Declan ONUNKWO 04/06/2023

CSC344 – Programming Languages

Assignment: **Prolog programming assignment 1 – Various computations**

LEARNING ABSTRACT

The project is aimed to introduce us to Prolog programming language. I was able to gain a basic understanding on the prolog function and syntax, as well as work with examples that feature code that introduces one to prolog. This project helped me to develop a deeper appreciation for the power and elegance of prolog, as well as to improve my problem-solving abilities and programming skills.

Task 1: Colors KB

Colors KB Code

```
%-----------------------------------------------------------------------------
% primary(P) :: P is a primary color

primary(blue).
primary(red).
primary(yellow).

%-----------------------------------------------------------------------------
% secondary(S) :: S is a secondary color

secondary(green).
secondary(orange).
secondary(purple).

%-----------------------------------------------------------------------------
% color(C) :: C is a color

color(C) :- primary(C).
color(C) :- secondary(C).
```

Colors KB Interaction

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

?- primary(blue).
ERROR: Unknown procedure: primary/1 (DWIM could not correct goal)
?- consult('~/prolog/colors.pro').
true.

?- primary(blue).
true.

?- primary(red).
true.

?- primary(green).
false.

?- secondary(green).
true.

?- secondary(purple).
true.

?- secondary(yellow).
false.

?- color(blue).
true
Unknown action: d (h for help)
Action? .

?- color(purple).
true.

?- primary(P).
P = blue ;
P = red ;
P = yellow.
```

```
?- secondary(S).
S = green ;
S = orange ;
S = purple.

?- color(C).
C = blue ;
C = red ;
C = yellow ;
C = green ;
C = orange ;
C = purple.

?- listing(primary).
primary(blue).
primary(red).
primary(yellow).

true.

?- listing(secondary).
secondary(green).
secondary(orange).
secondary(purple).

true.

?- listing(color).
color(C) :-
    primary(C).
color(C) :-
    secondary(C).

true.

?-
```

Task 2: Food KB

Food KB Code

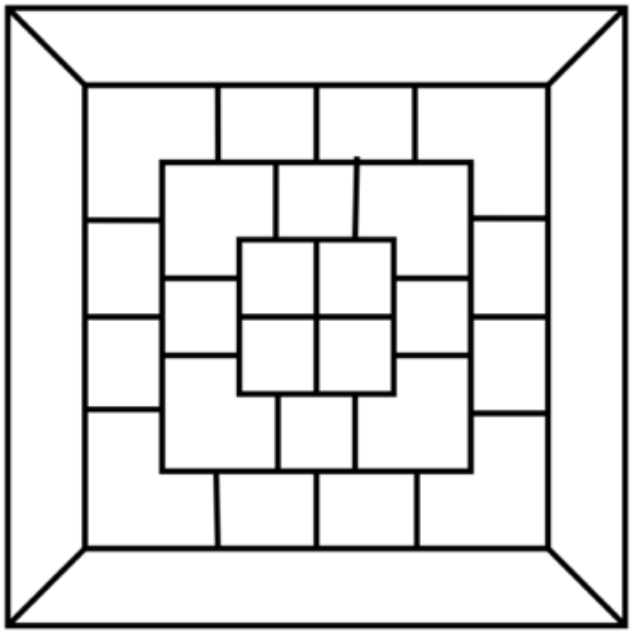
```
%----------------------------------------------------------------
% fruit(F) :: F is fruit

fruit(grapefruit).
fruit(avocado).
fruit(date).

%----------------------------------------------------------------------------
% vegetable(V) :: V is a vegetable

vegetable(asperagus).
vegetable(broccoli).
vegetable(carrot).

%----------------------------------------------------------------------------
% food(FF) :: FF is a food

food(FF) :- fruit(FF).
food(FF) :- vegetable(FF).

```

Food KB Interaction

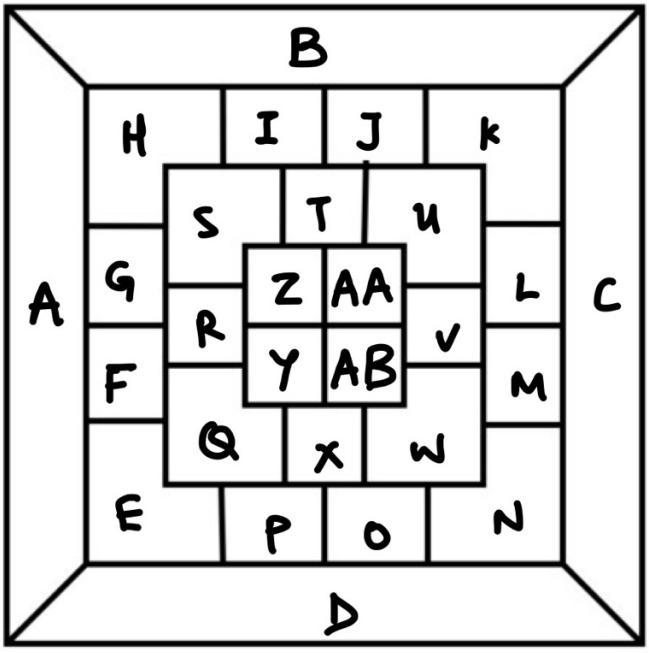
```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- consult('~/prolog/foods.pro').
true.

?- fruit(grapefruit).
true.

?- fruit(avocado).
true.

?- fruit(asperagus).
false.

?- vegetable(asperagus).
true.

?- vegetable(broccoli).
true.

?- vegetable(peas).
false.

?- food(date).
true .

?- food(carrot).
true.

?- fruit(F).
F = grapefruit ;
F = avocado ;
F = date.

?- vegetable(V).
V = asperagus ;
V = broccoli ;
V = carrot.

?- food(FF).
FF = grapefruit ;
FF = avocado ;
FF = date ;
FF = asperagus ;
FF = broccoli ;
FF = carrot.

?- listing(fruit).
fruit(grapefruit).
fruit(avocado).
fruit(date).

true.

?- listing(vegetable).
vegetable(asperagus).
vegetable(broccoli).
vegetable(carrot).

true.

?- listing(food).
food(FF) :-
    fruit(FF).
food(FF) :-
    vegetable(FF).

true.

?-
```

Task 3: Map Coloring

The Given Map



The Labeled Map



Code for Coloring the Map

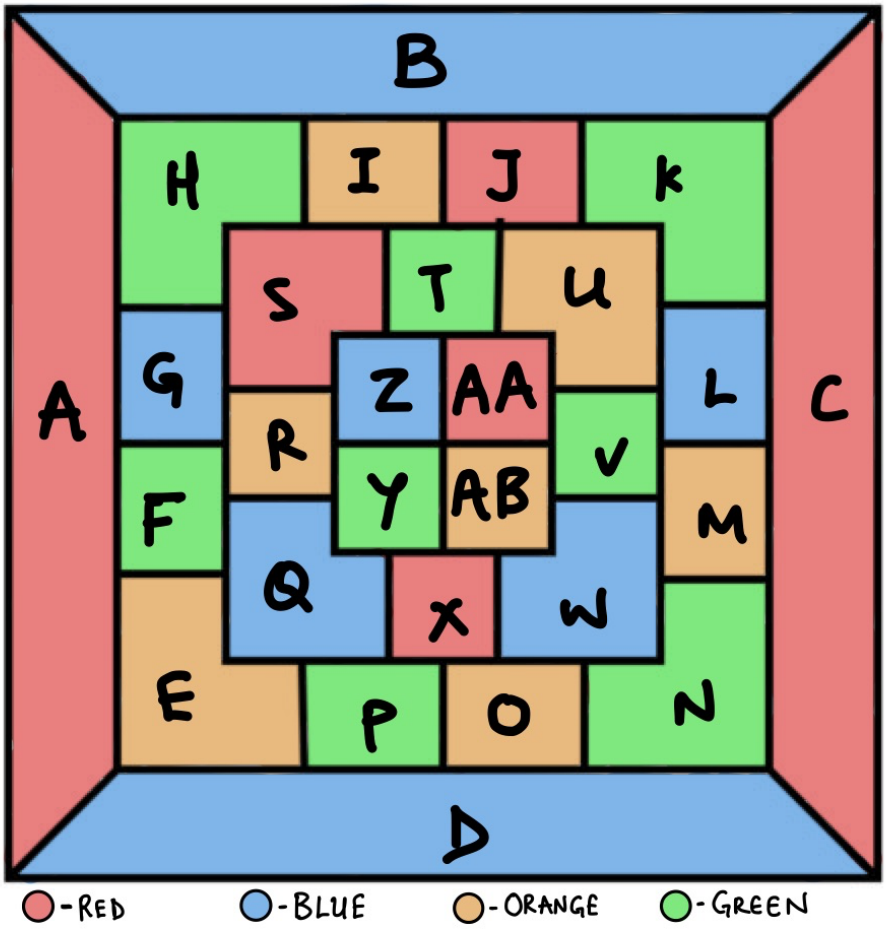
```
% ----------------------------------------------------------------------
% File: map_coloring.pro
% Line: Program to find a 4 color map rendering for South American countries.
% More: The colors used will be red, blue, green, orange.
% More: The standard abbrieviation are used to stand for the countries.

% ----------------------------------------------------------------------
% different(X,Y) :: X is not equal to Y

different(red,blue).
different(red,green).
different(red,orange).
different(green,blue).
different(green,orange).
different(green,red).
different(blue,green).
different(blue,orange).
different(blue,red).
different(orange,blue).
different(orange,green).
different(orange,red).

% ----------------------------------------------------------------------
% coloring(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB) ::
% The individual shapes that form the square are represented by the letters
% A-AB (which is A-Z then AA,AB).

coloring(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB) :-
different(A,B),
different(A,D),
different(A,H),
different(A,G),
different(A,F),
different(A,E),
different(C,B),
different(C,D),
different(C,K),
different(C,L),
different(C,M),
different(C,N),
different(B,H),
different(B,I),
different(B,J),
different(B,K),
different(D,E),
different(D,P),
different(D,O),
different(D,N),
different(H,I),
different(H,G),
different(H,S),
different(E,F),
different(E,Q),
```

```
54 different(E,P),
55 different(K,J),
56 different(K,U),
57 different(K,L),
58 different(N,O),
59 different(N,W),
60 different(N,M),
61 different(I,S),
62 different(I,T),
63 different(I,J),
64 different(J,T),
65 different(J,U),
66 different(G,S),
67 different(G,R),
68 different(F,R),
69 different(F,Q),
70 different(F,G),
71 different(P,Q),
72 different(P,X),
73 different(P,O),
74 different(O,X),
75 different(O,W),
76 different(M,W),
77 different(M,V),
78 different(M,L),
79 different(L,V),
80 different(L,U),
81 different(T,S),
82 different(T,Z),
83 different(T,AA),
84 different(T,U),
85 different(R,S),
86 different(R,Z),
87 different(R,Y),
88 different(R,Q),
89 different(X,Q),
90 different(X,Y),
91 different(X,AB),
92 different(X,W),
93 different(V,W),
94 different(V,AB),
95 different(V,AA),
96 different(V,U),
97 different(U,AA),
98 different(W,AB),
99 different(Y,Q),
100 different(Z,Y),
101 different(Z,AA),
102 different(Z,AB),
103 different(Z,S),
104 different(AB,Y),
105 different(AB,AA),
106 different(AA,Y).
```

Map Coloring Interaction

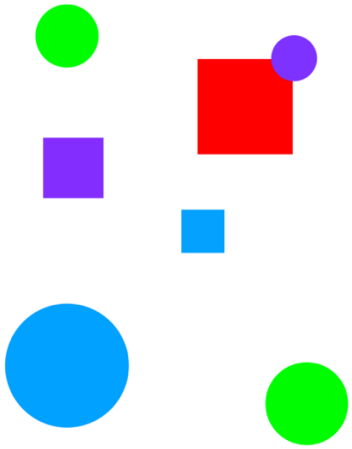
```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- consult('~/prolog/map_coloring.pro').
true.

?- coloring(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB).
A = C, C = J, J = S, S = X, X = AA, AA = red,
B = D, D = G, G = L, L = Q, Q = W, W = Z, Z = blue,
E = I, I = M, M = O, O = R, R = U, U = AB, AB = orange,
F = H, H = K, K = N, N = P, P = T, T = V, V = Y, Y = green
```

The Colored Map



Task 4: Floating Shapes World KB

Floating Shapes World Image



Floating Shapes World KB Code

```
%-------------------------------------------------------------------------------------------------
% File: shapes_world_1.pro
% Line: Loosely represented 2-D shapes world (simple take on SHRDLU)
%-------------------------------------------------------------------------------------------------
% Facts ...
%-------------------------------------------------------------------------------------------------
% square(N,side(L),color(C) :: N is the name of a square with side L and color C

square(sera,side(7),color(purple)).
square(sera,side(5),color(blue)).
square(sarah,side(11),color(red)).

%-------------------------------------------------------------------------------------------------
% circle(N, radius(R),color(C)) :: N is the name of a circle with radius R and color C

circle(carla,radius(4),color(green)).
circle(cora,radius(7),color(blue)).
circle(connie,radius(3),color(purple)).
circle(claire,radius(5),color(green)).

%-------------------------------------------------------------------------------------------------
% Rules ...
%-------------------------------------------------------------------------------------------------
% circles :: list the names of all of the circles

circles :- circle(Name,_,_), write(Name),nl,fail.
circles.

%-------------------------------------------------------------------------------------------------
% squares :: list the names of all of the squares

squares :- square(Name,_,_), write(Name),nl,fail.
squares.

%-------------------------------------------------------------------------------------------------
% shapes :: list the names of all of the shapes

shapes :- circles,squares.

%-------------------------------------------------------------------------------------------------
% blue(Name) :: Name is a blue shape

blue(Name) :- square(Name,_,color(blue)).
blue(Name) :- circle(Name,_,color(blue)).

%-------------------------------------------------------------------------------------------------
% large(Name) :: Name is a large shape

large(Name) :- area(Name,A), A >= 100.

%-------------------------------------------------------------------------------------------------
% small(Name) :: Name is a small shape

small(Name) :- area(Name,A), A < 100.

%-------------------------------------------------------------------------------------------------
% area(Name,A) :: A is the area of the shape with a Name

area(Name,A) :- circle(Name,radius(R),_), A is 3.14 * R * R.
area(Name,A) :- square(Name,side(S),_), A is S * S.
```

Floating Shapes World KB Interaction

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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('~/prolog/shapes_world_1.pro').
true.

?- listing(squares).
squares :-
    square(Name, _, _),
    write(Name),
    nl,
    fail.
squares.

true.

?- squares.
sera
sera
sarah
true.

?- listing(circles).
circles :-
    circle(Name, _, _),
    write(Name),
    nl,
    fail.
circles.

true.

?- circles.
carla
cora
connie
claire
true.

?- listing(shapes).
shapes :-
    circles,
    squares.

true.

?- shapes.
carla
cora
connie
claire
sera
sera
sarah
true.

?- blue(Shape).
Shape = sera ;
Shape = cora.

?- large(Name),write(Name),nl,fail.
cora
sarah
false.

?- small(Name),write(Name),nl,fail.
carla
connie
claire
sera
sera
false.

?- area(cora,A).
A = 153.86 .

?- area(carla,A).
A = 50.24 .

?-
```