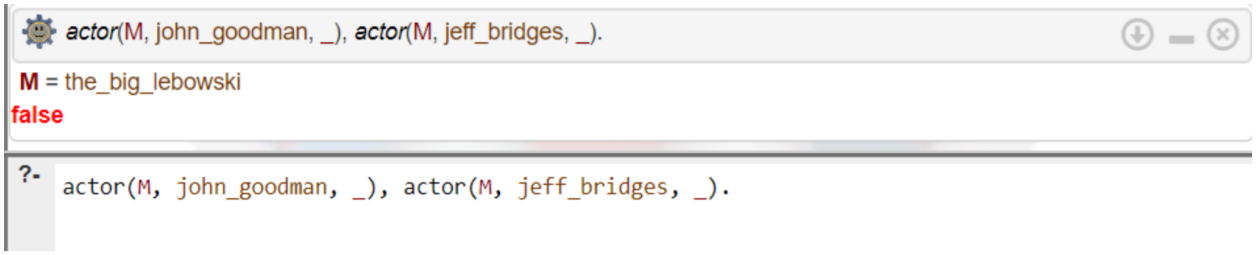
```
56
57 movie(american_beauty, 1999).
58 director(american_beauty, sam_mendes).
59 actor(american_beauty, kevin_spacey, lester_burnham).
60 actress(american_beauty, annette_bening, carolyn_burnham).
61 actress(american_beauty, thora_birch, jane_burnham).
62 actor(american_beauty, wes_bentley, ricky_fitts).
63 actress(american_beauty, mena_suvari, angela_hayes).
64 actor(american_beauty, chris_cooper, col_frank_fitts_usmc).
65 actor(american_beauty, peter_gallagher, buddy_kane).
66 actress(american_beauty, allison_janney, barbara_fitts).
67 actor(american_beauty, scott_bakula, jim_olmeyer).
68 actor(american_beauty, sam_robards, jim_berkley).
69 actor(american_beauty, barry_del_sherman, brad_dupree).
70 actress(american_beauty, ara_celi, sale_house_woman_1).
71 actor(american_beauty, john_cho, sale_house_man_1).
72 actor(american_beauty, fort_atkinson, sale_house_man_2).
73 actress(american_beauty, sue_casey, sale_house_woman_2).
74 actor(american_beauty, kent_faulcon, sale_house_man_3).
75 actress(american_beauty, brenda_wehle, sale_house_woman_4).
76 actress(american_beauty, lisa_cloud, sale_house_woman_5).
77 actress(american_beauty, alison_faulk, spartanette_1).
78 actress(american_beauty, krista_goodsitt, spartanette_2).
79 actress(american_beauty, lily_houtkin, spartanette_3).
80 actress(american_beauty, carolina_lancaster, spartanette_4).
81 actress(american_beauty, romana_leah, spartanette_5).
82 actress(american_beauty, chekeshka_van_putten, spartanette_6).
83 actress(american_beauty, emily_zachary, spartanette_7).
84 actress(american_beauty, nancy_anderson, spartanette_8).
85 actress(american_beauty, reshma_gajjar, spartanette_9).
86 actress(american_beauty, stephanie_rizzo, spartanette_10).
```

Queries:


movie(X,1999).
X = american_beauty
X = star_wars_episode_i__the_phantom_menace
X = torrance_rises
X = the_virgin_suicides
X = my_brother_the_pig
?-
movie(X,1999).


movie(american_beauty, Y).
Y = 1999
?-
movie(american_beauty, Y).


movie(M, Y), Y > 1999.
M = cq,
Y = 2001
M = down_from_the_mountain,
Y = 2000
M = girl_with_a_pearl_earring,
Y = 2003
M = hail_caesar,
Y = 2006
Next
10
100
1,000
Stop
?-
movie(M, Y), Y > 1999.

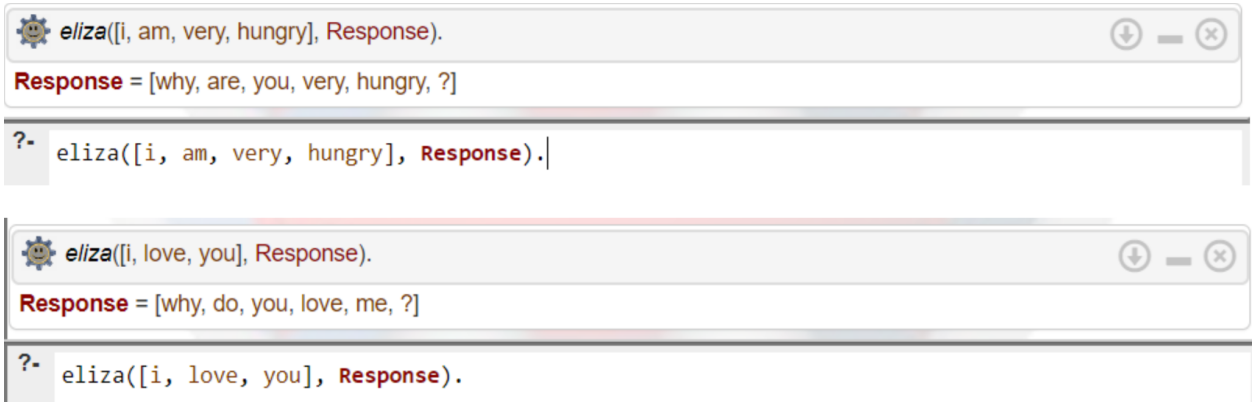


- **Eliza:** implements the classical shrink

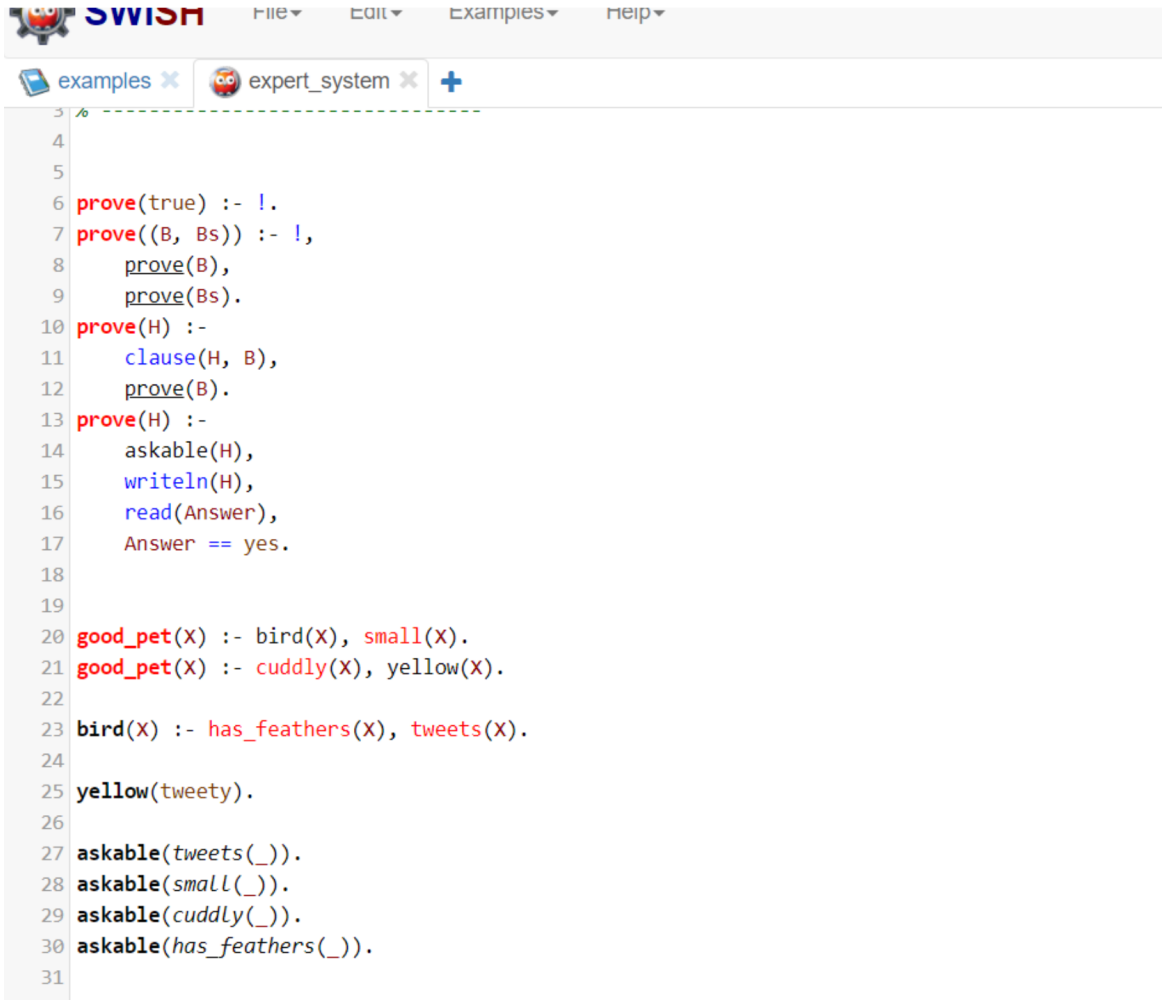
examples eliza

```
1  %%  eliza(+Stimuli, -Response) is det.
2  %   @param  Stimuli is a list of atoms (words).
3  %   @author Richard A. O'Keefe (The Craft of Prolog)
4
5  eliza(Stimuli, Response) :-
6      template(InternalStimuli, InternalResponse),
7      match(InternalStimuli, Stimuli),
8      match(InternalResponse, Response),
9      !.
10
11 template([s([i,am]),s(X)], [s([why,are,you]),s(X),w('?')]).
12 template([w(i),s(X),w(you)], [s([why,do,you]),s(X),w(me),w('?')]).
13
14
15 match([],[]).
16 match([Item|Items],[Word|Words]) :-
17     match(Item, Items, Word, Words).
18
19 match(w(Word), Items, Word, Words) :-
20     match(Items, Words).
21 match(s([Word|Seg]), Items, Word, Words0) :-
22     append(Seg, Words1, Words0),
23     match(Items, Words1).
24
25
26 /** <examples>
27
28 ?- eliza([i, am, very, hungry], Response).
29 ?- eliza([i, love, you], Response).
30
31 */
```

Queries:



- **Expert system:** illustrates simple meta-interpretation of rules and asking for missing knowledge.



Queries:

PART B : Conceptual Questions

Run the sample example given.

a) discuss the sample example given of Sam's likes and dislikes in food.

b) Identify the facts in the sample example.

Ans:- facts in the given below program are:

- Curry, Dahl, Tandoori, Kurma are Indian food.
- Dahl, Tandoori, Kurma are mild.
- Chow_mein, chop_suey, sweet_and_sour are Chinese food.
- Pizza and spaghetti are Italian food.

c) Identify the rules in the taken example

Ans: - rules in the given below program are:

- Sam likes the Food if it is Indian and Mild.
- Sam likes the Food if it is Chinese.
- Sam likes the Food if it is Italian.
- Sam likes the Chips.
- Mohit likes the food if Sam also likes that.

d) Run what does Sam likes.

e) Sem likes curry.

f) Add a new rule that Mohit eat whatever Sem likes.

g) Tracing the execution of a Prolog query allows you to see all of the goals that are

executed as part of the query, in sequence, along with whether or not they succeed. Show the steps occur in the above program.

Code:

```
likes(sam,Food) :-
    indian(Food),
    mild(Food).
likes(sam,Food) :-
    chinese(Food).
likes(sam,Food) :-
    italian(Food).
likes(sam,chips).

likes(mohit,Food) :- likes(sam,Food).


indian(curry).
indian(dahl).
indian(tandoori).
indian(kurma).

mild(dahl).
mild(tandoori).
mild(kurma).

chinese(chow_mein).
chinese(chop_suey).
chinese(sweet_and_sour).

italian(pizza).
italian(spaghetti).
```

Queries:

```
?-
% e:/btech/semester 5/csb310/lab01/que2 compiled 0.00 sec, -2 clauses
?- likes(sam,dahl).
true .

?- likes(sam,X).
X = dahl ;

?- likes(mohit,X).
X = dahl ;

?- likes(sam,curry).
false.

?-
```

3. Consider the following Knowledge Base:

The humidity is high or the sky is cloudy. If the sky is cloudy, then it will rain.
If the humidity is high, then it is hot.
It is not hot today.
Query : Will it rain today ?

Code:

```
humidity(high).
sky(cloudy, today).

rain(X) :- nothot(X), sky(cloudy, X).
hot(X) :- humidity(X).
nothot(today).
```

Queries:

4. Write a prolog program to calculate the sum of two numbers.

Code:

```
sum(X, Y, Z) :- Z is X + Y.
```

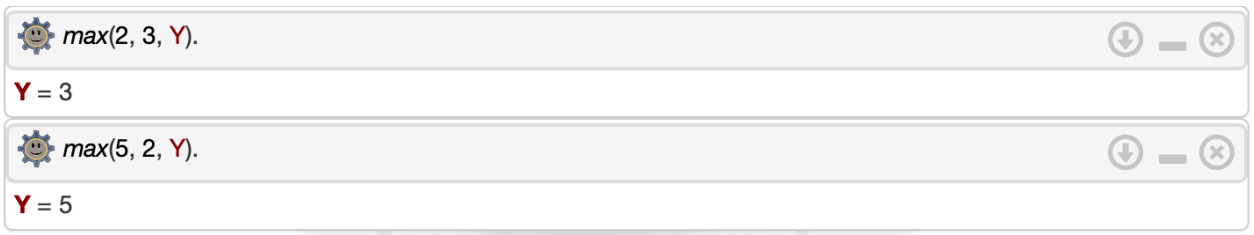
Queries:

5. Write a Prolog program to implement max(X, Y, Max) so that Max is the maximum of two numbers X and Y.

Code:

```
max(X,Y,R):-
 X >= Y ->
  R is X
 ;
  R is Y.
```

Queries:



6. Write a program in PROLOG to implement factorial (N, Fac) where Fac represents the factorial of a number N.

Code:

```
factorial(0, 1).
factorial(N, R) :- N > 0, N1 is N-1, factorial(N1, S), R is S*N.
```

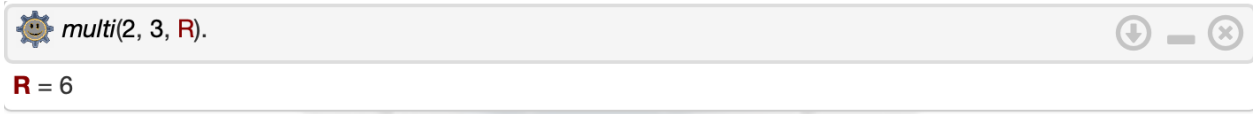
Queries:

7. Write a Prolog program to implement multi (N1, N2, Res) : where N1 and N2 denotes the numbers to be multiplied and Res represents the result.

Code:

```
multi(A, B, R) :- R is A*B.
```

Queries:



8. Consider a cyclic directed graph [edge (p, q), edge (q, r), edge (q, r), edge (q, s), edge (s,t)] where edge (A,B) is a predicate indicating directed edge in a graph from a node A to a node B. Write a program to check whether there is a route from one node to another node.

Code:

```
edge(p,q).
edge(q,r).
edge(r,q).
edge(q,s).
edge(s,t).

route(X,Y):- edge(X,Y).
route(X,Y):- edge(X,Z), route(Z,Y).
```

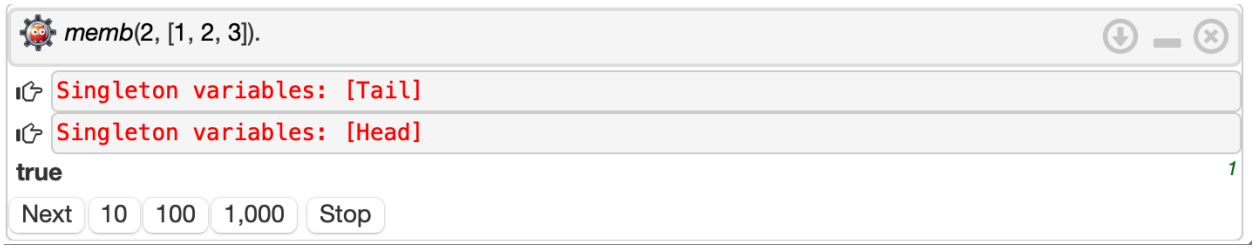
Queries:

9. Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.

Code:

```
memb(X,[X|Tail]).
memb(X,[Head|Tail]):- memb(X,Tail).
```

Queries:

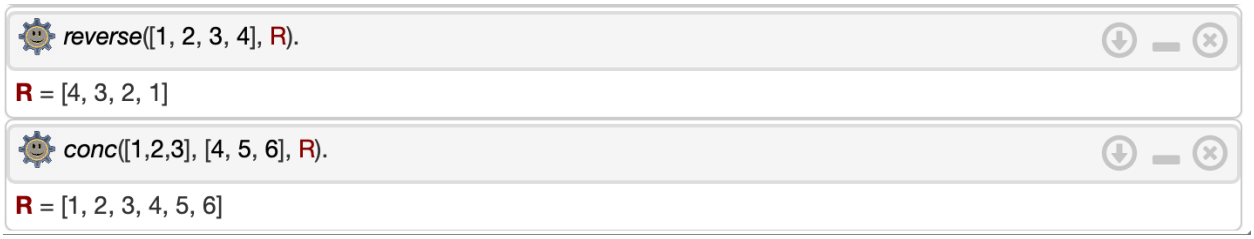


10. Write a Prolog program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.

Code:

```
conc( [],Y,Y).
conc([Head|Tail],Y,[Head|Rest]) :-   conc(Tail,Y,Rest).
```

Queries:



11. Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.

Code:

```
append([],L,L).
append([X|L1],L2,[X|L3]):- append(L1,L2,L3).

reverse([],[]).
reverse([H|T],R):- reverse(T,L1),append(L1,[H],R).
```

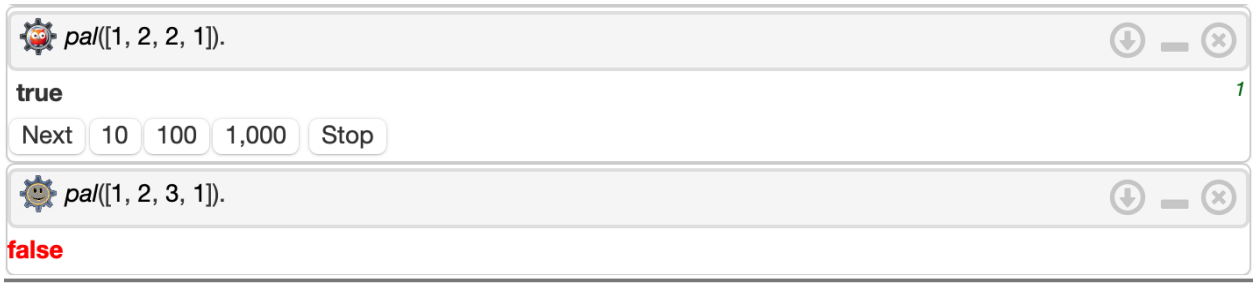
Queries:

12. Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.

Code:

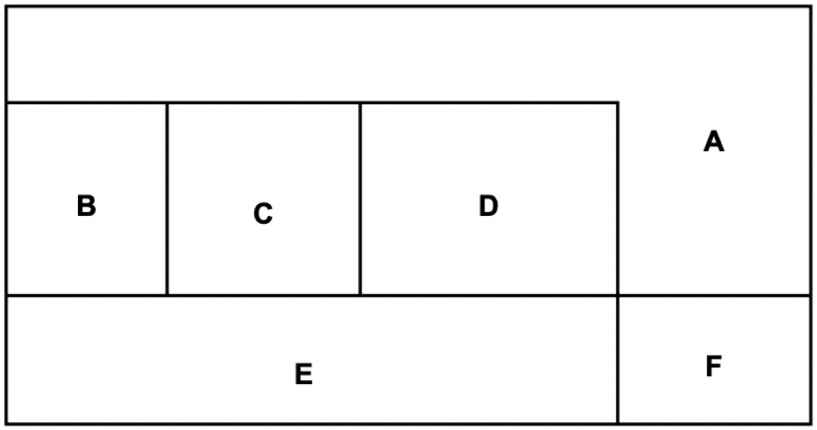
```
app([],L,L).
app([X|L1],L2,[X|L3]):- app(L1,L2,L3).
pal([]).
pal([_]).
pal(Plist) :- app([H|T],[H],Plist),pal(T).
```

Queries:



PART C : Exploratory Problem : Map Colorings

There is a famous problem in mathematics for coloring adjacent planar regions. Like cartographic maps, it is required that, whatever colors are used, no two adjacent regions may not have the same color. Two regions are considered adjacent provided they share some boundary line segment. Consider the following map.



Develop a Prolog program that can compute all possible colorings (Given colors to color with) are Red ,Blue, Green and Yellow.

Code:

```
%% que 13.

adjacent(a, b). adjacent(a, c).
adjacent(a, d). adjacent(a, f).
adjacent(b, e). adjacent(c, e).
adjacent(d, e). adjacent(f,e).
adjacent(b, c). adjacent(c, d).
adjacent(b, a). adjacent(c, a).
adjacent(d, a). adjacent(f, a).
adjacent(e, b). adjacent(e, c).
adjacent(e, d). adjacent(e, f).
adjacent(c, b). adjacent(d, c).

color(a, red, s1).
color(b, blue, s1).
color(c, green, s1).
color(d, blue, s1).
color(f, blue, s1).
color(e, red, s1).
color(a, red, s2).
color(b, blue, s2).
color(c, red, s2).
color(d, blue, s2).
color(f, blue, s2).
color(e, red, s2).

conflict(Coloring) :-
    adjacent(X, Y) , color(X, Color, Coloring), color(Y, Color, Coloring).

conflict(X, Y, Coloring) :-
    adjacent(X, Y) , color(X, Color, Coloring) , color(Y, Color, Coloring).
```

Queries:

In this question we can consider each box as a node of a graph and all adjacent boxes are interlinked with edges then we can try multiple coloring scheme on it. If two adjacent edges have same color then conflict will occur else not.

Observations:

- Learned some advanced concepts of prolog language like list, if- else.
- Map Coloring Can be visualized as a Graph Coloring problem.
- Learn about facts, rules and queries in more depth.