

Ejercicios de Programación Declarativa

Curso 2019/20

Hoja 6

En estos ejercicios usa el predicado del corte siempre que sea útil.

1. Escribe una versión del predicado *sumintersec*/3 utilizando la notación $(P \rightarrow Q; R)$.

$sumintersec(L_1, L_2, N) \longleftrightarrow L_1$ y L_2 son dos listas de enteros, **ordenadas de menor a mayor** y N es la suma de los elementos que están en la intersección de las dos listas.

2. *Implementación de conjuntos usando listas.* Escribe los siguientes predicados en Prolog:

(a) $nomiembro(X, L) \longleftrightarrow X$ no es un elemento de la lista L .

Utiliza este predicado para definir los siguientes.

(b) $hazconjunto(L, C) \longleftrightarrow C$ es un conjunto, representado por medio de una lista, que tiene los mismos elementos que L (no importa en que orden, pero sin que ninguno esté repetido).

(c) $union(C_1, C_2, C) \longleftrightarrow C$ es el conjunto formado por los elementos de la unión de los conjuntos C_1 y C_2 .

(d) $interseccion(C_1, C_2, C) \longleftrightarrow C$ es el conjunto formado por los elementos de la intersección de los conjuntos C_1 y C_2 .

3. Define en Prolog un predicado *treeSort*(L, LO), para implementar el algoritmo de ordenación de listas utilizando un árbol binario ordenado, que ya programaste en Haskell.

$treeSort(L, LO) \longleftrightarrow LO$ es la lista resultante de ordenar la lista L utilizando un árbol ordenado.

4. Sean A_1 y A_2 dos árboles binarios ordenados. Define un predicado para determinar si A_1 está contenido en A_2 . Es decir, cada nodo de A_1 es un nodo de A_2 . Haz uso de que los árboles están ordenados.

5. Programa en Prolog el siguiente predicado utilizando la aritmética de Prolog:

$polinomio(E, X) \longleftrightarrow$ la expresión E es un polinomio con incógnita X .

6. Considera el siguiente programa Prolog:

```
resta(X, cero, X).
```

```
resta(X, suc(Y), Z) :- resta(X, Y, suc(Z)).
```

(a) Describe el árbol de búsqueda de cada uno de los siguientes objetivos a partir del programa anterior:

$\backslash + \text{resta}(\text{suc}(\text{cero}), Y, \text{suc}^2(\text{cero})).$

$\backslash + \text{resta}(\text{suc}(\text{cero}), \text{suc}(\text{cero}), \text{suc}^2(\text{cero})).$

(b) Considera ahora la siguiente modificación del programa anterior:

```
resta'(X, cero, X) :- !.
```

```
resta'(X, suc(Y), Z) :- resta'(X, Y, suc(Z)).
```

Determina razonadamente cuáles son las respuestas de los siguientes objetivos Prolog:

```
resta(X, suc2(Y), suc2(cero)).
```

```
resta'(X, suc2(Y), suc2(cero)).
```

```
\+ resta'(suc3(cero), Y, suc(Z)).
```

7. Define un predicado Prolog para conseguir una lista con todas las sublistas de una lista dada, es decir:

$suslistas(L+, Ls-) \longleftrightarrow Ls$ es una lista cuyos elementos son las sublistas de la lista dada L .

$L+, Ls-$ indica que el modo de uso del primer argumento será de entrada y el segundo de salida.