E.T.S.I. Informática. Universidad de Málaga — Apellidos: **SOLUCIONES**

Examen Ordinario, Febrero de 2004

Programación Declarativa (tercer curso) — Nombre: _____

Puntuaciones:

1	2	3	4	5	6	7	total
1	1	1	2	1	2	2	10.0

☐ si ☐ no } deseo que se publique mi calificación si fuera negativa

---

Consideremos la siguiente estructura algebraica e inductiva para los enteros:

$$\begin{array}{lll} data\ E\ =\ O & & \text{— la letra } O\text{, que representa el cero} \\ \quad \mid S\ E \mid P\ E & & \text{— sucesor y predecesor} \end{array}$$

Por ejemplo, el entero 2 se representa como $S(S\,O)$ o también como $S(P(S(S\,O)))$, mientras que -2 se representa como $P(P\,O)$. Sea ahora la siguiente función de plegado de enteros:

$$\begin{array}{lll} pliega\ f\ g\ z\ O & = & z \\ pliega\ f\ g\ z\ (S\ x) & = & f\ (pliega\ f\ g\ z\ x) \\ pliega\ f\ g\ z\ (P\ x) & = & g\ (pliega\ f\ g\ z\ x) \end{array}$$

---

1 Describe su tipo de forma razonada: $pliega\ ::\ (a \to a) \to (a \to a) \to a \to E \to a$

Según la 1ª ecuación, el tipo del valor devuelto (por ejemplo a) coincide con el del 3º argumento. A la vista de la 2ª ecuación, el 1º y 2º argumentos son funciones de tipo $? \to a$; por tanto el tipo de pliega toma el aspecto $(? \to a) \to (? \to a) \to a \to E \to a$. Pero a la vista de la 2ª ecuación, el tipo del argumento $(pliega\ f\ g\ z\ x)$ de la función 2º argumento debe ser también a, e igual para la función 1º argumento.

---

2 Utilizando la función anterior, escribe la función $aInteger$ que transforma un dato de tipo E en un entero estándar (por ejemplo $aInteger(P(P\,O)) \Rrightarrow -2$):

$$\begin{array}{l} aInteger\ ::\ E \to Integer \\ aInteger\ =\ pliega\ (+1)\ (+\ -1)\ 0 \end{array}$$

---

3 Las dos siguientes funciones comprueban la paridad de un dato de tipo E:

$$\begin{array}{l} par, par'\ ::\ E \to Bool \\ par\ =\ pliega\ not\ not\ True \end{array} \qquad \begin{array}{lll} par'\ O & = & True \\ par'\ (S\ x) & = & not\ (par'\ x) \\ par'\ (P\ x) & = & not\ (par'\ x) \end{array}$$

Define, utilizando solamente *pliega*, la función:

$$\begin{array}{l} impar\ ::\ E \to Bool \quad \text{— comprueba si un dato de tipo } E \text{ es impar} \\ impar\ =\ pliega\ not\ not\ False \end{array}$$

---

4 Demuestra que $par = par'$, utilizando como técnica, inducción estructural sobre E.

Hay que demostrar: $\forall\ e\ .\ e :: E\ .\ pliega\ not\ not\ True\ =\ par'\ e$.

CASO BASE:

$pliega\ not\ not\ True = par'\ 0$

$=$! 1)$pliega$, 1)par'

$True = True$

PASOS INDUCTIVOS (complete solamente uno):

$pliega\ not\ not\ True\ (S\ e) = par'\ (S\ e)$

$=$! 2)$pliega$, 2)par'

$not(pliega\ not\ not\ True\ e) = not(par'e)$

$\Leftarrow$! sustitutividad

$pliega\ not\ not\ True\ e = par'e$

$\Leftarrow$

HI

Sea la sucesión b_n definida por la recurrencia:

$$b_0 = 1, \qquad b_1 = -1, \qquad b_2 = 1, \qquad b_{n+3} = b_{n+2} + b_n, \text{ para } n \geq 0.$$

5 Escribe *directamente* una función para su cómputo:

$$\begin{array}{ll} b & :: \mathit{Integer} \rightarrow \mathit{Integer} \\ b\ 0 & = 1 \\ b\ 1 & = -1 \\ b\ 2 & = 1 \\ b\ (n+3) & = b\ (n+2) + b\ n \end{array}$$

y justifica que el cómputo de b_n es de orden a^n, para cierto valor de a.

En efecto: Si buscamos que el número de sumas s_n que realiza la llamada $b\ n$ satisfaga $s_n \geq a^{n-2} - 1$, para los casos base basta tomar $a \geq 1$, y el paso inductivo es:

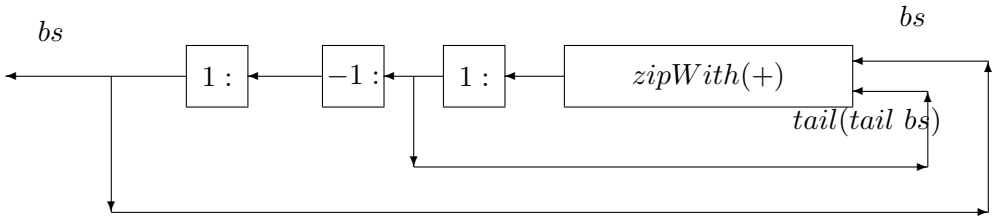
$$(s_{n+3} \geq a^{n+1}-1) \overset{4^{a}ec.}{\Leftarrow} (s_{n+2}+s_n+1 \geq a^{n+1}-1) \overset{HI}{\Leftarrow} (a^n-1+a^{n-2}-1+1 \geq a^{n+1}-1) \Leftarrow a^n+a^{n-2} \geq a^{n+1} \Leftarrow a^2+1 \geq a^3$$

El mejor valor es la raíz positiva de la ecuación $a^2 + 1 = a^3$, $a = 1, 4...$

---

6 Como alternativa a la definición anterior, utiliza la técnica de *redes de procesos* para computar la lista infinita $[b_0, b_1, \dots]$ de los elementos de la sucesión $\{b_n\}$. (Describa un dibujo con la red de procesos, así como sus ecuaciones en Haskell).

$$bs\ =\ 1\ :\ -1\ :\ 1\ :\ \mathit{zipWith}\ (+)\ bs\ (\mathit{tail}\ (\mathit{tail}\ bs))$$



La complejidad en este caso es lineal ya que no se recomputan los términos necesarios para calcular $b(n+3)$, ya que han sido calculados y están disponibles en el flujo.

---

7 Sea la siguiente función para calcular la mediana de tres datos:

$$\begin{array}{ll} \mathit{md} & :: (\mathit{Int}, \mathit{Int}, \mathit{Int}) \rightarrow \mathit{Int} \\ \mathit{md}\ (x,y,z) & \mid x < y \quad\ = \mathit{md}(y,x,z) \\ & \mid y < z \quad\ = \mathit{md}(x,z,y) \\ & \mid \mathit{otherwise} = y \end{array}$$

Pruebe, utilizando un razonamiento basado en conjuntos bien construidos, que la expresión $md(A, B, C)$ calcula la mediana de la terna (A, B, C):

1. La llamada $md(A, B, C)$ termina ya que la sucesión de ternas que aparecen como argumento en las llamadas generadas forman una sucesión decreciente para el orden lexicográfico inverso en el conjunto finito (y por tanto inductivo) $\mathcal{C} \equiv \mathcal{P}\mathit{ermutaciones}(A, B, C)$.

2. Si $md(A, B, C)$ termina, entonces computa la mediana de la terna (A, B, C). En efecto: Si termina es porque en la última llamada fallan las dos primeras guardas, luego, si ésta es $mcd(x, y, z)$ entonces y es la mediana. Por otro lado, las ternas en cada llamada verifican $(x, y, z) \in \mathcal{C}$. Luego $md(x, y, z) = \mathit{mediana}(A, B, C)$.