Programación Funcional

Prueba Escrita - 2024

Nombre:

CI:

Número:

---

1. Dada la siguiente definición:

   $$foo\ a\ b\ c\ d = foldr\ (+)\ c\ [fst\ a, \textbf{if}\ b == snd\ a\ \textbf{then}\ d\ \textbf{else}\ d + 1]$$

   El tipo más general es:

   (a) $foo :: (Num\ a, Eq\ b, Num\ c) \Rightarrow (a, b) \to b \to c \to a \to c$
   (b) $foo :: (Num\ a, Eq\ a) \Rightarrow (a, a) \to a \to a \to a \to a$
   (c) $foo :: (Num\ a, Eq\ b) \Rightarrow (a, b) \to b \to a \to a \to a$
   (d) No tiene

   Respuesta: c)

2. Dada la siguiente definición:

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

   ¿Cuál de las siguientes opciones **NO** es correcta?:

   (a) *twice twice* está mal tipada
   (b) *twice head* está mal tipada
   (c) El tipo más general de $twice \circ twice$ es $(a \to a) \to a \to a$
   (d) El tipo más general de *twice tail* es $[a] \to [a]$

   Respuesta: a)

3. Dada la siguiente definición:

   $$rara = zipWith\ flip\ (repeat\ div)\ [0\,..]$$

   ¿Cuál de las siguientes opciones **NO** es correcta?

   (a) $(rara\ !!\ 0)\ 0$, da error de ejecución
   (b) $(rara\ !!\ 1)\ 0$, da error de ejecución
   (c) $(rara\ !!\ 0)\ 1$, da error de ejecución
   (d) $(rara\ !!\ 10)\ 10$, retorna 1

   Respuesta: b)

4. Dada la siguiente definición:

$$
\begin{array}{l}
\textbf{data}\ T\ a\ b = E \mid NA\ (T\ a\ b)\ a\ (T\ a\ b) \mid NB\ (T\ a\ b)\ b\ (T\ a\ b) \\
foo\ E \qquad\qquad = ([\,], [\,]) \\
foo\ (NA\ l\ x\ r) = \textbf{let}\ (ll, lr) \ = foo\ l \\
\qquad\qquad\qquad\qquad\quad\ (rl, rr) = foo\ r \\
\qquad\qquad\qquad\quad\ \textbf{in}\ \ (x : ll +\!\!+ rl, lr +\!\!+ rr) \\
foo\ (NB\ l\ x\ r) = \textbf{let}\ (ll, lr) \ = foo\ l \\
\qquad\qquad\qquad\qquad\quad\ (rl, rr) = foo\ r \\
\qquad\qquad\qquad\quad\ \textbf{in}\ \ (ll +\!\!+ rl, x : lr +\!\!+ rr)
\end{array}
$$

Indique la opción correcta:

(a) $foo\ (NA\ (NB\ (NA\ E\ 10\ E)\ \texttt{'a'}\ (NB\ E\ \texttt{'s'}\ E))\ 9\ (NA\ E\ 89\ E))$, retorna $([9, 10, 89], [\texttt{'a'}, \texttt{'s'}])$

(b) $foo\ (NA\ (NB\ (NA\ E\ \texttt{'a'}\ E)\ True\ (NB\ E\ False\ E))\ \texttt{'g'}\ (NA\ E\ \texttt{'w'}\ E))$, retorna $([\texttt{'w'}, \texttt{'a'}, \texttt{'g'}], [False, True])$

(c) $foo\ (NA\ (NB\ (NA\ E\ E\ E)\ E\ (NB\ E\ E\ E))\ E\ (NA\ E\ E\ E))$, no compila

(d) $foo\ (NA\ (NB\ (NA\ E\ 1\ E)\ Nothing\ (NB\ E\ Nothing\ E))\ 2\ (NA\ E\ 3\ E))$, retorna $([1, 2, 3], [Nothing, Nothing])$

Respuesta: a)

5. ¿Cuál de las siguientes definiciones no equivale a las otras tres:?

(a)
$$
\begin{array}{l}
foo\ p\ xs = [x + y \mid (x, y) \leftarrow xs, not\ (p\ x)]
\end{array}
$$

(b)
$$
\begin{array}{l}
foo\ p = map\ (uncurry\ (+)) \circ filter\ (not \circ p \circ fst)
\end{array}
$$

(c)
$$
\begin{array}{l}
foo\ p\ [\,] = [\,] \\
foo\ p\ ((x, y) : xs) \mid not\ (p\ x)\ = x + y : foo\ p\ xs \\
\qquad\qquad\qquad\quad\ \mid otherwise = foo\ p\ xs
\end{array}
$$

(d)
$$
\begin{array}{l}
foo\ p = zipWith\ (+) \circ filter\ (\lambda(x, \_) \rightarrow not\ (p\ x))
\end{array}
$$

Respuesta: d)

6. La función

$$split :: [a] \rightarrow ([a], [a])$$

divide una lista en dos listas colocando sus elementos de forma alternada. Por ejemplo, $split\ [2, 4, 6, 8, 7]$ retorna $([2, 6, 7], [4, 8])$. Se puede implementar *usando foldl* de la siguiente forma:

$$split = foldl\ step\ ([\,], [\,]) \circ ini$$

Indique la opción que permite una implementación correcta de *split*.

(a)
$$
\begin{array}{l}
ini = id \\
step\ (ls, rs)\ x = (x : ls, rs)
\end{array}
$$

(b)
$$
\begin{array}{l}
ini = id \\
step\ (ls, rs)\ x = (x : rs, ls)
\end{array}
$$

(c)
$$
\begin{array}{l}
ini = reverse \\
step\ (ls, rs)\ x = (x : ls, rs)
\end{array}
$$

(d)
$$
\begin{array}{l}
ini = reverse \\
step\ (ls, rs)\ x = (x : rs, ls)
\end{array}
$$

Respuesta: d)

7. Dadas la siguientes definiciones:

$$loop = tail\ loop$$
$$ls = (head\ loop) : rs$$
$$rs = 1 : map\ (+1)\ ls$$

Para cada una de las siguientes expresiones indique el resultado de su evaluación o si la misma diverge (si pone diverge en todas las opciones anula la pregunta).

(a) $head\ rs$ — 1

(b) $length\ (take\ 3\ rs)$ — 3

(c) $(head \circ tail \circ tail)\ rs$ — 2

(d) $(head \circ tail)\ rs$ — diverge

(e) $take\ 4\ \$\ map\ snd\ \$\ filter\ ((/= 0) \circ (`mod`\ 2) \circ fst)\ \$\ zip\ [1..]\ rs$ — [1,2,3,4]

(f) $take\ 4\ \$\ map\ fst\ \$\ filter\ ((/= 0) \circ (`mod`\ 2) \circ fst)\ \$\ zip\ [1..]\ rs$ — [1,3,5,7]

(g) $length \circ take\ 4\ \$\ filter\ (/= 0)\ rs$ — diverge

(h) $head \circ head\ \$\ map\ (:loop)\ rs$ — 1

8. Consideramos la siguiente representación de los números naturales.

$$\textbf{data}\ Nat = Zero \mid Succ\ Nat$$
$$foldN :: (a \to a) \to a \to Nat \to a$$
$$foldN\ h\ e\ Zero = e$$
$$foldN\ h\ e\ (Succ\ n) = h\ (foldN\ h\ e\ n)$$

¿Cuál de las siguientes afirmaciones **NO** es correcta?

(a) Dado un natural n, la aplicación $foldN\ (+1)\ 0\ n$ retorna el entero equivalente

(b) Dado un natural n, la aplicación $foldN\ ([]{+\!\!+})\ []\ n$ retorna una lista con n listas vacías

(c) Dados dos naturales n y m, la aplicación $foldN\ Succ\ m\ n$ retorna el natural que resulta de la suma de n y m

(d) Dado un natural n, la aplicación $foldN\ (Succ \circ Succ \circ Succ)\ Zero\ n$ retorna el natural que triplica a n

Respuesta: b)

9. Implemente usando *recursión explícita* la función:

$$\mathit{filterElem} :: \mathit{Eq}\ a \Rightarrow [a] \to (a \to [a])$$

que dada una lista de elementos comparables devuelve una función que dado un elemento retorna la lista sin ese elemento. Por ejemplo, $\mathit{filterElem}\ [1,2,3,4,5,2,3,4,3]\ 3$ devuelve $[1,2,4,5,2,4]$.

$$\begin{aligned}&\mathit{filterElem}\ [\,] \quad\ = \mathit{const}\ [\,]\\ &\mathit{filterElem}\ (x : xs) = \lambda e \to \mathbf{if}\ x == e\ \mathbf{then}\ rs\ e\ \mathbf{else}\ x : rs\ e\\ &\quad \mathbf{where}\ rs = \mathit{filterElem}\ xs\end{aligned}$$

Implemente la misma función, pero *como un foldr*.

$$\mathit{filterElem} = \mathit{foldr}\ (\lambda x\ rs\ e \to \mathbf{if}\ x == e\ \mathbf{then}\ rs\ e\ \mathbf{else}\ x : rs\ e)\ (\mathit{const}\ [\,])$$

10. Implemente, sin usar recursión, el operador

$$(.*>) :: (a \to [b]) \to (b \to c) \to (a \to [c])$$

que dados $f.*>g$ retorna la función equivalente a aplicar g a los elementos de la lista resultante de aplicar f. Por ejemplo, $(\mathit{replicate}\ 4.*>\mathit{show})\ 2$ devuelve la lista `["2","2","2","2"]`.

$$f.*>g = \mathit{map}\ g \circ f$$