Programación Funcional

Segunda Prueba - 2023

Nombre:

CI:

---

1. Dada la siguiente definición:

   $mapCut\ f\ xs = map\ f\ \$\ cut\ 2\ xs$
   $\quad$ **where** $cut\ n\ xs = take\ n\ xs +\!\!+ cut\ n\ (drop\ n\ xs)$

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

   (a) $mapCut\ (const\ 1)\ [\,]$ diverge
   (b) El tipo más general de $mapCut$ es $([a] \to b) \to [a] \to [b]$
   (c) $mapCut\ id\ [[1,2],[3,4]]$ retorna $[[1,2],[3,4]]$
   (d) $mapCut\ (+1)\ [1,2,3,4,5]$ no compila

   Respuesta: a)

2. Dadas las siguientes definiciones:

   $ones = 1 : ones$
   $inter\ (x : xs)\ ys = x : inter\ ys\ xs$
   $ww = inter\ ones\ (ww +\!\!+ reverse\ ones)$
   $zz = inter\ zz\ (ones +\!\!+ reverse\ ones)$

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

   (a) la expresión $(head\ (tail\ ww))$ reduce al valor 1
   (b) la expresión $(head\ ww)$ reduce al valor 1
   (c) la expresión $(head\ zz)$ reduce al valor 1
   (d) la expresión $(head\ \$\ zipWith\ (+)\ (tail\ (ww +\!\!+ zz))\ ones)$ reduce al valor 2

   Respuesta: c)

3. Dadas las siguientes definiciones:

   data $GTree\ a = A\ a \mid B\ [GTree\ a]$

   $mapG\ f\ (A\ x)\ \ = A\ [f\ x]$
   $mapG\ f\ (B\ gs) = B\ .\ foldl\ (flip\ (:))\ [\,]\ .\ map\ (mapG\ f)\ \$\ gs$

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

   (a) $mapG\ head\ \$\ mapG\ id\ (B\ [A\ 1, A\ 2])$ retorna $B\ [A\ [1], A\ [2]]$
   (b) El tipo más general de $mapG$ es $(a \to b) \to GTree\ a \to GTree\ [b]$
   (c) $mapG\ (tail\ .\ head)\ \$\ mapG\ (:[\,])\ (B\ [A\ 1, A\ 2])$ retorna $B\ [A\ [1], A\ [2]]$
   (d) $mapG\ (flip\ const\ 1)\ (B\ [A\ 1, A\ 2])$ retorna $B\ [A\ [2], A\ [1]]$

   Respuesta: c)

4. Dada la siguiente definición:

$$\begin{aligned} sec\ m\ m'\ v = \mathbf{do}\ & x \leftarrow m\ v \\ & \mathbf{let}\ z = [x] \\ & y \leftarrow m'\ z \\ & print\ (x : y) \end{aligned}$$

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

(a) Al evaluar $(sec\ (return\ .\ const\ 2)\ return\ 3)$ se imprime la lista $[2, 2]$
(b) Al evaluar $(sec\ return\ return\ 3)$ se imprime la lista $[3, 3]$
(c) Al evaluar $(sec\ return\ (\lambda x \to return\ 2 \gg return\ x)\ [\,])$ se imprime la lista $[[\,], [\,]]$
(d) El tipo más general de *sec* es $Show\ b \Rightarrow (a \to IO\ b) \to (b \to IO\ [b]) \to a \to IO\ ()$

Respuesta: d)

5. Dadas las siguientes definiciones:

$$\mathbf{data}\ T\ a\ b = T\ (a, b)$$

$$\begin{aligned} foo\ (T\ (True, x)) &= x \\ foo\ (T\ z) &= (fst\ z, T\ (True, True)) \end{aligned}$$

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

(a) El tipo de *foo* es $T\ Bool\ (Bool, T\ Bool\ Bool) \to (Bool, T\ Bool\ Bool)$.
(b) El tipo de *foo* es $T\ Bool\ a \to (a, a)$.
(c) *foo* está mal tipada.
(d) El tipo de *foo* es $T\ Bool\ (Bool, (Bool, Bool)) \to (Bool, Bool)$.

Respuesta: a)

6. Dada la siguente definición:

$$\begin{aligned} foldn\ f\ g\ [x] &= g\ x \\ foldn\ f\ g\ (x : xs) &= f\ x\ (foldn\ f\ g\ xs) \end{aligned}$$

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

(a) $foldn\ (:)\ (:[\,])\ xs$ es equivalente a $foldr\ (:)\ [\,]\ xs$, para cualquier lista finita xs
(b) $foldn\ (+)\ (flip\ const\ True)\ [1, 2, 3]$ compila correctamente
(c) $foldn\ (+)\ (const\ 0)\ [\,]$ no compila
(d) El tipo más general de *foldn* es $(a \to b \to b) \to (a \to b) \to [b] \to a$

Respuesta: b)

7. Dada la siguiente definición:

$$foldC\ f\ g\ e = foldr\ (f\ .\ g)\ e$$

¿Cuál de las siguientes afirmaciones es **incorrecta**? Suponga xs finita del tipo apropiado.

(a) Para g del tipo apropiado, $length\ (foldC\ (:)\ g\ [\,]\ xs) == length\ xs$
(b) $foldC\ (flip\ const)\ id\ [\,]\ xs == xs$
(c) $foldC\ (+)\ (+1)\ 0\ xs == sum\ xs + length\ xs$
(d) Para f y g del tipo apropiado, $foldC\ f\ g\ e\ xs == foldr\ f\ e\ (map\ g\ xs)$

Respuesta: b)

8. Se desea implementar funciones para convertir entre las notaciones decimal y binaria de números enteros no negativos. La representación binaria de un número será dada por una lista de enteros donde se asume que todos tienen valor 0 o 1.

   type $Bin = [Int]$

   Los digitos binarios están dados en orden decreciente de significación. Por ejemplo, la lista $[1, 1, 0]$ representa el 6. El número cero es representado por una lista de ceros.

   (a) Escribir una función $int2bin :: Int \to Bin$ que convierta un entero en notación decimal a un número binario.

```
int2bin :: Int → Bin
int2bin 0 = [0]
int2bin n = i2b [] n
  where i2b ds 0 = ds
        i2b ds n = i2b (n `mod` 2 : ds) (n `div` 2)
```

   (b) Implementar **como** un *foldl* una función $bin2int :: Bin \to Int$ que convierte un número binario en un entero en notación decimal. Se debe realizar una **única** recorrida sobre la lista de bits.

```
bin2int :: Bin → Int
bin2int = foldl (λn b → n * 2 + b) 0
```

9. Dadas las siguientes definiciones:

$$\textbf{data}\ Tree\ a = Leaf\ a \mid Fork\ (Tree\ a)\ (Tree\ a)$$
$$tree = Fork\ tree\ (Fork\ (Leaf\ 2)\ tree)$$

$$\begin{aligned}
&der\ (Leaf\ x) &&= [\,]\\
&der\ (Fork\ l\ r) &&= r : izq\ r\\
&izq\ (Leaf\ x) &&= [\,]\\
&izq\ (Fork\ l\ r) &&= l : der\ r\\
&val\ (Leaf\ x) &&= x\\
&val\ (Fork\ l\ (Leaf\ x)) &&= x\\
&val\ (Fork\ l\ r) &&= val\ r
\end{aligned}$$

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

(a) $(map\ val\ .\ der\ \$\ tree) == [\,]$

$\boxed{False}$

(b) $length\ .\ izq\ \$\ tree$

diverge

(c) $val\ .\ head\ .\ tail\ .\ izq\ \$\ tree$

diverge

(d) $filter\ (>2)\ .\ map\ val\ .\ der\ \$\ tree)$

diverge

(e) $length\ .\ take\ 5\ .\ izq\ \$\ tree$

$\boxed{5}$

(f) $val\ .\ (!!1)\ .\ der\ \$\ tree$

$\boxed{2}$

(g) $val\ tree == 2$

diverge

(h) $head\ .\ tail\ .\ der\ \$\ tree$

$\boxed{Leaf\ 2}$