Lógica Computacional 2020-1
Programación en Haskell

Sara Doris Montes Incin

30 de enero de 2020

Contenido

Programación funcional

¿Qué es la programación funcional?

Es un estilo de programación en el que el método básico de cálculo es la aplicación de funciones a argumentos.

Un lenguaje funcional es aquel que soporta y fomenta el estilo funcional.

Haskell

Lenguaje de programación puramente funcional multipropósito.

En lugar de realizar acciones en secuencia, evalúa expresiones.

Perezoso y estático

Intérprete

- Es un programa que analiza un programa (script) y lo ejecuta.
- Recordemos que los compiladores toman un programa, lo convierten a código máquina (compilación) y este ya puede ser ejecutado en la computadora.
- Los intérpretes toman un programa y convierten en código máquina hasta que sea necesario, es decir, hasta que el usuario solicite ejecutar una instrucción.
- La principal ventaja de los programas interpretados es su facilidad de interacción con el programador, la desventaja es que son ineficientes a comparación de los programas compilados.

¿Y Haskell?

Los dos intérpretes más populares de Haskell son: GHC y Hugs.

Nosotros usaremos GHC.

Instalación

En la página de https: `//www.haskell.org/downloads` podrán encontrar los paquetes de Haskell.

Existen dos versiones Haskell Platform que incluye el manejador de paquetes, el compilador GHC y otras herramientas. O también pueden descargar el compilador GHC que tiene el interpretre ghci.

Uso

Para abrir el intérprete, sólo deben abrir su terminal y escribir la orden:

ghci

Una vez abierto mostrará la siguiente expresión:

>prelude

Comandos básicos

- :l <ruta Archivo> → Carga un programa al intérprete.
- :r → Teniendo un programa abierto, lo vuelve a cargar.
- :h [Comando] → Muestra algunos comandos y una breve descripción de estos.
- :cd <rutaNueva> → Cambia el directorio actual de trabajo.
- :t <Expresion> → Devuelve el tipo de la expresión dada.
- :q → Cierra el intérprete.

¿Qué es un tipo?

Es un nombre para una colección de valores relacionados

Tipos básicos

- Bool
  - True
  - False
- Char: 'L','o','g','+'
- Int: Enteros entre -2^{31} y 2^{31}-1.
- Integers: Enteros de precisión arbitraria (es decir que no tienen longitud determinada)
- Float (Reales de precisión arbitraria): 1.2, -23.45, 45e-7
- Double (Reales de precisión doble)

¿String es tipo básico?

Tipos compuestos

- String. No es tipo básico, ya que un String es una lista de caracteres:
  :t "Hola"
  "Hola" :: [Char]
- Listas. Sucesión de elementos del mismo tipo:
  ['a','b','c']
  [1,2,3,4,5]
- Tuplas. Sucesión de elementos donde no necesariamente tienen que ser del mismo tipo:
  (False,'a',True)

Tipos funciones

Una función es un mapeo (aplicación) de valores de un tipo a valores de otro tipo

En general...

T1 $\rightarrow$ T2 es el tipo de las funciones que aplica valores del tipo T1 en valores del tipo T2. Ejemplos de funciones:

Tipos definidos por el usuario

A través de una declaración *data*

data Color = Rojo — Amarillo — Azul

data Punto a = Pt a a
Pt 2.0 3.0 :: Punto Float

Ejemplos de funciones en Haskell

1.

2. Función que calcula el volumen de una esfera tal que (volumenEsfera r) es el volumen de la esfera de radio r

```
volumenEsfera r = (4/3)*pi*r^3
```

3. Función potencia. Dados dos números n y m, calcular n elevado a la m.

```
potencia :: Integer -> Integer -> Integer
potencia m 0 = 1
potencia m n = m*(potencia m (n-1))
```

Tipos recursivos

Práctica 1

Funciones currificadas (parcializadas)

Funciones con múltiples argumentos pueden regresar funciones.

Pero, ¿cómo lo hacen?

Las funciones que reciben más de un argumento se interpretan como funciones que reciben un argumento y devuelven una función con un argumento menos.

Ejemplo de función currificada

mult:: Int $\rightarrow$ (Int $\rightarrow$ (Int$\rightarrow$ Int))

mult x y z = x*y*z

¿Cómo funciona?

mult 1 2 3

Variable de tipo

No comienza con mayúscula, entonces... ¿es un tipo?

Podemos pensarla como los genéricos en los demás lenguajes

Las funciones que tienen variables de tipos son llamadas *funciones polimórficas*.