Programación Funcional

Prueba Escrita - 2020

Nombre:

CI:

---

1. Dada la siguiente definición:

$$foo\ a\ b\ c\ d\ e = ((a == e)\ ||\ (b > d), c + d)$$

El tipo más general es:

(a) $foo :: (Eq\ a) \Rightarrow a \to Int \to Int \to Int \to a \to (Bool, Int)$

(b) $foo :: (Eq\ a, Ord\ b, Num\ b) \Rightarrow a \to b \to b \to b \to a \to (Bool, b)$

(c) $foo :: (Eq\ a, Ord\ b, Num\ b, Num\ c) \Rightarrow a \to b \to c \to b \to a \to (Bool, b)$

(d) $foo :: (Eq\ a, Eq\ d, Ord\ b, Ord\ c, Num\ c) \Rightarrow a \to b \to c \to c \to d \to (Bool, c)$

Respuesta: b)

2. Dada la siguiente definición:

$$\begin{array}{ll} tip\ f\ [\,] & \_\ \ zs = reverse\ zs \\ tip\ f\ \_ & [\,]\ \ zs = reverse\ zs \\ tip\ f\ (x : xs)\ ys\ zs = & tip\ f\ xs\ (tail\ ys)\ (f\ (head\ ys)\ x : zs) \end{array}$$

Considerando que xs, ys y zs son listas finitas, ¿cuál de las siguientes afirmaciones es **incorrecta**?

(a) $tip\ f\ xs\ ys\ [\,]$ es equivalente a $zipWith\ (flip\ f)\ xs\ ys$

(b) $tip\ f\ xs\ ys\ zs$ es equivalente a $reverse\ zs +\!\!+ tip\ f\ xs\ ys\ [\,]$

(c) $tip\ f\ xs\ ys\ zs$ es equivalente a $reverse\ (zipWith\ f\ ys\ xs +\!\!+ reverse\ zs)$

(d) $tip\ f\ xs\ ys\ zs$ es equivalente a $reverse\ zs +\!\!+ zipWith\ f\ ys\ xs$

Respuesta: c)

3. Dadas las siguientes definiciones:

$$\textbf{data } Foo\ a = Foo\ a\ [Foo\ a]$$
$$foo\ e\ (Foo\ x\ y) = x : foldl\ foo\ e\ y$$

¿Cuál de las siguientes afirmaciones es correcta?

(a) *foo* no compila.
(b) El tipo más general de *foo* es $foo :: [a] \to Foo\ a \to [a]$.
(c) El tipo más general de *foo* es $foo :: a \to Foo\ a \to [a]$.
(d) El tipo más general de *foo* es $foo :: a \to Foo\ [a] \to [a]$.

Respuesta: b)

4. Implemente utilizando **recusión explícita** la función:

$takeUntil :: (a \to Bool) \to [a] \to [a]$

que aplicada a un predicado p y una lista xs retorna el prefijo más grande de xs (posiblemente vacío) que no cumple con el predicado p. Por ejemplo: $takeUntil\ (>3)\ [2,1,5,3,6]$ retorna la lista $[2,1]$.

$$\begin{aligned} takeUntil\ p\ [\,] &= [\,] \\ takeUntil\ p\ (x:xs) \mid p\ x &= [\,] \\ \mid otherwise &= x : takeUntil\ p\ xs \end{aligned}$$

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

$$takeUntil\ p = foldr\ (\lambda x\ xs \to \textbf{if } p\ x \textbf{ then } [\,] \textbf{ else } x : xs)\ [\,]$$

5. Indique a cuál de las siguientes funciones **no** se le puede asignar el tipo $(a \to b) \to a \to [a] \to b$

(a) $f1\ f\ a\ b = f1\ (\lambda x \to f\ a)\ a\ b$
(b) $f2\ f\ \_\ b = f\ (head\ b)$
(c) $f3\ f\ \_\ b = f3\ f\ (head\ b)\ (tail\ b)$
(d) $f4\ f\ a\ b = head\ [f\ a, f4\ f\ (tail\ a)\ b]$

Respuesta: d)

6. Dadas las siguientes definiciones:

$$\textbf{data } T = N \mid C\ T\ T$$
$$\begin{aligned} baz\ f\ k\ N &= [\,] \\ baz\ f\ k\ (C\ t\ N) &= f\ (baz\ f\ (k+1)\ t) \end{aligned}$$

$$baz\ f\ k\ (C\ t\ u)\ = k : baz\ f\ (k+1)\ u$$
$$tree = C\ N\ (C\ tree\ N)$$

¿Cuál de las siguientes afirmaciones es **incorrecta**?

(a) $take\ 3\ \$\ baz\ id\ 0\ tree$ retorna $[0, 2, 4]$

(b) $take\ 3\ \$\ baz\ (map\ (+1))\ 0\ tree$ retorna $[0, 3, 6]$

(c) $head\ \$\ filter\ (\leqslant 0)\ \$\ baz\ id\ 0\ tree$ diverge

(d) $tail\ \$\ baz\ tail\ 0\ tree$ diverge

Respuesta: c)

7. Dadas las siguientes definiciones:

$$rep\ p\ f\ x = \textbf{let}\ x' = f\ x \\ \textbf{in if}\ p\ x'\ \textbf{then}\ rep\ p\ f\ x'\ \textbf{else}\ x'$$
$$foo\ n\ m = snd\ \$\ rep\ ((\leqslant)\ m \circ fst)\ (\lambda(x, y) \to (x - m, y + 1))\ (n, 0)$$

¿Cuál de las siguientes afirmaciones es **incorrecta**?

(a) El tipo de rep es $rep :: (a \to Bool) \to (a \to a) \to a \to a$

(b) El resultado de evaluar foo 45 10 es 4

(c) El resultado de evaluar foo 20 100 es 0

(d) El resultado de evaluar $rep\ ((>0) \circ head)\ tail\ [1, 2, -3, 4]$ es $[-3, 4]$

Respuesta: c)

8. Dadas las siguientes definiciones que implementan a una calculadora:

$$\textbf{type}\ Calc = Int \to Int$$
$$mas, menos, por, dividido :: Int \to Calc$$
$$mas\ \ n = \lambda m \to m + n$$
$$menos\ \ n = \lambda m \to m - n$$
$$por\ \ n = \lambda m \to m * n$$
$$dividido\ n = \lambda m \to m\ `div`\ n$$

Implemente el operador $(>\!\!=\!\!>) :: Calc \to Calc \to Calc$, para secuenciar operaciones, tal que por ejemplo $(mas\ 2\ >\!\!=\!\!>\ menos\ 1\ >\!\!=\!\!>\ por\ 10)\ 20$ resulta en 210.

$$op1\ >\!\!=\!\!>\ op2 = \lambda m \to op2\ (op1\ m)$$

9. Dadas las siguientes definiciones:

$$
\begin{aligned}
&\textbf{data } \mathit{Tree}\ a = \mathit{Node}\ (\mathit{Tree}\ a)\ a\ (\mathit{Tree}\ a) \mid \mathit{Empty}\\
&\mathit{mkTree}\ n = \mathit{Node}\ (\mathit{mkTree}\ \$\ n+1)\ n\ (\mathit{mkTree}\ \$\ n+2)\\
&\mathit{goLeft}\ (\mathit{Node}\ l\ a\ \_) \quad = a : \mathit{goLeft}\ l\\
&\mathit{goRight}\ (\mathit{Node}\ \_\ a\ r) = a : \mathit{goRight}\ r\\
&\mathit{recorre}\ t = \mathit{go}\ [t]\\
&\mathit{go}\ (\mathit{Node}\ l\ v\ r : ts) \quad = v : \mathit{go}\ (ts +\!\!+ [l] +\!\!+ [r])
\end{aligned}
$$

Para cada una de las siguientes expresiones indique el resultado de su evaluación o si la misma diverge.

(a) $(\mathit{take}\ 4 \circ \mathit{goLeft}\ \$\ \mathit{mkTree}\ 1)$ — [1,2,3,4]

(b) $(\mathit{take}\ 4 \circ \mathit{goRight}\ \$\ \mathit{mkTree}\ 1)$ — [1,3,5,7]

(c) $(\mathit{take}\ 3 \circ \mathit{foldr}\ (:)\ [0] \circ \mathit{goRight}\ \$\ \mathit{mkTree}\ 1)$ — [1,3,5]

(d) $(\mathit{take}\ 3 \circ \mathit{foldl}\ (\mathit{flip}\ (:))\ [0] \circ \mathit{goRight}\ \$\ \mathit{mkTree}\ 1)$ — diverge

(e) $(\mathit{head}\ (\mathit{goLeft}\ (\mathit{mkTree}\ 1) +\!\!+ \mathit{goRight}\ (\mathit{mkTree}\ 1)))$ — 1

(f) $(\mathit{take}\ 4 \circ \mathit{recorre}\ \$\ \mathit{mkTree}\ 1)$ — [1,2,3,3]

(g) $(0 == (\mathit{length} \circ \mathit{filter}\ (<4) \circ \mathit{recorre}\ \$\ \mathit{mkTree}\ 1))$ — diverge

(h) $(\mathit{null} \circ \mathit{filter}\ (<4) \circ \mathit{recorre}\ \$\ \mathit{mkTree}\ 1)$ — False

10. Dada la siguiente definición:

$$
\begin{aligned}
&\mathit{main} = \textbf{do}\ \mathit{ps} \leftarrow \mathit{sequence} \circ \mathit{take}\ 2 \circ \mathit{repeat}\ \$\ \mathit{putStr}\ \texttt{"hola"}\\
&\qquad\qquad \mathit{putStr}\ \$\ \mathit{show}\ \mathit{ps} +\!\!+ \texttt{"chau"}
\end{aligned}
$$

¿Qué imprime el programa?

(a) `[(),()]chau`

(b) `holaholachau`

(c) Infinitos `hola`

(d) `holahola[(),()]chau`

Respuesta: d)