**EJERCICIO 1 (1 punto).** Implementa un predicado pertenece (X, L) que compruebe si un elemento está en una lista. El predicado debe satisfacerse tantas veces como elementos repetidos contenga.

**EJERCICIO 2 (1 punto).** Implementa el predicado invierte(L, R) que se satisface cuando R contiene los elementos de L en orden inverso. Utiliza el predicado concatena/3:

Este ejercicio se resume en dos sentencias:

* El caso base es la lista vacia, esta se invierte a si misma, no dando false como resultado.
* El caso general se reduce a coger el primer elemento y concatenerlo por detrás de la inversión del resto de la lista.

**EJERCICIO 3 ​(1 punto).** Implementa los predicados preorder(Tree, List), inorder(Tree, List) y postorder(Tree, List) que recorren un árbol binario y devuelven el recorrido en una lista. Para representar árboles usaremos las funciones tree(Info, Left, Right) y nil. También usaremos el predicado concatena/3. Ejemplo:

Este ejercicio tiene tres apartados dependiendo del orden de inserción de los elementos, sin embargo, el algoritmo es el mismo para los tres casos:

INORDER:

PREORDER:

POSTORDER:

SI (I = NULL)

DEVOLVER lista vacia

EN OTRO CASO:

DEVOLVER concatenar(inorder(L), I, inorder(R)).

DEVOLVER concatenar( I,preorder(L) , preorder(R)).

DEVOLVER concatenar(postorder(L), postorder(R), I).

Dado que “nil” no es un símbolo definido para prolog, nos vemos obligados a añadir casos adicionales. Esto complica la programación y la hace más extensa que la que se podría realizar en otros lenguajes.

**EJERCICIO 4 (1 punto).** Implementar el predicado insertar(X, L, P, R) que inserte un elemento (X) en una lista (L) en una posición (P), desplazando el resto de elementos. Se considera que la primera posición de una lista es la 1.

En este ejercicio, creados un predicado que nos permite recorrer la lista de manera recursiva. La recursión se lleva a cabo utilizando el predicado “is”. Este nos permite ir reduciendo la posición en la que debemos insertar a la par que reducimos la lista quitando elementos por la parte de adelante.

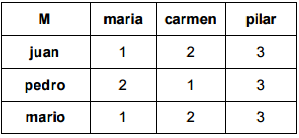
El caso de parada de esta recursión es cuando en la posición donde se quiere insertar tenemos un 1, en este caso concatenamos por delante mediante la realización de un “cons de prolog”

**EJERCICIO 5 (1 punto).** Implementar el predicado extract(L1, X, L2) que extrae un elemento (X) de una lista (L1) y deja en otra lista (L2) la lista original (L1) sin el elemento.

Siguiendo un prodecimiento similar al ejercicio anterior, en este caso vamos iterando hasta que el elemento a extraer (X) sea el primero de la lista, en este caso, devolvemos la lista restante.

La recursión en este caso se realiza cuando el primer elemento de la lista no es el elemto buscado. Cuando esto ocurre, llamamos a extraer sobre el resto de la lista y concatenamos el 1er elemento con la lista resultante de la recursión realizada.

**EJERCICIO 6​ (1 punto).** Lista de preferencias



Este ejercicio es sencillo, simplemente se realiza una regla por cada hombre/mujer en la que se especifica la lista de preferencias de la persona en cuestión ordenas de mayor a menor. Este ejercicio no consta de variables sino de constantes.

**EJERCICIO 7 (2 puntos).** Inestabilidad de los matrimonios

UNSTABLE(M1-W1, marriages)

SI MARRIGES = VACIO

RETURN FALSE

M\_Pref = obtenerPreferencias(M1)

M2-W2 = primero(marriages)

W\_Pref = obtenerPreferencias(W2)

SI posición(W2, M\_pref) < posición(W1, M\_pref) AND posición(M1, W\_pref) < posición(M2, W\_pref)

RETURN TRUE

ELSE

RETURN UNSTABLE(M1-W1, resto(Marriages))

**EJERCICIO 8 (2 puntos).** SMP

SMP(FreeMen, FreeWomen, MarriageIn)

SI FreeMen = vacio AND FreeWomen = vacio

RETURN MarriageIn

ELSE

W = primero(FreeWomen)

Pref = elección(W)

M = primero(Pref)

WHILE NOT pertenece(M, FreeMen)

M = siguiente(Pref)

SI NOT UNSTABLE(M-W, MarriageIn)

RETURN SMP(extraer(M, FreeMen), resto(FreeWomen), insertar(M-W, MarriageIn))

En prolog el bucle While lo conseguimos mediante el predicado de “pertenece” implementado en el ejercicio 1. Dicho predicado coge el primer elemento de la lista de preferencias de la mujer, cuando el predicado “extract” falla porque el hombre no está disponible, vuelve a la llamada a “pertenece” y coge el siguiente elemento de la lista, así hasta que encuentra un hombre en la lista de hombres ibres.