

Programación Funcional - Práctico 2

Curso 2017

1. (a) Implemente la función *map* usando listas por comprensión.
(b) Implemente la función *filter* usando listas por comprensión.
2. Usando *map*, defina una función *squares* $:: [Int] \rightarrow [Int]$ que dada una lista de enteros retorne una lista con los cuadrados de los elementos de la lista.
3. Defina la función *length* en términos de *map* y *sum*.
4. Usando *filter*, defina una función *all* $:: (a \rightarrow Bool) \rightarrow [a] \rightarrow Bool$ que dada una condición y una lista, verifique si todos los elementos de la lista cumplen con dicha condición. Ejemplos:

all (>0) $[1, 2, 3]$ retorna *True*

all ($\equiv 'a'$) $['a', 'b', 'c']$ retorna *False*

5. Indique el tipo y explique lo que hace la siguiente función:

$$rara\ p = filter\ p \circ filter\ (not \circ p)$$

6. Dada la siguiente función

$$dup\ x = (x, x)$$

Explique en que difieren $(dup \circ dup)$ y $(dup\ dup)$.

7. Indique el tipo y explique lo que hace la siguiente función:

$$rara2 = zipWith\ (\circ)\ [length, sum]\ [drop\ 4, take\ 4]$$

Muestre un ejemplo de aplicación correcta de la expresión $(head\ rara2)$ y su resultado.

8. Dada la función *twice*

$$twice\ f = f \circ f$$

Explique el resultado de hacer *twice tail* $[1, 2, 3, 4]$. ¿Es posible hacer *twice head* $[1, 2, 3, 4]$? Justifique.

9. La función *flip* tiene el siguiente tipo: $(a \rightarrow b \rightarrow c) \rightarrow b \rightarrow a \rightarrow c$. Observando el tipo, ¿puede determinar qué hace la función? ¿Puede implementarla?
10. (a) Utilizando *flip*, *mod*, *length*, *map* y *filter*, defina una función que dada una lista de enteros retorne la cantidad de elementos pares que tiene la lista.
- (b) Haga lo mismo, pero sin usar *map*.
- (c) Haga lo mismo, pero sin usar *flip*.
11. Sea $h\ x\ y = f\ (g\ x\ y)$. ¿Cuáles de las siguientes afirmaciones son correctas?
- (a) $h \equiv f \circ g$
- (b) $h\ x \equiv f \circ g\ x$
- (c) $h\ x\ y \equiv (f \circ g)\ x\ y$
12. Se puede definir la función *filter* en términos de *concat* y *map*:

$$\begin{aligned} \text{filter } p &= \text{concat} \circ \text{map } \text{box} \\ &\quad \textbf{where } \text{box } x = \dots \end{aligned}$$

Dar la definición de *box*.

13. Explique por qué la siguiente definición no es aceptada por el sistema de tipos de Haskell:

$$\text{dobleAp } f = (f\ \text{True}, f\ 'a')$$