

# Relaciones

Ernesto Rodriguez

Universidad del Itsmo

*erodriguez@unis.edu.gt*

# Tipos inductivos en Elm

- Elm permite declarar conjuntos de valores (conocidos como **tipos**) mediante la palabra reservada **type**.
  - **type** Booleano = T | F
  - **type** Natural = Succ Natural | Cero
- Los tipos se dividen en **casos**, cada caso se separa mediante “ | ”.
- Un tipo puede tener una cantidad arbitraria de casos distintos.
- Los casos pueden ser **recursivos**, similar a nuestra definición original de **Naturales Unarios**

# Utilización de Tipos

- Se pueden definir funciones para valores de nuestros tipos nuevos mediante **análisis de casos**, coloquialmente conocido como **pattern matching**.
- Elm (a diferencia de SML, Haskell, etc.) requiere que los casos sean **exhaustivos**, es decir que se abarquen todos los casos.
  - Ayuda a mejorar la robustez de nuestro código
  - Puede ser difícil agregar casos nuevos a tipos existentes
- Elm tampoco permite **casos redundantes**, ya que pueden llevar a **inconsistencias**.

```
This "case" does not have branches for all possibilities:
5| esCero n = case n of
6|   Succ n_ -> False
Missing possibilities include:
  Cero
```

```
The 3rd pattern is redundant:
5| esCero n = case n of
6|   Succ n_ -> False
7|   Cero -> True
8|   Succ (Succ n_) -> False
  ^^^^^^^^^^^^^^^^^
```

# Ejemplo: Dias de la Semana

- Definición del tipo:  
> **type** dia = Lun | Mar | Mie | Jue | Vie | Sab | Dom
- Función para reconocer dias de trabajo:

```
diaDeTrabajo dia = case dia of
                      Sab → False
                      Dom → False
                      _   → True
```

- Es posible utilizar la función:  
> diaDeTrabajo Dom  
False

# Ejemplo: Numeros Naturales Unarios

Ver la carpeta de de ejemplos (“../ejemplos”)

# Abstraccion: Representando objetos como tipos

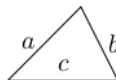
- Se describiran tres figuras geometricas diferentes:



Circle ( $r$ )



Square ( $a$ )



Triangle ( $a, b, c$ )

**Matematicamente:**  $\mathbb{R}^+ \uplus \mathbb{R}^+ \uplus (\mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+)$

- En Elm, el conjunto  $\mathbb{R}$  se aproximara mediante el tipo `Float`
- Esto permite definir las figuras por casos:

# Abstraccion: Representando objetos como tipos

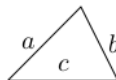
- Se describiran tres figuras geometricas diferentes:



Circle ( $r$ )



Square ( $a$ )



Triangle ( $a, b, c$ )

**Matematicamente:**  $\mathbb{R}^+ \uplus \mathbb{R}^+ \uplus (\mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+)$

- En Elm, el conjunto  $\mathbb{R}$  se aproximara mediante el tipo Float
- Esto permite definir las figuras por casos:

```
type Figura = Circulo Float  
             | Cuadrado Float  
             | Triangulo (Float,Float,Float)
```

# Abstracción: Area de Figuras

Ver la carpeta de de ejemplos (“../ejemplos”)