Programación Declarativa

Sesión de laboratorio 3

Curso 2020/21

- Realiza los siguientes ejercicios individualmente en un mismo fichero .hs.
- Escribe tu nombre al comienzo del fichero como líneas comentadas. Incluye comentarios significativos y no olvides `declarar los tipos` de las expresiones que definas.
- Sube el fichero al Campus Virtual antes de que acabe la clase.

1. Define mediante `foldr` o `foldl`, en lugar de mediante recursión explícita, las siguientes funciones predefinidas en Prelude. Expresa mediante λ-expresiones el primer argumento de la la función `fold` que utilices.
   - `last`,
   - `reverse`,
   - `all`,
   - `minimum`,
   - `map`,
   - `filter`,
   - `takeWhile`.

2. Programa, las siguientes variantes de `foldl` y `foldr`, que operan con listas no vacías y no usan valor acumulado inicial:
   - $foldr1 \bigoplus [x_1, \ldots, x_n] = x_1 \bigoplus x_2 \bigoplus \ldots \bigoplus x_n$ (con $\bigoplus$ asociando por la derecha)
   - $foldl1 \bigoplus [x_1, \ldots, x_n] = x_1 \bigoplus x_2 \bigoplus \ldots \bigoplus x_n$ (con $\bigoplus$ asociando por la izquierda)

3. Define expresiones Haskell usando funciones de orden superior predefinidas de Haskell y/o listas intensionales para representar:
   (a) La lista $[1, -1, 2, -2, 3, -3, 4, -4, \ldots]$.
   (b) Una lista infinita $[(0,0), (0,1), (1,0), (0,2), (1,1), (2,0), (0,3), (1,2), \ldots]$, que sirva como enumeración de todas las parejas de números naturales.

   Puedes definir cada expresión de un par de formas diferentes.

4. Programa las siguientes funciones, usando orden superior o listas intensionales.
   (a) `sufijos xs` = lista de todos los sufijos de `xs`. Por ejemplo:
   `sufijos [1,2,3] = [[1,2,3],[2,3],[3],[]]`.
   (b) `sublistas xs` = lista de todas las sublistas de `xs`. Por ejemplo:
   `sublistas [1,2,3] = [[],[1],[2],[3],[1,2],[2,3],[1,2,3]]`.
   (c) `permutaciones xs` = lista de todas las permutaciones de `xs`. Por ejemplo:
   `permutaciones [1,2,3]=[[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]`.
   (d) `sumandos n` devuelve la lista de todas las descomposiciones en sumandos positivos de `n`. Po ejemplo:
   `sumandos 3 = [[1,1,1],[1,2],[3]]`.