Programación Funcional

Primera Prueba Escrita - 2023

Nombre:

CI:

---

1. Dada la siguiente definición:

   $$uuf = uncurry\ (uncurry\ flip)$$

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

   (a) El tipo más general de la función es $uuf :: ((a \to b \to c, b), a) \to c$

   (b) $curry\ uuf$ es equivalente a $uncurry\ flip$

   (c) $uuf\ ((flip\ (+\!\!+), [3,4]), [1,2]) == [1,2,3,4]$

   (d) $uuf\ ((take, [1,2,3,4]), 2) == [1,2]$

   Respuesta: c)

2. Dada la siguiente definición:

   $$\begin{aligned} &itera\ f\ n\ xs = (iterate\ f\ xs)\ !!\ n \\ &\quad \textbf{where}\ iterate\ f\ x = x : iterate\ f\ (f\ x) \end{aligned}$$

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

   (a) $itera\ (drop\ 1)\ n\ xs == drop\ n\ xs$

   (b) $itera\ id\ n\ xs == xs$

   (c) $itera\ tail\ n\ xs == drop\ n\ xs$

   (d) $itera\ (take\ 1)\ n\ xs == take\ 1\ xs$

   Respuesta: c)

3. Dadas las siguientes definiciones:

   $$\begin{aligned} &nats = 0 : map\ (+1)\ nats \\ &testNats\ p = \textbf{case}\ dropWhile\ p\ nats\ \textbf{of} \\ &\qquad\qquad [\,] \to True \\ &\qquad\qquad \_ \to False \end{aligned}$$

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

   (a) $testNats\ (\leqslant 10) == False$

   (b) $testNats\ (const\ False) == False$

(c) $testNats\ (== 0) == False$

(d) $testNats\ (\geqslant 0) == True$

Respuesta: d)

4. Dadas las siguientes definiciones:

$$\begin{aligned}
&\textbf{data}\ T = L\ Int \mid F\ T\ T\\
&\\
&list :: T \to [Int]\\
&list\ (L\ x) = [x]\\
&list\ (F\ l\ r) = list\ r +\!\!+ list\ l\\
&\\
&buildT :: [Int] \to T\\
&buildT\ [x] = L\ x\\
&buildT\ (x : xs) = F\ (L\ x)\ (buildT\ xs)
\end{aligned}$$

¿Cuál de las siguientes afirmaciones es **incorrecta**? Suponga xs no vacía.

(a) $buildT\ xs == foldr\ (\lambda x\ t \to F\ (L\ x)\ t)\ (L\ (last\ xs))\ (init\ xs)$

(b) $buildT\ xs == foldl\ (\lambda t\ x \to F\ t\ (L\ x))\ (L\ (head\ xs))\ (tail\ xs)$

(c) $list\ (buildT\ xs) == reverse\ xs$

(d) $list\ (foldl\ (\lambda t\ x \to F\ (L\ x)\ t)\ (L\ (head\ xs))\ (tail\ xs)) == xs$

Respuesta: b)

5. Dada la siguiente definición:

$$\begin{aligned}
&foo\ x\ y = \textbf{do}\ s \leftarrow getLine\\
&\qquad\qquad putStrLn\ (show\ x +\!\!+ s +\!\!+ show\ y)\\
&\qquad\qquad t \leftarrow getLine\\
&\qquad\qquad return\ (s +\!\!+ t)
\end{aligned}$$

El tipo más general es:

(a) $foo :: Show\ a \Rightarrow a \to a \to IO\ ()$

(b) $foo :: Show\ a \Rightarrow a \to a \to IO\ String$

(c) $foo :: (Show\ a, Show\ b) \Rightarrow a \to b \to IO\ String$

(d) $foo :: (Show\ a, Show\ b) \Rightarrow a \to b \to IO\ ()$

Respuesta: c)

6. Dadas las siguientes definiciones:

$$\begin{aligned}
&\textbf{data}\ Nat = Z \mid S\ Nat\\
&\\
&ntimes\ f\ x\ Z\ \ \ \ \ \ = id\\
&ntimes\ f\ x\ (S\ n) = curry\ f\ x\ .\ ntimes\ f\ x\ n
\end{aligned}$$

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

(a) $ntimes\ (uncurry\ (+))\ 1\ (S\ (S\ Z))\ m == m + 2$, para $m :: Num\ a \Rightarrow a$

(b) El tipo más general de la función es $ntimes :: (a \to b \to b) \to a \to Nat \to b \to b$

(c) $ntimes\ fst\ True\ (S\ Z)\ b == True$, para $b :: Bool$

(d) $ntimes\ snd\ True\ (S\ Z)\ v == v$, para cualquier v

Respuesta: b)

7. Dada la siguiente definición:

$$takeS\ xs\ (z : zs) = show\ z : map\ show\ (take\ (length\ zs)\ xs)$$

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

(a) El tipo más general es $takeS :: (Show\ a, Show\ b) \Rightarrow [a] \to [b] \to [String]$

(b) $takeS$ no compila correctamente

(c) El tipo más general es $takeS :: (Show\ a) \Rightarrow [a] \to [Char] \to [String]$

(d) El tipo más general es $takeS :: Show\ a \Rightarrow [a] \to [a] \to [String]$

Respuesta: a)

8. Considere la siguiente definición de árbol binario formado por nodos internos y hojas:

data $Arb = H \mid N\ Arb\ Arb$

Un *camino* en un árbol es una secuencia de pasos que comienza en la raíz y termina en una hoja H. Vamos a representar secuencias de esta clase mediante el siguiente tipo:

type $Sec = [Paso]$
data $Paso = I \mid D$

tal que, estando en un nodo interno, I representa tomar a la izquierda y D tomar a la derecha. Por ejemplo, dado el árbol $t = N\ H\ (N\ H\ H)$, las secuencias $[I]$, $[D, I]$, $[D, D]$ son caminos en ese árbol. En cambio, las secuencias $[]$, $[D]$, $[I, I]$ no lo son.

(a) Escribir una función $esCamino :: Sec \to Arb \to Bool$ que dada una secuencia de pasos y un árbol determina si la secuencia es efectivamente un camino en el árbol.

```
esCamino :: Sec → Arb → Bool
esCamino [] H = True
esCamino (I : ps) (N l r) = esCamino ps l
esCamino (D : ps) (N l r) = esCamino ps r
esCamino _ _ = False
```

(b) Escribir una función $caminos :: Arb \to [Sec]$, que dado un árbol lista todos sus caminos. El orden en que los caminos son listados es irrelevante.

```
caminos :: Arb → [Sec]
caminos H = [[]]
caminos (N l r) = map (I:) (caminos l) ++ map (D:) (caminos r)
```

9. Dadas las siguientes definiciones:

$$loop\ x = map\ (const\ x)\ (loop\ x)$$
$$a = map\ loop\ [1..]$$
$$z = 0 : zipWith\ (curry\ snd)\ a\ [1..]$$

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

(a) $last\ (foldl\ (flip\ (:))\ []\ z)$

diverge

(b) $take\ 5\ (loop\ 3)$

diverge

(c) $map\ (const\ 1)\ (take\ 5\ a)$

[1,1,1,1,1]

(d) $sum\ \$\ take\ 5\ z$

10

(e) $length\ \$\ map\ head\ \$\ take\ 5\ a$

5

(f) $head\ \$\ tail\ \$\ foldr\ (:)\ (loop\ 0)\ [1, 2]$

2

(g) $a\ !!\ 2$

diverge

(h) $length\ \$\ takeWhile\ (==\ True)\ (loop\ False)$

diverge