

Combinadores recursivos y evaluación perezosa

Curso 2004/05, Fecha:2/12/2004

Enunciado 1 (primo) Definir una función `primo :: Integer → Bool` que indique si un número es primo

Enunciado 2 (ruffini) Definir una función `ruffini :: Float → [Float] → Float` que a partir de un valor a y los coeficientes de un polinomio devuelve el resto de dividir dicho polinomio entre $x - a$

```
?- ruffini 2 [1,1,-1,-1]
9
```

Enunciado 3 (d2b) Definir una función `d2b :: Int → [Int]` que convierte un número decimal en su valor binario representado como una lista de ceros y unos

```
?- d2b 13
[1,1,0,1]
```

Enunciado 4 (b2d) Definir una función `b2d :: [Int] → Int` que convierte un número binario representado como una lista de dígitos binarios en su valor decimal

```
?- b2d [1,1,0,1]
13
```

Enunciado 5 (Arboles) Definir un tipo de datos que represente árboles binarios y definir varios valores de dicho tipo.

- Definir una función `foldA` similar a la función **foldr** pero que trabaje con árboles
- Utilizando dicha función, construir funciones que devuelvan la suma de los nodos de un árbol, la altura de un árbol y los nodos del árbol

Enunciado 6 (Búsqueda(Opcional)) Un árbol binario de búsqueda cumple que en un nodo n cualquiera, todos los hijos de la izquierda son menores que n y todos los de la derecha son mayores que n .

- Definir una función `inserta :: Int → Arbol Int → Arbol Int` que toma un valor x y un árbol de búsqueda a y devuelve el árbol de búsqueda resultante de insertar x en a .

- Definir una función ordena $:: [\mathbf{Int}] \rightarrow [\mathbf{Int}]$ que ordena una lista insertando primero todos los nodos en un árbol de búsqueda y luego recorriendo dicho árbol

Enunciado 7 (Anchura(Opcional)) Definir una función que tome un árbol binario y devuelva la lista de nodos mediante un recorrido en anchura

Enunciado 8 (Listas infinitas) Definir funciones que generen las siguientes listas infinitas:

- Unos: $[1,1,1,1,\dots]$
- Números pares: $[2,4,6,8,10,\dots]$
- Sucesión de Fibonacci: $[1,1,2,3,5,8,13,\dots]$
- Números primos: $[1,2,3,5,7,11,13,17,\dots]$

Enunciado 9 (Tartaglia(Opcional)) Construir una función que genere el triángulo de Tartaglia. No es necesario realizar la representación gráfica, basta con representar cada fila como una lista de números. Por ejemplo, las 5 primeras filas serían: $[[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1],\dots]$

Enunciado 10 (Factorizar(Opcional)) Factorizar un número consiste en representarlo como el producto de números primos. Por ejemplo, la factorización del número 665 sería $[5,7,19]$. Construir la función factoriza $:: \mathbf{Integer} \rightarrow [\mathbf{Integer}]$ que toma un número n y devuelve la lista de números primos cuyo producto es n .

Nota: La factorización de números es la base de los algoritmos de encriptación RSA