

Programación Declarativa

15 de Diciembre de 2005

TEORÍA Y PROBLEMAS

Nombre:.....

Apellidos:.....

DNI:

Grupo:

EJERCICIO 1. Unificación. (1 Punto)

Implementar el predicado *lista_arboles/2*, que se cumple cuando el primer argumento es una lista de términos y el segundo es una lista que contiene todos aquellos elementos de la primera que son árboles binarios.

La representación que se deberá utilizar para un árbol binario será: *arbol(Valor,HIZq,HDer)* donde HIZq y HDer son árboles binarios o el árbol vacío que se representará como *nil*.

EJERCICIO 2. Matrices. (1 Punto)

Programe el predicado *mayores/3* que unifica cuando su primer argumento es un número, el segundo es una matriz bidimensional representada como lista de listas y el tercero es una lista con todos los elementos mayores que el primer argumento, y que estén contenidos en la matriz.

EJERCICIO 3. Árbol de resolución. (1 Punto)

Considerando que en nuestra base de conocimiento *Prolog* tenemos las siguientes cláusulas:

```
p(1,[]).
p(2,[C|R]):- q(C),r(R).

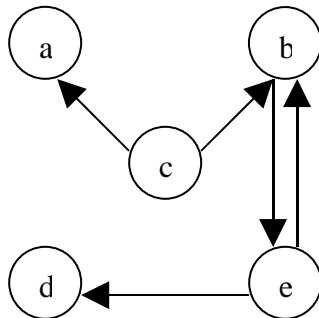
q( a(2) ):- !.
q( a(3) ):- r([]).

r([]).
r([C|R]):- !, R=[1,2,3], s(C).

s(a):-!.
s(b).
```

1. Dibuje el árbol de resolución para el objetivo: $p(X,Y)$.
2. Explícite las soluciones que *Prolog* encuentra para el objetivo propuesto.

EJERCICIO 4. Grafos. (2 Puntos)



1. Elija la representación adecuada del grafo para la resolución de este ejercicio.
2. Programe el predicado *camino* (elija usted la aridad que desee) que sea verdadero cuando exista un camino entre dos nodos. El predicado debe evitar entrar en ciclos.
3. Programe el predicado *todos* (elija usted la aridad que desee), con el que se pueda obtener una lista con todos los nodos a los que se puede llegar desde uno dado.
4. Escriba la consulta que se debería realizar en *Prolog* para obtener la lista de los nodos a los que se puede llegar desde el nodo *c*

Ejercicio 5. Haskell. (1 Punto)

1. Programe la función *divisores* con un único argumento y que devuelva una lista de enteros con los divisores del primer argumento.
2. Utilizando la función anterior *divisores*, programe la función *primo* con un único argumento entero que sea verdadera cuando el argumento sea un número primo (positivo).
3. ¿Cómo podría utilizar la función que ha programado para obtener una lista con los primos entre 1 y 100?