



Práctica 1

1. Definir las siguientes funciones y determinar su tipo:
 - a) `five`, que dado cualquier valor, devuelve 5
 - b) `apply`, que toma una función y un valor, y devuelve el resultado de aplicar la función al valor dado
 - c) `id`, la función identidad
 - d) `first`, que toma un par ordenado, y devuelve su primera componente
 - e) `derive`, que aproxima la derivada de una función dada en un punto dado
 - f) `sign`, la función signo
 - g) `abs`, la función valor absoluto (usando `sign` y sin usarla)
 - h) `pot`, que toma un entero y un número, y devuelve el resultado de elevar el segundo a la potencia dada por el primero
 - i) `xor`, el operador de disyunción exclusiva
 - j) `max3`, que toma tres números enteros y devuelve el máximo entre ellos
 - k) `swap`, que toma un par y devuelve el par con sus componentes invertidas
2. Dar al menos dos ejemplos de funciones que tengan cada uno de los siguientes tipos:
 - a) $(\text{Int} \rightarrow \text{Int}) \rightarrow \text{Int}$
 - b) $\text{Int} \rightarrow (\text{Int} \rightarrow \text{Int})$
 - c) $(\text{Int} \rightarrow \text{Int}) \rightarrow (\text{Int} \rightarrow \text{Int})$
 - d) $\text{Int} \rightarrow \text{Bool}$
 - e) $\text{Bool} \rightarrow (\text{Bool} \rightarrow \text{Bool})$
 - f) $(\text{Int}, \text{Char}) \rightarrow \text{Bool}$
 - g) $(\text{Int}, \text{Int}) \rightarrow \text{Int}$
 - h) $\text{Int} \rightarrow (\text{Int}, \text{Int})$
 - i) $a \rightarrow \text{Bool}$
 - j) $a \rightarrow a$
3. Reescribir cada una de las siguientes definiciones sin usar `let`, `where` o `if`:
 - a) `f x = let (y,z) = (x,x) in y`
 - b) `greater (x,y) = if x > y then True else False`
 - c) `f (x,y) = let z = x + y in g (z,y) where g (a,b) = a - b`

4. Suponiendo que f y g tienen los siguientes tipos

$f :: c \rightarrow d$

$g :: a \rightarrow b \rightarrow c$

y sea h definida como

$h\ x\ y = f\ (g\ x\ y)$

Determinar el tipo de h e indicar cuáles de las siguientes definiciones de h son equivalentes a la dada:

$h = f \cdot g$

$h\ x = f \cdot (g\ x)$

$h\ x\ y = (f \cdot g)\ x\ y$

¿Cuál es el tipo de la función (\cdot) ?

5. Definir una función que determine si un año es bisiesto o no, de acuerdo a la siguiente definición:

año bisiesto 1. **m.** El que tiene un día más que el año común, añadido al mes de febrero. Se repite cada cuatro años, a excepción del último de cada siglo cuyo número de centenas no sea múltiplo de cuatro. (*Diccionario de la Real Academia Española*, 22^o ed.)

¿Cuál es el tipo de la función definida?

6. Sin usar funciones predefinidas, defina recursivamente las siguientes funciones y determine su tipo más general:

- a) **suma**, que suma todos los elementos de una lista de números
- b) **alguno**, que devuelve **True** si algún elemento de una lista de valores booleanos es **True**, y **False** en caso contrario
- c) **todos**, que devuelve **True** si todos los elementos de una lista de valores booleanos son **True**, y **False** en caso contrario
- d) **codes**, que dada una lista de caracteres, devuelve la lista de sus ordinales
- e) **restos**, que calcula la lista de los restos de la división de los elementos de una lista de números dada por otro número dado
- f) **cuadrados**, que dada una lista de números, devuelva la lista de sus cuadrados
- g) **longitudes**, que dada una lista de listas, devuelve la lista de sus longitudes
- h) **orden**, que dada una lista de pares de números, devuelve la lista de aquellos pares en los que la primera componente es menor que el triple de la segunda
- i) **pares**, que dada una lista de enteros, devuelve la lista de los elementos pares
- j) **letras**, que dada una lista de caracteres, devuelve la lista de aquellos que son letras (minúsculas o mayúsculas)
- k) **masDe**, que dada una lista de listas xss y un número n , devuelve la lista de aquellas listas de xss con longitud mayor que n

7. El producto escalar de dos listas de enteros de igual longitud es la suma de los productos de los elementos sucesivos (misma posición) de ambas listas. Usando listas por comprensión defina una función `scalarproduct` que devuelva el producto escalar de dos listas.

Sugerencia: Usar las funciones `zip` y `sum`.

- `zip [1,2,3] [4,5,6] = [(1,4),(2,5),(3,6)]`
- `sum [1,2,3] = 6`

8. Definir las siguientes funciones usando listas por comprensión:

- a) `divisors`, que dado un entero positivo x devuelve la lista de los divisores de x (y la lista vacía si el entero no es positivo).
- b) `matches`, que dados un entero x y una lista de enteros descarta de la lista los elementos distintos a x .
- c) `unique`, que dada una lista `xs` de enteros, devuelve la lista con los elementos no repetidos de `xs`.
- d) `cuadrupla`, que dados cuatro enteros a, b, c y d tales que $0 < a, b, c, d, \leq 100$, devuelve las cuadruplas (a, b, c, d) que cumplen $a^2 + b^2 = c^2 + d^2$.

9. Definir el tipo de datos `Direction` cuyos valores describan los puntos cardinales. Definir la función `move` que dado un punto en el plano (representado como un par de enteros) y una dirección devuelva el punto que se obtiene al desplazarse una unidad en dicha dirección.

10.

- a) Definir las operaciones de suma y producto módulo 2 para el tipo

`data DigBin = Cero | Uno`

- b) Definir las operaciones de suma binaria, producto por dos, cociente y resto de la división por dos para el tipo:

`type NumBin = [Digbin]`

donde convenimos que el primer elemento de la lista de dígitos es el dígito menos significativo del número representado.

- c) Redefinir las funciones del ítem anterior, observando una convención opuesta.
- d) Definir funciones que multipliquen n números binarios de acuerdo a las dos convenciones.

11. Dada las siguientes representaciones de los árboles generales y de los árboles binarios

`data GenTree a = EmptyG | NodeG a [GenTree a]`

`data BinTree a = EmptyB | NodeB (BinTree a) a (BinTree a)`

definir una función que dado un árbol general, lo transforme en un árbol binario de la siguiente manera:

Cada nodo `NodeG` en orden en el árbol general corresponde a un nodo `NodeB` en el árbol binario; el hijo de la izquierda de `NodeB` es el nodo correspondiente al primer hijo de `NodeG`, y el hijo derecho de `NodeB` es el nodo correspondiente al siguiente hermano de `NodeG`, es decir, el próximo nodo en orden entre los hijos de los padres de `NodeG`.