Estructuras Discretas

Introducción a Haskell

Repaso de tipos

- Números enteros (Int).
- Números con punto flotante (Float).
- Booleanos (Bool).
- Caracteres (Char).
- Listas
- Tuplas.
- Cadenas de texto (String / [Char]).
- Funciones (*a* -> *a*, donde *“a”* es un tipo de los anteriores mencionados)

Tipos: Listas

[*a*] es el tipo de una lista de elementos de tipo *a*.

```
ghci> :t ['a', 'b']
['a', 'b'] :: [Char]
ghci> :t ["uno", "dos"]
["uno", "dos"] :: [String]
ghci> :t [True]
[True] :: [Bool]
```

Variables de tipo

Una variable de tipo es muy parecida a los genéricos de otros lenguajes, ya que indica que, por ejemplo, una función puede recibir una lista que contenga elementos de cualquier tipo, como pasa con la función *head:*

```
ghci> :t head
head :: [a] -> a
```

Si un tipo contiene una variable de tipo, se dice que este es polimórfico y una función es polimórfica si su tipo es polimórfico.

Tipos: Funciones

Una función es una aplicación que se hace a valores de cierto tipo obteniendo como resultado un valor de otro tipo. Entonces se tiene que $a \longrightarrow b$ es el tipo de la función que se aplica a valores del tipo *a* y retorna valores del tipo *b*.

```
ghci> :t (++)
(++) :: [a] -> [a] -> [a]
ghci> :t not
not :: Bool -> Bool
```

Inferencia de tipos

La inferencia de tipos es, en resumen, un sistema el cual puede deducir de forma inteligente que tipo de dato corresponde a un trozo de código dado y sigue la siguiente regla:

$$\frac{f :: A \rightarrow B \qquad e :: A}{f\ e :: B}$$

```
ghci> a = True
ghci> :t a
a :: Bool
ghci> :t not
not :: Bool -> Bool
ghci> :t not a
not a :: Bool
```

Clases de tipos

Una clase de tipos es, por ejemplo, como una interfaz del lenguaje de programación Java, o en otras palabras es un mecanismo para definir cierto tipo de comportamiento, es decir, si un tipo forma parte de una clase de tipos, este tipo cumple con el comportamiento que define la clase de tipos.

```
(==) :: Eq a => a -> a -> Bool
ghci> :t mod
mod :: Integral a => a -> a -> a
```