

PROGRAMACIÓN DECLARATIVA

CLASE PRÁCTICA 1

Tema: *Sintaxis de listas, comprensión y lectura de programas declarativos y definición del programa familia.*

1. Represente las siguientes listas utilizando las dos notaciones vistas en la conferencia:
 - a) La lista que tiene dos elementos y uno de ellos es el 2.
 - b) La lista que tiene al menos dos elementos y uno de ellos es el 2.
 - c) La lista cuyo primer elemento es la lista vacía y tiene un resto.
 - d) La lista que tiene un primer elemento y el resto es una lista con exactamente dos elementos.
 - e) La lista que tiene un primer elemento y el resto contiene un elemento que es una lista con exactamente dos elementos.
2. Escribir los programas Prolog vistos en conferencia comentándolos a partir de su lectura declarativa y analícelos paso a paso.
 - a) `concatenar(L1, L2, R)`: triunfa si R es el resultado de concatenar las listas L1 y L2.
 - b) `isList(L)`: triunfa si L es una lista.
 - c) `member(X, L)`: triunfa si X es un elemento de la lista L.
 - d) `divide(L, I, P)`: triunfa si L es la concatenación alternada de las listas I y P.
 - e) `push(X, L, R)`: triunfa si R es el resultado de adicionar al inicio de L a X.
 - f) `pop(L, R)`: triunfa si R es el resultado de eliminar el primer elemento de L.
 - g) `add(X, L, R)`: triunfa si R es el resultado de adicionar X al final de L.
3. Dados los predicados:

`hombre(X)`: triunfa si X es hombre.

`mujer(X)`: triunfa si X es mujer.

`progenitor(X,Y)`: triunfa si X es progenitor de Y.

`dif(X, Y)`: triunfa si X es diferente de Y.

Implemente las siguientes relaciones familiares:

- a) esMadre(X): triunfa si X es una mujer que tiene al menos un hijo.
- b) esPadre(X): triunfa si X es un hombre con al menos un hijo.
- c) hijoDe(X, Y): triunfa si X es hijo de Y.
- d) hermanaDe(X, Y): triunfa si X es hermana de Y.
- e) abueloDe(X, Y): triunfa si X es abuelo de Y.
- f) hermanos(X, Y): triunfa si X y Y son hermanos.
- g) tia(X, Y): triunfa si X es tía de Y.
- h) ancestro(X, Y): triunfa si X es ancestro de Y.

4. Realizar los ejercicios de la conferencia.

- a) Implementar member, push, pop y add a partir del concatenar.
- b) insertar(X, L, R): triunfa si R es el resultado de insertar a X en alguna posición de L.
- c) eliminar(X, L, R): triunfa si R es el resultado de eliminar X de la lista L.
- d) eliminar(L, R): triunfa si R es el resultado de eliminar el último elemento de la lista L.
- e) permutacion(L, R): triunfa si R es una permutación de L.
- f) consecutivos(X, Y, R): triunfa si X, Y son elementos consecutivos de L.
- g) reverso(L, R): triunfa si R es el reverso de L.
- h) palindromo(L): triunfa si L es palíndromo.
- i) subconjunto(S, L): triunfa si S es un subconjunto de la lista L.