Programación Funcional

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

Definiciones y Tipos Básicos en Haskell

Definición de funciones

- Un programa en PF consiste esencialmente de una lista de definiciones (de funciones) que se expresan en una notación semejante a la usada en matemática.
- Las definiciones son escritas como ecuaciones, eventualmente acompañadas de una descripción de su tipo (signatura). Ejemplo:

$$num :: Integer$$
$$num = 3 + 6$$

- Podemos omitir la signatura y el tipo se infiere:

$$num = 3 + 6$$

- Para definir funciones, especificamos sus parámetros formales

$$square :: Integer \rightarrow Integer$$
$$square\ x = x * x$$

Tipado

- El tipado juega un rol importante en la producción de software confiable.
- Ayuda a encontrar errores de mala formación de expresiones e impone restricciones que van mas allá de aspectos sintácticos.
- Los tipos forman parte de la especificación del sistema.

Tipado en Haskell

Tipado fuerte: toda expresión debe tener un tipo. Todas las constantes, variables, operadores y funciones tienen un tipo asociado.

Chequeo estático de tipos: los tipos se chequean en tiempo de compilación.

Type safety: al ejecutar, los programas bien tipados nunca se trancan por errores de tipo

well-typed programs never "go wrong"

Tipos básicos

Bool	booleanos
Char	caracteres simples
String	cadena de caracteres
Int	enteros acotados
Integer	enteros no acotados
Float	reales
Double	reales

Nota: Los nombres de tipo en Haskell deben comenzar con mayúscula.

Booleanos

- El tipo de los booleanos es *Bool* y sus constantes son denotadas *True* y *False*
- Las operaciones primitivas son las habituales: *not*, &&, ||.

$$
\begin{aligned}
&exOr :: Bool \to Bool \to Bool \\
&exOr\ x\ y = (x \,||\, y) \,\&\&\, not\ (x \,\&\&\, y)
\end{aligned}
$$

- Podemos definir las funciones usando pattern matching

$$
\begin{aligned}
&exOr :: Bool \to Bool \to Bool \\
&exOr\ True\ y = not\ y \\
&exOr\ False\ y = y
\end{aligned}
$$

Preguntas

Enteros

- Existen dos tipos de enteros: *Int* (enteros acotados) e *Integer* (enteros no acotados).

- Los operadores sobre enteros son los habituales: $+, -, *, \hat{}$, *div*, *mod*, *abs*, *negate*.

- Los enteros pueden ser comparados por operadores relacionales ($<, >, \geqslant, ==, /=, ...$).

$$min :: Integer \to Integer \to Integer$$
$$min\ x\ y = \text{if } x \leqslant y \text{ then } x \text{ else } y$$

- La conversión entre los dos tipos de enteros se debe hacer en forma explícita a través de las funciones:

$$fromInteger :: Integer \to Int$$
$$toInteger \quad :: Int \to Integer$$

Caracteres

- Los caracteres son codificados por la tabla ASCII.
- Se escriben entre comillas simples: `'a'`, `'H'`, `'8'`.
- Algunos caracteres especiales se representan de esta manera:

`'\t'`	*tab*
`'\n'`	*newline*
`'\\'`	*backslash*
`'\''`	*comilla simple*
`'\"'`	*comilla doble*

- La conversión entre un caracter y su codificación se hace a través de las siguientes funciones:

$$fromEnum :: Char \rightarrow Int$$
$$toEnum :: Int \rightarrow Char$$

Strings

- El tipo *String* se define como una lista de caracteres.
- Constantes de tipo *String* se escriben entre comillas dobles: "Funcional", `"\tHaskell\n"`.
- Las funciones *show* y *read* permiten convertir entre valores de otros tipos y *String*.

  Por ejemplo,

  $$\textit{show}\ (4+3) \rightsquigarrow \texttt{"7"}$$

  $$\textit{read}\ \texttt{"3"} + 4 \rightsquigarrow 7$$

Reales

- Existen 2 tipos que representan los reales en punto flotante: *Float* y *Double*.
- Las operaciones sobre estos tipos son las estándar (aritméticas, trigonométricas, etc).
- Operador de división real: $/$
- Hay funciones de conversión hacia y desde los enteros.

$$\begin{aligned} &\mathit{ceiling} :: \mathit{Float} \to \mathit{Integer} \\ &\mathit{floor} \quad :: \mathit{Float} \to \mathit{Integer} \\ &\mathit{round} \;\; :: \mathit{Float} \to \mathit{Integer} \end{aligned}$$

$$\begin{aligned} &\mathit{fromInteger} \;\; :: \mathit{Integer} \to \mathit{Float} \\ &\mathit{fromIntegral} :: \mathit{Int} \to \mathit{Float} \end{aligned}$$

Preguntas

Programación Funcional

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