Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información.
CI–3661 – Laboratorio de Lenguajes de Programación.
Septiembre–Diciembre 2012

Problemario de Semana II: Haskell – Tipos y Funciones

1. Para cada una de las siguiente expresiones en Haskell, diga cual es su tipo:

   a) `10 * 3 + 20 / 2 + 2`
   b) `"Hello World!"`
   c) `(3 > 4, 3 + 4)`
   d) `(1, 2, 3)`
   e) `(1, (2, 3))`
   f) `[1, 2, 3]`
   g) `[1, 3..]`
   h) `[(x, y) | x <- [1, 2, 3], y <- [True, False]]`
   i) `(+ 2)`
   j) `(< 2)`
   k) `(== 2)`
   l) `(+ 2) . (3 +)`
   m) `\x -> x * x - 1`
   n) `\x -> \y -> x + y`
   ñ) `\x y -> x + y`
   o) `\x y -> x y`
   p) `\x -> x . x`
   q) `(\x -> x . x) (+ 3)`
   r) `let f 0 = 1 ; f n = n * f (n - 1) in f`
   s) `let f n | n == 0 = 1 | otherwise = n * f (n - 1) in f`
   t) `let f y = y (f y) in f`

2. Para cada una de las siguiente funciones en Haskell, diga cual es su firma:

   a) `sum3 x y z = x + y + z`
   b)
   ```
   isTheAnswer 42 = True
   isTheAnswer _  = False
   ```
   c)
   ```
   zigZag _ [] = []
   zigZaz True  (x:xs) =  x : zigZag False xs
   zigZaz False (x:xs) = -x : zigZag True xs
   ```
   d)
   ```
   addOne = myMap (+1)
     where myMap _ []     = []
           myMap f (x:xs) = f x : myMap xs
   ```
   e) `myCurry f x = \y -> f (x, y)`

3. Evalúe cada una de las siguientes expresiones en Haskell:

a)
```
(\x -> x . x) (+ 3) 6
```

b)
```
take 10 ((\x -> [x, x + x..]) 3)
```

c)
```
[(x, y) | x <- [1..3], y <- [1..x]]
```

d)
```
let myAnd False _ = False
    myAnd True  x = x
in myAnd False (length (repeat 0) == 0)
```

e)
```
let myAnd False _ = False
    myAnd True  x = x
in myAnd True (length (repeat 0) == 0)
```

f)
```
let listAdd []     _      = []
    listAdd _      []     = []
    listAdd (x:xs) (y:ys) = x + y : listAdd xs ys
    f = 0 : 1 : listAdd f (tail f)
in take 10 f
```

g)
```
let listProd []     _      = []
    listProd _      []     = []
    listProd (x:xs) (y:ys) = x * y : listProd xs ys
in sum (listProd [1, 2, 3] [4, 5, 6])
```