



TECNOLÓGICO NACIONAL DE MÉXICO

Campus Colima

Departamento de Sistemas y Computación

INGENIERÍA EN SISTEMAS COMPUTACIONALES

“Programación Lógica y Funcional”

A 2.3 Tipos y clases de tipos en
Haskell.

Que presenta:

Barragán Flores Luis Francisco

Villa de Álvarez, Colima, México a 15 de marzo de 2022

Palabras que identifican al tipo	Descripción del tipo de dato	Rango de valores del tipo	Ejemplos
Int	Representa números enteros, por lo que 7 puede ser un Int pero 7.2 no puede. Int está acotado, lo que significa que tiene un valor máximo y un valor mínimo.	Normalmente en máquinas de 32bits el valor máximo de Int es 2147483647 y el mínimo -2147483648.	addThree :: Int -> Int -> Int -> Int addThree x y z = x + y + z
Integer	Representa enteros también, la diferencia que tiene con el int es que no están acotados así que pueden representar números muy grandes. Sin embargo, Int es más eficiente.	Puede tener cualquier longitud sin ninguna limitación.	factorial :: Integer -> Integer factorial n = product [1..n] ghci> factorial 50
Float	Es un número real en coma flotante de simple precisión.	Va de 6 a 7 dígitos decimales, y su float solo puede tomar 4 bytes para almacenar en la memoria.	circumference :: Float -> Float circumference r = 2 * pi * r ghci> circumference 4.0 25.132742
Double	Es un número real en coma flotante de... ¡Doble precisión!	Contiene números de punto flotante de precisión doble IEEE de 64 bits (8 bytes) firmados que van desde -1,79769313486231570E+308 hasta -4,94065645841246 544E-324 para valores negativos y de 4,94065645841246544E-324 a 1,79769313486231570E+308 para valores positivos.	circumference' :: Double -> Double circumference' r = 2 * pi * r ghci> circumference' 4.0 25.132741228718345
Bool	Solo puede tener dos valores: True o False.	Unicode de 16 bits.	Pt True False :: Point Bool
Char	Representa un carácter. Se define rodeado por comillas	8 bits rango de -128 a 127	'a' :: Char "HELLO!" :: [Char]

Variables de tipos

- Descripción:

es como una etiqueta que posee toda expresión. Esta etiqueta nos dice a que categoría de cosas se ajusta la expresión. La expresión True es un booleano, "Hello" es una cadena, etc.

- Uso:

Los principales usos que tienen las variables de tipos de datos es que permiten a los programadores poder definir fácilmente funciones muy generales.

- Ejemplos:

```
ghci> :t head
head :: [a] -> a
ghci> :t fst f
fst :: (a, b) -> a
```

Clases de tipos:

- Descripción:

La clase de tipos no es más que una especie de interfaz que define algún tipo de comportamiento tales como (Como comparar por igualdad, comparar por orden, una enumeración, etc). El comportamiento de una clase de tipos se consigue definiendo funciones para luego implementarlas.

- Uso:

Algunos usos pueden ser: evaluar si dos números son iguales o no, evaluar si dos caracteres son iguales, evaluar algunas expresiones lógicas o booleanas, etc.

- Ejemplos:

```
ghci> :t (==)
(==) :: (Eq a) => a -> a -> Bool
ghci> 5 == 5
True
ghci> 5 /= 5
False
ghci> 'a' == 'a'
True
ghci> "Ho Ho" == "Ho Ho"
True
ghci> 3.432 == 3.432
True
ghci> :t (>)
(>) :: (Ord a) => a -> a -> Bool
ghci> "Abrakadabra" < "Zebra"
True
ghci> "Abrakadabra" `compare` "Zebra"
LT
```

```
ghci> 5 >= 2
True ghci> 5 `compare` 3
GT
```

Palabras que identifican la clase	Descripción de la clase.	Rango de valores de clase.	Ejemplos
Eq	Es utilizada por los tipos que soportan comparaciones por igualdad	Los miembros de esta clase implementan las funciones == o /= en algún lugar de su definición	ghci> 5 == 5 True ghci> 5 /= 5 False
Ord	Es para tipos que poseen algún orden	Todos los tipos que hemos llegado a ver excepto las funciones son parte de la clase Ord. Ord cubre todas las funciones de comparación como >, <, >= y <=.	ghci> :t (>) (>) :: (Ord a) => a -> a -> Bool
Show	Pueden ser representados por cadenas.	Todos los tipos que hemos visto excepto las funciones forman parte de Show. la función más utilizada que trabaja con esta clase de tipos es la función show. Toma un valor de un tipo que pertenezca a la clase Show y lo representa como una cadena de texto	show False == "False" show 'a' == "'a'" show 123 == "123"
Read	Es como la clase de tipos opuesta a Show. La función read toma una cadena y devuelve un valor del tipo que es miembro de Read.	Todos los tipos que hemos visto excepto las funciones forman parte de esta clase de tipos.	ghci> read "True" False True ghci> read "8.2" + 3.8 12.0
Enum	Son tipos secuencialmente ordenados, es decir, pueden ser enumerados.	Los miembros en las listas aritméticas. También tienen definidos los sucesores y predecesores, por lo que podemos usar las funciones succ y pred. Los tipos de esta clase son: (), Bool, Char, Ordering, Int, Integer, Float y Double.	ghci> ['a'..'e'] "abcde" ghci> [LT .. GT] [LT,EQ,GT]
Bounded	Poseen límites inferiores y superiores, es decir están acotados.	minBound y maxBound	ghci> minBound :: Int -2147483648 ghci> maxBound :: Char

			'1114111'
Num	Es la clase de tipos numéricos. Sus miembros tienen la propiedad de poder comportarse como números. Vamos a examinar el tipo de un número	Incluye todos los números, incluyendo números reales y enteros.	ghci> :t 20 20 :: (Num t) => t

Conclusión

Como conclusión de esta actividad puedo decir que aprendí más acerca de Haskell, ya que, conocí los tipos y las clases de tipos que esta pose y con las cuales trabaja en su programación, también aprendí a identificar que son, que hacen y pude ver los ejemplos de las clases de tipos y de los tipos.

Además, se me hizo interesante las variables de tipos con las que trabaja haskell, como por ejemplo la eq, la cual contiene los tipos cuyos valores son comparables por igualdad, así a simple vista se me hizo algo confusa, pero después de leerla de nuevo pude comprender un poco más acerca de esta y de la función que desempeña en haskell.

Referencias

- [1] A. C. G. y. C. G. Díaz, «Tema 3: Tipos en Haskell,» 2010-2011. [En línea]. Available: <https://www.cs.us.es/cursos/i1m-2010/i1m-g1/temas/tema-03-p.pdf>. [Último acceso: 15 03 2022].
- [2] J. A., «I1M2018: Tipos y clases en Haskell,» 05 10 2018. [En línea]. Available: <https://www.glc.us.es/~jalonso/vestigium/i1m2018-tipos-y-clases-en-haskell/>. [Último acceso: 15 03 2022].
- [3] aprendehaskell.es, «Tipos y clases de tipos,» aprendehaskell.es, [En línea]. Available: <http://aprendehaskell.es/content/Tipos.html>. [Último acceso: 15 03 2022].
- [4] programador clic, «Notas de introducción a Haskell (4),» programador clic, [En línea]. Available: <https://programmerclick.com/article/67431403462/>. [Último acceso: 15 03 2022].