

INFORMATICA.

EXAMEN PARCIAL Febrero 2004.

Haskell

Apellidos, Nombre

Grupo

NOTA: Indica también el tipo de todas las funciones que definas en los distintos ejercicios.

1) Sean los siguientes tipos para representar matrices como listas de vectores (uno por cada fila de la matriz):

```
type Vector = [ Float ]  
type Matriz = [ Vector ]
```

```
m :: Matriz  
m = [ [1,2,3]  
      , [4,5,6]  
      , [7,8,9] ]
```

a) (1 punto) Define recursivamente una función `diagonal` que tome una matriz cuadrada y devuelva su diagonal principal como un `Vector`. Por ejemplo:

```
diagonal m ⇒ [1,5,9]
```

b) (1.5 puntos) Define la función `diagonal` de modo no recursivo usando las funciones predefinidas `zip` y `map` y el operador de composición `(.)`.

c) (1.5 puntos) Define, usando la función predefinida `iterate`, la función `identidad` que devuelva una matriz de infinitas filas, donde cada fila es a su vez un vector infinito que contiene un uno en la posición correspondiente a la diagonal principal y un cero en las demás posiciones, es decir,

```
identidad ⇒ 
$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

```

d) (1 punto) Define, usando la función `identidad`, la función `identidadN` de modo que `identidadN n` devuelva la matriz `identidad` de dimensión `n`. Por ejemplo:

```
identidadN 3 ⇒ 
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```

e) (1 punto) Define la función `esDiagonal` que tome una matriz cuadrada y compruebe si es diagonal, es decir, si todos los elementos que no están en la diagonal principal son nulos. Por ejemplo:

```
[ [1,0,0]  
esDiagonal , [0,3,0] ⇒ True  
            , [0,0,8] ]
```

2) (1.5 pts) Define, usando listas por comprensión, la función `distribuye` que compruebe si una función distribuye sobre un operador para todos los elementos de un conjunto. Por ejemplo, si

`doble x = 2*x,`

entonces

`distribuye doble (+) [1..10] ⇒ True`

ya que la función `doble` distribuye con la suma para el conjunto `[1..10]`:

$\forall x, y \in [1..10] . \text{doble } x + \text{doble } y = \text{doble}(x + y)$

La función, el operador y el conjunto se pasarán como parámetros. Indica además el tipo polimórfico de dicha función.

3)

a) (1 punto) Define usando `foldr` una función que tome una lista de enteros y compruebe si todos sus elementos son pares.

b) (1.5 puntos) Define usando `foldl` la función `reverseMap`, con el siguiente comportamiento:

`reverseMap f [x1, x2, ..., xn] ⇒ [f xn, f xn-1, ..., f x1]`