

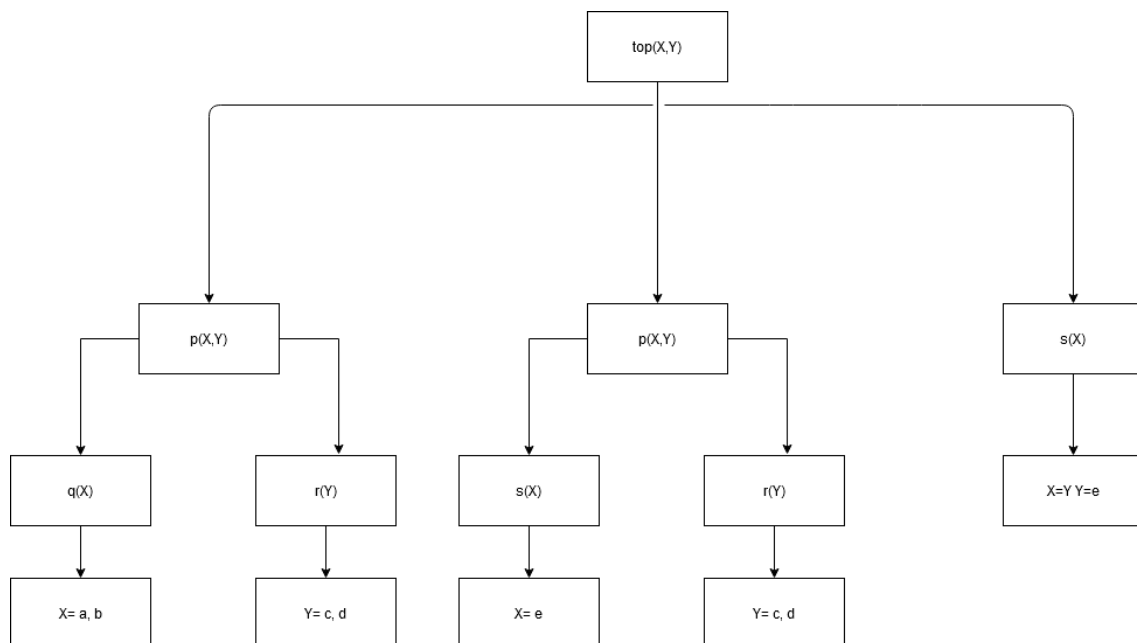
Prácticas de Prolog – Jesús Boán Mascareñas

Práctica 1

Ejercicio 1

```
?- top(X,Y) .  
X = a,  
Y = c ;  
X = a,  
Y = d ;  
X = b,  
Y = c ;  
X = b,  
Y = d ;  
X = e,  
Y = c ;  
X = e,  
Y = d ;  
X = Y, Y = e .
```

El árbol resultante es:



Ejercicio 2

p("Mareas Vivas",1998).

p("Pratos combinados", 1995).

p("Padre Casares",2008).

p("Serramoura",2014).

p("Matalobos",2009).

p("Libro de familia",2005).

p("As leis de Celavella",2003).

p("Fariña",2018).

p("Vivir sin permiso",2018).

p("Néboa",2020).

a("Emma Suarez").

a("Isabel Naveira").

a("Luis Zahera").

a("Luis Tosar").

a("Ernesto Chao").

a("Mabel Rivera").

a("María Castro").

a("Antonio Durán").

a("Pedro Alonso").

a("Alfonso Agra").

a("Uxía Blanco").

a("Sabela Arán").

a("Luis Iglesia").

a("Mónica Camaño").

a("María Mera").

a("Tacho González").

a("José Coronado").

a("Manuel San Martín").

a("Mayka Braña").

a("Lucía Regueiro").

a("Manuel Cortés").

a("Eva Fernández").

a("Javier Rey").

a("Tristán Ulloa").

a("Antonio Durán").

a("Antonio Mourellos").

a("Álex González").

a("Claudia Traisac").

actua("Mareas Vivas", "Luis Zahera").

actua("Mareas Vivas", "Luis Tosar").

actua("Pratos Combinados", "Ernesto Chao").

actua("Pratos Combinados", "Mabel Ribeira").

actua("Pratos Combinados", "María Castro").

actua("Padre Casares", "Antonio Durán").

actua("Padre Casares", "Pedro Alonso").

actua("Padre Casares", "Tacho González").

actua("Serramoura", "Alfonso Agra").

actua("Serramoura", "Uxía Blanco").

actua("Serramoura", "Alfonso Agra").

actua("Matalobos", "Luis Iglesia").

actua("Matalobos", "Mónica Camaño").

actua("Matalobos", "Maria Mera").

actua("Vivir sin permiso", "Álex González").

actua("Vivir sin permiso", "ClaudiaTraisac").

actua("Vivir sin permiso", "José Coronado").

actua("Fariña", "Javier Rey").

actua("Fariña", "Tristán Ulloa").

actua("Fariña", "Antonio Durán").

actua("As leis de Celavella", "Antonio Mourelos").

actua("As leis de Celavella", "Eva Fernández").

actua("As leis de Celavella", "Manuel Cortés").

actua("Libro de familia", "Lucía Regueiro").

actua("Libro de familia", "Mayka Braña").

actua("Libro de familia", "Manuel San Martín").

actua("Neboa", "Emma Suarez").

actua("Neboa", "Isabel Naveira").

estrenada_despues(P, A):- p(P,A1),A1>A.

participan(A1, A2):- actua(P,A1),actua(P,A2),A1\==A2.

estrenada_antes(P, A):-p(P,A1), A1<A.

mismo_año(P1, P2):- p(P1,A), p(P2,A),P1\==P2,! . %se establece un corte en esta regla para que no devuelva las mismas dos series cambiadas de orden.%

mas_de_una_serie(A):-actua(P1,A),actua(P2,A),P1\==P2,!.

a)

p("Mareas Vivas",A)

b)

p(P,2010)

c)

estrenada_antes(P, 2010)

d)

estrenada_despues(P, 2010)

e)

mismo_año(P1,P2)

f)

mas_de_una_serie(A)

g)

participan("Luis Tosar", "Willy Toledo")

3.

a)

arista(1,2).

```

arista(1,3).
arista(1,4).
arista(2,5).
arista(2,3).
arista(3,5).
arista(4,5).
arista(3,4).
arista(X,Y):- arista(Y,X).

```

b)

El problema de arista(5,1). es que no hay una conexión directa entre el vértice 1 y el 5, hay que pasar por alguno de los otros vértices. Lo que pasa con esto es que prolog entra en un bucle infinito

d)

No hay diferencia de resultado entre las dos reglas de los apartados. Sólo que en el d) es excluyente, mientras que en el c) no.

e)

```

camino(X,Y,Z):- encuentra_camino(X,Y,[X],W),reverse(W,Z).
encuentra_camino(A,B,C,[B|C]):-conexion(A,B).
encuentra_camino(A,B,VISITADO,CAM):- conexion(A,C), C\==B,
not(member(C,VISITADO)),encuentra_camino(C,B,[C|VISITADO],CAM).

```

Práctica 2

1.

a)

final(e3).

inicial(e1).

trans(e1,a,e1).

trans(e1,a,e2).

trans(e1,b,e1).

trans(e2,b,e3).

trans(e3,b,e4).

trans(e2,lambda,e4).

trans(e3,lambda,e1).

b)

acepta(L) :- aceptacadena(e1,L).

aceptacadena(E,[]) :- final(E).

aceptacadena(E,[X|L]) :- trans(E,X,E1), aceptacadena(E1,L).

aceptacadena(E,L) :- trans(E,lambda,E1), aceptacadena(E1,L).

c)

?- acepta([X,Y,Z])

X=Y,Y=a,Z=b; X=Z,Y=a,Z=b;

d)

aceptacadenalg(L,N) :- length(L,N), acepta(L).

e)

aceptacadena_menor(L,N):- between(0,N,M), length(L,M),acepta(L).

2.

1)

Un estado se representa con todos los lugares en los que están los objetos importantes del problema: donde está el mono en la sala (puerta, medio(debajo de los platanos) o ventana), donde está el mono (en el suelo o encima de la silla), donde está la silla (puerta, medio(debajo de los platanos) o ventana) y si el mono tiene o no los plátanos.

Inicial: estado(puerta,en_suelo,ventana,sin_platanos)

Final: estado(X,X,X,con_platanos)

2)

movimiento(estado(medio,en_silla,medio,sin_platanos),
coge_platanos,
estado(medio,en_silla,medio,con_platanos)).

movimiento(estado(P,en_suelo,P,H),
sube_silla,
estado(P,en_silla,P,H)).

movimiento(estado(P1,en_suelo,P1,H),
empuja(P1,P2),
estado(P2,en_suelo,P2,H)).

movimiento(estado(P1,en_suelo,B,H),
mover(P1,P2),
estado(P2,en_suelo,B,H)).

3)

solucion(estado(_,_,_,con_platanos)).

solucion(Estado1) :-

movimiento(Estado1,_,Estado2),
solucion(Estado2).

4)

Si se cambia el orden de operadores, el programa no funciona bien ya que realiza las operaciones en el orden incorrecto y no se llega a una buena solución.