E.T.S.I. Informática. Universidad de Málaga
Primer Parcial, 9 de Noviembre de 2005
Programación Declarativa, Prog. Funcional (3º)

APELLIDOS: **SOLUCIONES**

NOMBRE:

	1	2	3	4	5	6	7	8	9	total
PUNTOS	0.5	1.5	0.5	0.5	1.0	1.0	1.0	2.0	2.0	10.0

☐ si
☐ no
} publique mi calificación si fuera negativa

$\boxed{\mathbf{1}}$ Infiere el tipo más general de la función de ecuación: $const\ k\ x\ =\ k$.

SOL Si asignamos tipos a los argumentos ($k :: \alpha,\ x : \beta$) obtenemos directamente $const\ ::\ \alpha \rightarrow \beta \rightarrow \alpha$

Consideremos la siguiente función-esquema para representar iteradores sobre los números naturales:

$$iter\ f\ e\ O = e$$
$$iter\ f\ e\ m@(n+1) = f\ m\ (iter\ f\ e\ n)$$

$\boxed{\mathbf{2}}$ Describe su tipo de forma razonada:

SOL $iter :: (Integer \rightarrow a \rightarrow a) \rightarrow a \rightarrow Integer \rightarrow a$. En efecto. Asignemos tipos a los argumentos: $f :: \alpha,\ e :: \beta$ (el tercer argumento claramente es de tipo $Integer$).

Ya que el resultado es del mismo tipo que el segundo argumento, tendremos:

$$iter :: \alpha \rightarrow \beta \rightarrow Integer \rightarrow \beta$$

Por la segunda ecuación $f\ m\ (iter\ f\ e\ n) :: \beta$, de donde, f tiene dos argumentos con tipos $Integer$ y β, y por tanto $f :: Integer \rightarrow \beta \rightarrow \beta$, y éste tipo debe coincidir con α, de donde:

$$iter :: (Integer \rightarrow \beta \rightarrow \beta) \rightarrow \beta \rightarrow Integer \rightarrow \beta$$

$\boxed{\mathbf{3}}$ Sea la función

$$mágica :: Integer \rightarrow Bool$$
$$mágica = iter\ (const\ not)\ True$$

Evalúa los valores $const\ not\ 1007\ True$, $mágica\ 0$, $mágica\ 1$, ...

SOL

$const\ not\ 1007\ True$
$\equiv \because$ asociatividad a izquierda de la aplicación
$(const\ not\ 1007)\ True$
$\rightarrow \because$ def. de $const$
$not\ True$
$\rightarrow \because$ def. de not
$False$

$mágica\ 0$
$\equiv \because$ def. de $mágica$
$iter\ (const\ not)\ True\ 0$
$\rightarrow \because$ primera ecuación de $iter$
$True$

$mágica\ 1$
$\equiv \because$ def. de $mágica$
$iter\ (const\ not)\ True\ 1$
$\rightarrow \because$ segunda ecuación de $iter$
$(const\ not)\ 1\ (iter\ (const\ not)\ True\ 0)$
$\rightarrow \because$ asociatividad
$(const\ not\ 1)\ (iter\ (const\ not)\ True\ 0)$
$\rightarrow \because$ def. de $const$
$not\ (iter\ (const\ not)\ True\ 0)$
$\rightarrow \because$ primera ecuación de $iter$
$not\ True$
$\rightarrow \because$ def. de not
$False$

$\boxed{\mathbf{4}}$ ¿Que interesante valor computa la función $mágica$?

SOL Siguiendo evaluando observamos: $mágica\ 2 \rightarrow \ldots True$, $mágica\ 3 \rightarrow \ldots False$, de donde $mágica$ comprueba si un $Integer$ es **PAR**.

$\boxed{\mathbf{5}}$ Usando únicamente $iter$, escribe una función para comprobar si un número natural representado con un Integer es impar.

SOL $impar = iter\ (const\ not)\ False$

---

6 Escribe una función para calcular el factorial usando únicamente *iter*

$$factorial :: Integer \rightarrow Integer$$

SOL $factorial \ = \ iter\ (*)\ 1$

---

7 Escribe una función para calcular el producto de dos naturales usando únicamente *iter*

$$producto :: Integer \rightarrow Integer \rightarrow Integer$$

SOL $producto\ x \ = \ iter\ (const\ (x+))\ 0$

---

8 Pretendemos demostrar que la siguiente función calcula la **mediana** de cinco números enteros:

$$\begin{array}{ll} md :: (Integer, Integer, Integer, Integer, Integer) \rightarrow Integer & \\ md\ (x,y,z,t,u) & \mid x > z \quad = md(z,y,x,t,u) \\ & \mid y > z \quad = md(x,z,y,t,u) \\ & \mid z > t \quad = md(x,y,t,z,u) \\ & \mid z > u \quad = md(x,y,u,t,z) \\ & \mid otherwise = z \end{array}$$

Utilizando un razonamiento basado en conjuntos inductivos o bien construidos, prueba que la expresión $md(A,B,C,D,E)$ calcula en efecto la mediana de la tupla de enteros (A,B,C,D,E) a partir del siguiente esquema:

SOL

1. *La llamada $md(A,B,C,D,E)$ termina ya que* los sucesivos argumentos en la llamada a md pertenecen al conjunto $\mathcal{C}$ de permutaciones de los valores iniciales (A,B,C,D,E). Este conjunto $\mathcal{C}$ es inductivo por ser finito (tiene 5! elementos). Además, para el orden lexicográfico, las tuplas argumentos de las sucesivas llamadas a md forman una cadena descendente para el orden lexicográfico. Por ejemplo, si $x > z$, por la primera guarda, pasamos de la tupla (x,y,z,t,u) a la tupla (z,y,x,t,u) que es menor para el orden lexicográfico. En definitiva, esta sucesión de tuplas debe ser finita, y la llamada termina en a lo sumo 5! pasos.

2. *Si $md(A,B,C,D,E)$ termina, entonces computa la mediana de la tupla (A,B,C,D,E).* En efecto: Al terminar es porque en la última llamada para su argumento $(x,y,z,,t,u)$ fallan todas las guardas, de donde $x,y \leq z \leq t,u$. Es decir, z es la mediana, que en efecto es lo que devuelve según la expresión con guarda *otherwise*. Por otro lado, cada tupla es una permutación de (A,B,C,D,E), luego la tupla de la última llamada también es una permutación de los valores iniciales, que tendrán la misma mediana.

3. *Prueba con un ejemplo que tal función no ordena la tupla inicial.* Un ejemplo trivial es la llamada $md(5,4,8,12,10)$, que termina sin ordenar la tupla ya que fallan todas las guardas.

---

9 Dado un conjunto de enteros $\mathcal{C}$, se define el segundo menor en la forma $seg\,\mathcal{C} \doteq \text{mín}(\mathcal{C} \setminus \text{mín}\,\mathcal{C})$. Por ejemplo $seg\{1,5,2,7\} \doteq 2$. Escribe la función seg para calcular el segundo menor de un conjunto de cinco enteros dados por una tupla (A,B,C,D,E): $seg :: (Integer, Integer, Integer, Integer, Integer) \rightarrow Integer$

SOL

$$\begin{array}{ll} seg\ (x,y,z,t,u) & \mid x > y \quad = sg(y,x,z,t,u) \\ & \mid y > z \quad = sg(x,z,y,t,u) \\ & \mid y > t \quad = sg(x,t,z,y,u) \\ & \mid y > u \quad = sg(x,u,z,t,y) \\ & \mid otherwise = y \end{array}$$

Obsérvese que en la última llamada fallan todas las guardas para la tupla (x,y,z,t,u), de donde $x \leq y \leq z,t,u$. Es decir, y es el segundo menor de la tupla. La corrección sigue vía un razonamiento parecido al del ejercicio anterior.