E.T.S.I. Informática. Universidad de Málaga
Examen Extraordinario, Setiembre de 2004
Programación Declarativa (tercer curso)

APELLIDOS: _____

NOMBRE: _____

PUNTUACIONES:	1	2	3	4	5	6	total
	1	*1*	*2*	*2*	*2*	*2*	*10.0*

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

Sea la siguiente estructura para representar colas:

$$data\ Cola\ a\ =\ V\ |\ Cola\ a\ :>\ a \qquad deriving\ (Show, Eq)$$

donde los elementos se *encolan/extraen* directamente "al/desde el" final de la cola a través de tres funciones:

$$\begin{array}{l} encola\ ::\ a\ \to\ Cola\ a\ \to\ Cola\ a \\ primeroDe\ ::\ Cola\ a\ \to\ a \\ restoDe\ ::\ Cola\ a\ \to\ Cola\ a \end{array}$$

de forma que:

$$\begin{array}{lll} restoDe\ (V\ :>\ 3\ :>\ 4\ :>\ 5) & \rightsquigarrow & V\ :>\ 3\ :>\ 4 \\ primeroDe\ (V\ :>\ 3\ :>\ 4\ :>\ 5) & \rightsquigarrow & 5 \\ encola\ 1\ (V\ :>\ 3\ :>\ 4\ :>\ 5) & \rightsquigarrow & V\ :>\ 1\ :>\ 3\ :>\ 4\ :>\ 5 \end{array}$$

1 Defina las tres funciones anteriores:
SOLUCION

$$\begin{array}{l} restoDe\ (c\ :>\ x)\ =\ c \\ primeroDe\ (c\ :>\ x)\ =\ x \\ encola\ x\ V\ =\ V\ :>\ x \\ encola\ x\ (c\ :>\ y)\ =\ encola\ x\ c\ :>\ y \end{array}$$

— no es necesario parentizar $(encola\ x\ c)\ :>\ y$

2 Consideremos ahora la siguiente función de plegado de colas:

$$\begin{array}{ll} pliegaCola\ f\ z\ V & =\ z \\ pliegaCola\ f\ z\ (\ c\ :>\ x) & =\ f\ (pliegaCola\ f\ z\ c)\ x \end{array}$$

Infiera de forma razonada el tipo de la función *pliegaCola*.
SOLUCION Según la 1ª ecuación, el tipo del valor devuelto (por ejemplo a) coincide con el del 2º argumento. A la vista de la 2ª ecuación, el tercer argumento es una cola (con tipo $Cola\ b$), y el primer argumento es una función de dos argumentos, con tipo $? \to ? \to a$; por tanto el tipo de *pliegaCola* toma el aspecto $(? \to ? \to a) \to a \to Cola\ b \to a$. Pero a la vista de la 2ª ecuación, el tipo del argumento $(pliegaCola\ f\ z\ c)$ (que coincide con el primer argumento de $? \to ? \to a$) debe ser también a, mientras que el tipo de x debe coincidir con el tipo base de $Cola\ b$. Luego

$$pliegaCola\ ::\ (a\ \to\ b\ \to\ a)\ \to\ a\ \to\ Cola\ b\ \to\ a$$

---

3 Defina las siguientes funciones

$$\begin{array}{l} listaAcola\ ::\ [a]\ \to\ Cola\ a \\ colaAlista\ ::\ Cola\ a\ \to\ [a] \\ invierteLista\ ::\ [a]\ \to\ [a] \end{array}$$

De forma que tengan los siguientes comportamientos:

$$\begin{array}{lll} listaAcola\ [1,2,3] & \rightsquigarrow & ((V\ :>\ 1)\ :>\ 2)\ :>\ 3 \quad \text{— visita en post-orden} \\ colaAlista\ (V\ :>\ 1\ :>\ 2\ :>\ 3) & \rightsquigarrow & [3,2,1] \quad \text{— visita en orden} \\ invierteLista\ [1,2,3] & \rightsquigarrow & [3,2,1] \end{array}$$

Complete sus definiciones
SOLUCION

$$
\begin{array}{lcl}
listaAcola & = & foldl\ (:>)\ V \\
colaAlista & = & pliegaCola\ (flip\ (:))\ [] \\
invierteLista & = & colaAlista\ .\ listaAcola
\end{array}
$$

---

4 Demuestra por inducción sobre colas que se verifica $\forall\ c\ \centerdot\ c :: Cola\ a\ \centerdot\ longitudCola\ c\ \geq\ 0$, donde

$$
\begin{array}{lcl}
longitudCola & :: & Cola\ a\ \rightarrow\ Integer \\
longitudCola & = & pliegaCola\ (const\ .\ (1+))\ 0
\end{array}
$$

Demostraremos en forma equivalente: $\forall\ c\ \centerdot\ c :: Cola\ a\ \centerdot\ pliegaCola\ (const\ .\ (1+))\ 0\ c\ \geq\ 0$:

CASO BASE:

$pliegaCola\ (const.(1+))\ 0\ c \geq 0$
$=$! 1)*pliegaCola*
$c \geqslant 0$
Cierto

PASO INDUCTIVO:

$pliegaCola\ (const.(1+))\ 0\ (c :> x) \geq 0$
$=$! 2)*pliegaCola*
$(const.(1+))\ (pliegaCola\ (const.(1+))\ 0\ c))\ x \geq 0$
$=$! def. composición
$const((1+)\ (pliegaCola\ (const.(1+))\ 0\ c))\ x \geq 0$
$=$! def *const*
$(1+)\ (pliegaCola\ (const.(1+))\ 0\ c) \geq 0$
$=$! hipótesis de inducción
Cierto

---

5 Utilizando un razonamiento basado en conjuntos bien construidos, pruebe que la expresión $seg(A, B, C, D)$ calcula el segundo menor elemento de la tupla (A, B, C, D), es decir,

$$seg(A, B, C, D)\ =\ mínimo\ (\ \{A, B, C, D\}\ \setminus\ mínimo\{A, B, C, D\})$$

siendo

$$
\begin{array}{lcl}
seg\ (x, y, z, t) & & \\
\quad |\ x > y & = & seg\ (y, x, z, t) \\
\quad |\ y > z & = & seg\ (x, z, y, t) \\
\quad |\ y > t & = & seg\ (x, t, z, y) \\
\quad |\ otherwise & = & y
\end{array}
$$

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

2. *Si $seg(A, B, C, D)$ termina, entonces computa el segundo menor de la terna (A,B,C,D), ya que ...*, si termina es porque en la última llamada fallan las tres primeras guardas, luego, si la última llamada es $seg(x, y, z, t)$, entonces se verifica $x \leq y \leq z, t$, de donde y es el segundo menor de la tupla (x, y, z, t). Por otro lado, ya que solamente se permutan elementos, las tuplas en cada llamada verifican $(x, y, z, t) \in \mathcal{C}$. Luego $seg(x, y, z, t) =$ *segundo menor de* (A, B, C, D).

---

6 Describa una red de procesos y sus ecuaciones Haskell para computar la lista de factoriales.
SOLUCIÓN Vea la pág. 212 de nuestro texto *Razonando con Haskell.*