Programación Funcional

Instituto de Computación, Facultad de Ingeniería
Universidad de la República, Uruguay

Tipos Estructurados

Tipos Estructurados

- Además de los tipos básicos, podemos definir **tipos estructurados** compuestos por un conjunto de valores.

- Veremos:
  - Tuplas
  - Listas
  - Tipos definidos por el usuario

Tuplas

- Una **tupla** combina en un mismo objeto a un número fijo de valores, cada uno con un tipo determinado (posiblemente distintos).

- Similar a Registros o Estructuras en otros lenguajes.

- Ejemplos
  - Coordenadas cartesianas:
    $(8, 23) :: (Int, Int)$
  - Empleados (nombre, edad, sueldo):
    $(\texttt{"Juan"}, 23, 60000) :: (String, Int, Int)$

- En general, el tipo $(t_1, ..., t_n)$ consiste en tuplas de valores: $(v_1, ..., v_n)$ donde $v_1 :: t_1$, ..., $v_n :: t_n$

Funciones sobre Tuplas

- Se suele usar **pattern matching**:

$$nombre :: (String, Int, Int) \to String$$
$$nombre\ (n, e, s) = n$$

- Al aplicarse, los componentes del patrón se matchean con los componentes del argumento.
$nombre\ (\texttt{"Pedro"}, 44, 60000) \rightsquigarrow \texttt{"Pedro"}$

- Se pueden ignorar componentes.

$$edad\ \ (\_, e, \_) = e$$
$$sueldo\ (\_, \_, s) = s$$

- Se pueden forzar componentes usando literales.

$$esPedro\ (\texttt{"Pedro"}, \_, \_) = True$$
$$esPedro\ (\_, \quad \_, \_) = False$$

Sinónimos de Tipos

- En Haskell podemos dar nombres a los tipos.
- Al usar un **sinónimo** se tiene el mismo efecto que al usar el tipo que representa.
- Ejemplos:
  - type $Coord = (Int, Int)$
  - type $Empleado = (String, Int, Int)$

Preguntas

Listas

- Una **lista** combina en un mismo objeto a un número arbitrario de valores, todos de un mismo tipo.
- Ejemplos
  - $[8, 2, 3, 1, 24] :: [Int]$
  - $[(\texttt{"Juan"}, 23, 60000), (\texttt{"Pedro"}, 44, 60000)] :: [Empleado]$
- En general, el tipo $[t]$ consiste en listas de valores: $[v_1, ..., v_n]$ donde $v_1 :: t$, ..., $v_n :: t$
- Definimos inductivamente a las listas de tipo $[t]$ como:
  - $[]$ - lista vacía
  - $v : vs$ - lista con al menos un elemento, donde $v :: t$ y $vs :: [t]$

Funciones sobre Listas

- Existe una gran cantidad de funciones sobre listas en *Prelude*
- Ejemplos: $(+\!\!+)$, (!!), *length*, *replicate*, *take*, *drop*, *reverse*, etc.
- Nuevas funciones se definen combinando las anteriores o usando pattern matching

$$dupList\ xs = xs +\!\!+ xs$$

$$\begin{aligned} null\ [\,] &= True \\ null\ (\_ : \_) &= False \end{aligned}$$

$$\begin{aligned} head\ (x : \_\ ) &= x \\ tail\ \ (\_ : xs) &= xs \end{aligned}$$

Listas por Comprensión

- Notación que permite construir una lista a partir de una descripción de la misma en términos de otra.

- A partir de una lista se **generan** elementos, que se **testean** y **transforman** para formar los elementos de la lista resultante.

- Un primer ejemplo:
  $[x \mid x \leftarrow [1,2,3,4]] \rightsquigarrow [1,2,3,4]$

- Más ejemplos:
  $[x \mid x \leftarrow [1,2,3,4], \mathit{isEven}\ x] \rightsquigarrow [2,4]$

  $[\mathit{isEven}\ x \mid x \leftarrow [1,2,3,4]] \rightsquigarrow [\mathit{False}, \mathit{True}, \mathit{False}, \mathit{True}]$

  $[(x+y) \mid (x,y) \leftarrow [(1,2),(2,3),(3,4)]] \rightsquigarrow [3,5,7]$

  $[(x,y) \mid x \leftarrow [1,2], y \leftarrow [3,4]] \rightsquigarrow [(1,3),(1,4),(2,3),(2,4)]$

Preguntas

Tipos de Datos Algebraicos

- Mediante la definición de **tipos de datos algebraicos** podemos introducir nuevos tipos.

- Nos permiten definir:
  - Enumerados:

$$\begin{aligned}&\text{data } \mathit{PuntoCardinal} = \mathit{Norte} \mid \mathit{Sur} \mid \mathit{Este} \mid \mathit{Oeste}\\&\quad \text{deriving } (\mathit{Show}, \mathit{Eq})\end{aligned}$$

  - Productos:

$$\begin{aligned}&\text{data } \mathit{Empleado} = \mathit{Empleado}\ \mathit{String}\ \mathit{Int}\ \mathit{Int}\\&\quad \text{deriving } (\mathit{Show}, \mathit{Eq})\end{aligned}$$

  - Alternativas:

$$\begin{aligned}&\text{data } \mathit{Forma} = \mathit{Circulo}\ \ \ \mathit{Float}\\&\qquad\qquad \mid \mathit{Rectangulo}\ \mathit{Float}\ \mathit{Float}\\&\quad \text{deriving } (\mathit{Show}, \mathit{Eq})\end{aligned}$$

Tipos de Datos Algebraicos (2)

- En general:

$$\begin{aligned} \text{data } T = & \; C_1 \; t_{11} \; ... \; t_{1k_1} \\ & \; ... \\ & \mid C_n \; t_{n1} \; ... \; t_{nk_n} \end{aligned}$$

donde T es el nuevo tipo a introducir y cada C_i es un **constructor** de dicho tipo, que toma k_i parámetros.

- Funciones:

$$\begin{aligned} & area :: Forma \rightarrow Float \\ & area \; (Circulo \; r) \qquad = 3.14 * r * r \\ & area \; (Rectangulo \; b \; a) = b * a \end{aligned}$$

$$cuadrado \; l = Rectangulo \; l \; l$$

Preguntas