Programación declarativa (Grupo 1) (22–Enero–2010)

Apellidos:

Nombre:

Observaciones:

1. *En la evaluación se tendrá en cuenta la corrección, simplicidad y eficiencia de las respuestas.*
2. *Hay que describir las definiciones auxiliares (menos las del sistema).*

Ejercicio 1 (2 puntos) Definir la función

```
palabrasDeLongitud :: Int -> [Char] -> [String]
```

tal que (`palabrasDeLongitud n cs`) es la lista de las cadenas de longitud `n` formadas con los caracteres de `cs`. Por ejemplo,

```
*Main> palabrasDeLongitud 3 "ab"
["aaa","aab","aba","abb","baa","bab","bba","bbb"]
```

Solución:

```
palabrasDeLongitud 0 cs = [[]]
palabrasDeLongitud n cs =
    [x:xs | x <- cs,
            xs <- palabrasDeLongitud (n-1) cs]
```

Ejercicio 2 (2 puntos) Definir la función

```
intercala :: a -> [a] -> [a]
```

tal que (`intercala x ys`) es la lista obtenida intercalando `x` entre los elementos de `ys`. Por ejemplo,

```
intercala ',' "abcde" ==> "a,b,c,d,e"
```

Solución:

```
intercala _   []   = []
intercala _   [y]  = [y]
intercala x (y:ys) = y : x : intercala x ys
```

Ejercicio 3 (2 puntos) Los grafos pueden representarse mediante listas de arcos, donde cada arco se representa mediante un par de nodos. Definir la función

```
rutas :: Eq a => [(a,a)] -> a -> a -> [[a]]
```

tal que (`rutas g x y`) es la lista de las rutas desde `x` hasta `y` en el grafo g. Por ejemplo,

```
*Main> rutas [(1,2),(1,3),(2,4),(3,5),(5,6),(3,6)] 1 6
[[1,3,5,6],[1,3,6]]
```

Solución:

```
rutas g x y = rutasAux x y []
    where rutasAux x y vis
              | x == y    = [x:vis]
              | otherwise = concat [rutasAux x y' (y:vis)
                                   | (y',y'') <- g,
                                     y'' == y,
                                     notElem y' vis]
```

Ejercicio 4 (2 puntos) Usando la definición

```
f []     ys = ys              -- f1
f (x:xs) ys = x : (f xs ys)   -- f2
```

demostrar por inducción que para todo xs, ys, zs se verifica que

```
f xs (f ys zs) = f (f xs ys) zs
```

Solución: Demostración por inducción en xs:

- Caso base xs=[]: Reduciendo el lado izquierdo

  ```
  f xs (f ys zs)
  = f [] (f ys zs)       [por hipótesis]
  = f ys zs              [por f.1]
  ```

  y reduciendo el lado derecho

  ```
  f (f xs ys) zs
  = f (f [] ys) zs       [por hipótesis]
  = f ys zs              [por f.1]
  ```

  Luego, `f xs (f ys zs) = f (f xs ys) zs`

- Caso inductivo xs=a:as: Suponiendo la hipótesis de inducción

  ```
  f as (f ys zs) = f (f as ys) zs
  ```

  hay que demostrar que

  ```
  f (a:as) (f ys zs) = f (f (a:as) ys) zs
  ```

  ```
  f (a:as) (f ys zs)
  = a:(f as (f ys zs))       [por f.2]
  = a:(f (f as ys) zs)       [por hip. ind.]
  = f (a:(f as  ys)) zs      [por f.2]
  = f (f (a:as) ys) zs       [por f.2]
  ```

Ejercicio 5 (2 puntos) Se realizó el miércoles 20 de enero en el aula de laboratorio.