Aula T3 - Anotações

4 de outubro de 2023 21:16

$$\begin{cases} id : A \to A \\ id\ a = a \end{cases}$$

$$\begin{tikzcd} A \arrow[r, bend left, "id"] \arrow[r, phantom, "\cong"] & A \arrow[l, bend left, "id"] \end{tikzcd}$$

$id \cdot id = id$
$id \circ id = id$

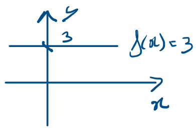
$$\begin{cases} swap : A \times B \to B \times A \\ swap\ (a, b) = (b, a) \end{cases}$$

Isomorfismo $\Rightarrow$ Não há perda de informação

$$\begin{cases} zero\ x = 0 \\ one\ x = 1 \end{cases}$$

$\hookrightarrow$ one 10 = 1
one "string" = 1

$\Rightarrow$ Função constante



$f\ x = 3$
$\hookrightarrow f\ 12 = 3$
f "string" $= 3$

Type?

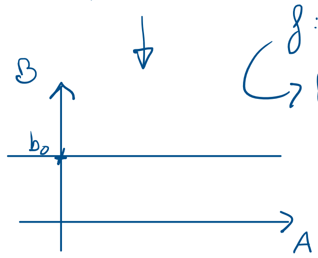
$f :: \boxed{Num\ p_1} \Rightarrow \boxed{p_2} \to p_1$

$\hookrightarrow$ Polimórfica

$\hookrightarrow$ qq tipo p_2

$$\boxed{\begin{array}{l} b_0 \in B \\ \begin{cases} \underline{b_0} : A \to B \\ \underline{b_0}\ a = b_0 \end{cases} \end{array}}$$

Função constante



(Haskell const b_0)

$b \circ f = \underline{b}$

$g \cdot \underline{b} = \underline{g\ b}$

$$\begin{tikzcd} & C \arrow[dl, "\underline{b}"'] \arrow[d, "f"] \\ B \arrow[d, "g"'] & A \arrow[l, "\underline{b}"'] \arrow[dl, "\underline{g\ b}"] \\ D & \end{tikzcd}$$

$\Rightarrow$ Saída FIXA, Entrada variável

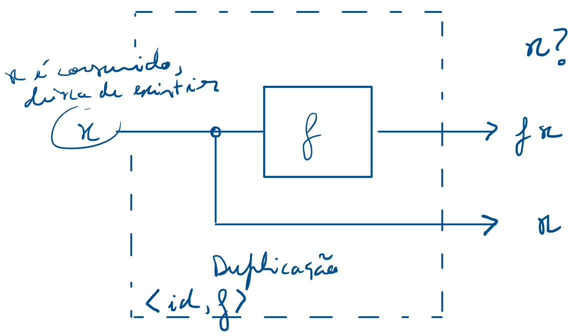
```
Prelude> :t +d const 3
const 3 :: b -> Integer
Prelude> :t const 3
const 3 :: Num a => b -> a
```

Mecanismo de inferência de tipos

Gestão de informação

$$\begin{tikzcd} A \arrow[r, bend left, "\langle id, f \rangle"] \arrow[r, phantom, "\cong"] & A \times B \arrow[l, bend left, "\pi_1"] \end{tikzcd}$$

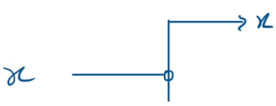
$\langle id, f \rangle\ a = (a, f\ a)$



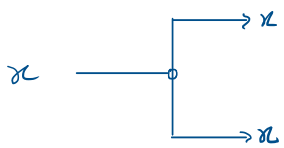
"Information is physical"

Duplicação de informação

(Split id id)



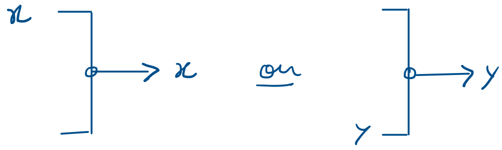
$dup = \langle id, id \rangle$



$dup = \langle id, id \rangle$

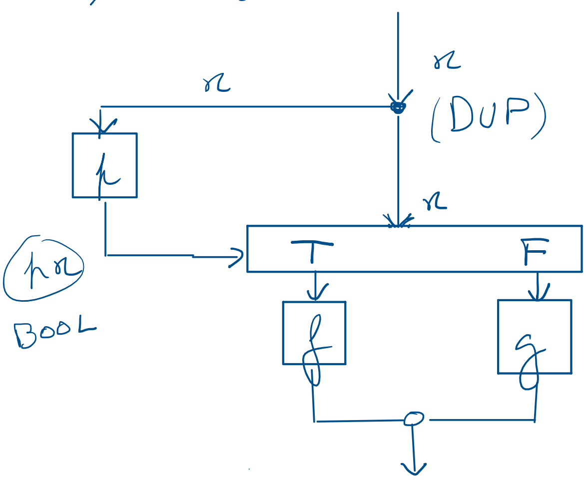
Junção de informação

"ou passa o x ou passa o y"



$join = [id, id]$

$y = \text{if } p\,x \text{ then } f\,x \text{ else } g\,x$



→ é uma descrição matemática?

data Bool = False | True

$x \in A$

$(p\,x, x) \in \mathbb{B} \times A$

$\mathbb{B} = \{F, T\} \cong \{False, True\} \cong \{0, 1\} \cong \underline{2}$

$$\begin{tikzcd} \mathbb{B} \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & \{0,1\} \arrow[ll, bend left, "f"] \end{tikzcd}$$

$$A + A \xrightarrow{join} A$$

$$A \xrightarrow{\langle p, id \rangle} 2 \times A$$

Será que:

$$\begin{tikzcd} 2 \times A \arrow[rr, bend left, "g"] \arrow[rr, phantom, "\cong"] & & A + A \arrow[ll, bend left, "f"] \end{tikzcd}$$

α°

A com etiqueta do lado esquerdo
e um A com etiqueta do lado direito

Etiquetas explícitas:

$g \Rightarrow \alpha$

Etiquetas explícitas:
Um A concreto, e com etiqueta 1
ou etiqueta 2

$$g = [\langle T, id\rangle, \langle F, id\rangle] \qquad g = \ldots$$

Alternativa entre um split True, id, e um split False ID

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "T"'] \arrow[dr, "\langle T, id\rangle"] \arrow[drr, "id" near end] & A + A \arrow[d, "{[\langle T, id\rangle, \langle F, id\rangle]}"] & A \arrow[l, "i_2"'] \arrow[dll, "F" near end] \arrow[dl, "\langle F, id\rangle"'] \arrow[d, "id"] \\
2 & 2 \times A \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A
\end{tikzcd}$$

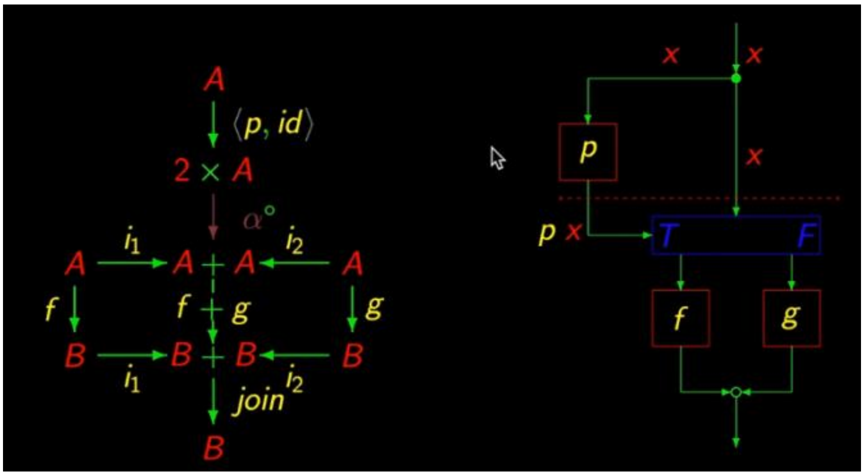
Recordando: $\alpha y = z$

$\Leftrightarrow y = \alpha^{-1} z$

$\Rightarrow \alpha \circ g = h$

$\Leftrightarrow g = h \circ \alpha^{\circ}$

(Só para α isomorfismo)



Condicional de McCarthy

$p \to f\ ,\ g$ "se p então f senão g"

↳ Notação que não usa variáveis

$$p \to f, g = join \circ (f+g) \circ \alpha^\circ \circ \langle p, id\rangle$$

Do diagrama: $p \to f\ ,\ g \;=\; join \cdot (f+g) \cdot \alpha^\circ \cdot \langle p, id\rangle$

Ora $join \cdot (f+g) = [id, id] \cdot (f+g) = [f, g]$

Logo:

$$p \to f\ ,\ g \;=\; [f, g] \cdot \underbrace{\alpha^\circ \cdot \langle p, id\rangle}_{p?} \longrightarrow \text{Guarda}$$

Guarda de p

Guarda de p:

$$\begin{tikzcd} A \arrow[r, "\langle p, id\rangle"] \arrow[rr, bend right, "p?"'] & 2 \times A \arrow[r, "\alpha^\circ"] & A + A \end{tikzcd}$$

Condicional:

$$\boxed{p \to f\ ,\ g \;=\; [f, g] \cdot p?}\ !$$

"p interrogado"

$$p? : A \longrightarrow A + A$$

$$\left\{\ p?\,a = \begin{cases} p\,a \Rightarrow i_1\,a \\ \neg(p\,a) \Rightarrow i_2\,a \end{cases}\right\}$$

$$\left[\ (p \to f, g)\,a = \begin{cases} p\,a \Rightarrow f\,a \\ \neg(p\,a) \Rightarrow g\,a \end{cases}\right]$$

↳ Análise por casos

↳ Não!

$$h \cdot (p \to f, g)$$

$$\Leftrightarrow \underbrace{h \circ [f, g]}_{\{\text{Fusão-+}\}} \cdot p?$$

$\Leftrightarrow [h \cdot f, h \cdot g] \cdot p?$

$\Leftrightarrow p \to (h \cdot f), (h \cdot g)$

E se h correr antes do condicional?

$(p \to f, g) \cdot h = ?$

$(p \cdot h) \to (f \cdot h), (g \cdot h)$

Ter-se-á:

$$(p \to f, g) \cdot h = (p \cdot h) \to (f \cdot h), (g \cdot h)$$

Intuitivamente 🙂

Prova? Só mais tarde.

(precisamos de saber algo mais sobre p?)

T03 - 2º Parte

$A + A \cong 2 \times A$

Etiquetas do tipo 2

Geralmente Booleanos

Faz-se uma "metáfora"/ analogia com a aritmética, mas são coisas diferentes.
Esta expressão diz que a união disjunta de A com A é "de forma semelhante" (isomorfismo) o produto do tipo 2 (canonicamente boolean) com o tipo A (bool, A) - os elementos do tipo A ficam com etiquetas do tipo 2

$A \times A \cong A^2$?

$A \times 1 \cong A$?

$A \times 0 \cong 0$?

$A + 0 \cong A$?

Isomorfismos

$A \times (B + C) \cong A \times B + A \times C$

Distributividade efetuada para a direita

$$\begin{tikzcd} A \times (B + C) \arrow[r, bend left, "distr"] & A \times B + A \times C \arrow[l, bend left, "undistr"] \end{tikzcd}$$

$$\begin{tikzcd} (B + C) \times A \arrow[r, bend left, "distl"] & B \times A + C \times A \arrow[l, bend left, "undistl"] \end{tikzcd}$$

$$\begin{tikzcd} A + B \arrow[d, "\alpha"] \\ {} \end{tikzcd} \Longrightarrow \begin{tikzcd} A \arrow[r, "i_1"] \arrow[dr, "i"'] & A + B \arrow[d, "{[i, j]}"] & B \arrow[l, "i_2"'] \arrow[dl, "j"] \\ & {} & \end{tikzcd}$$

$$\begin{tikzcd} & A + B \arrow[dl, "m"'] \arrow[d, "{\langle m, n \rangle}"] \arrow[dr, "n"] & \\ {} & {} & {} \end{tikzcd}$$

$$\begin{tikzcd} {} \arrow[d, "\alpha"] \\ C \times D \end{tikzcd} \Longrightarrow \begin{tikzcd} {} \arrow[dr, "i"'] & {} \arrow[d, "{[i,j]}"] & {} \arrow[dl, "j"] \\ & C \times D & \end{tikzcd}$$

$$\begin{tikzcd} & {} \arrow[dl, "m"'] \arrow[d, "{\langle m, n \rangle}"] \arrow[dr, "n"] & \\ C & C \times D \arrow[l, "\pi_2"'] \arrow[r, "\pi_2"] & B \end{tikzcd}$$

$\langle m, n \rangle = [i, j]$ ⇝ 4 incógnitas

↳ Leis universais

$$\Leftrightarrow \begin{cases} \pi_1 \cdot [i, j] = m \\ \pi_2 \cdot [i, j] = n \end{cases} \quad \Leftrightarrow \begin{cases} [\pi_1 \cdot i, \ \pi_1 \cdot j] = m \\ [\pi_2 \cdot i, \ \pi_2 \cdot j] = n \end{cases}$$

$$\Leftrightarrow \quad \{ \text{ universal-+ duas vezes } \}$$

$$\begin{cases} \begin{cases} m \cdot i_1 = \pi_1 \cdot i \\ m \cdot i_2 = \pi_1 \cdot j \end{cases} \\ \begin{cases} n \cdot i_1 = \pi_2 \cdot i \\ n \cdot i_2 = \pi_2 \cdot j \end{cases} \end{cases}$$

→

$$\Leftrightarrow \quad \{ \text{ re-organizando equações } \}$$

$$\begin{cases} \begin{cases} m \cdot i_1 = \pi_1 \cdot i \\ n \cdot i_1 = \pi_2 \cdot i \end{cases} \\ \begin{cases} m \cdot i_2 = \pi_1 \cdot j \\ n \cdot i_2 = \pi_2 \cdot j \end{cases} \end{cases}$$

$$\Leftrightarrow \quad \{ \text{ universal-} \times \text{ duas vezes } \}$$

$$\begin{cases} i = \langle m \cdot i_1, n \cdot i_1 \rangle \\ j = \langle m \cdot i_2, n \cdot i_2 \rangle \end{cases}$$

Juntando tudo

$$\begin{cases} m = [f, g] \\ n = [h, k] \\ i = \langle f, h \rangle \\ j = \langle g, k \rangle \end{cases}$$

Substituindo soluções em

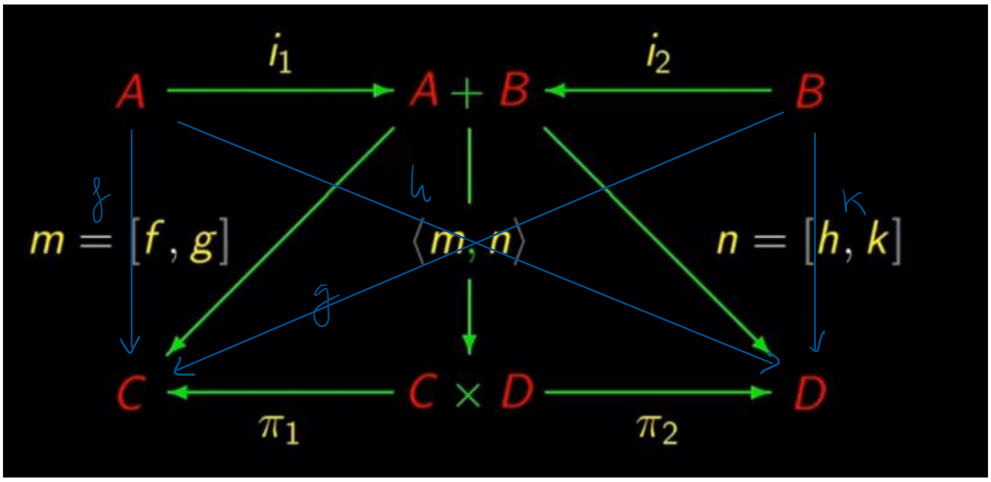
$$\langle m, n \rangle = [i, j]$$

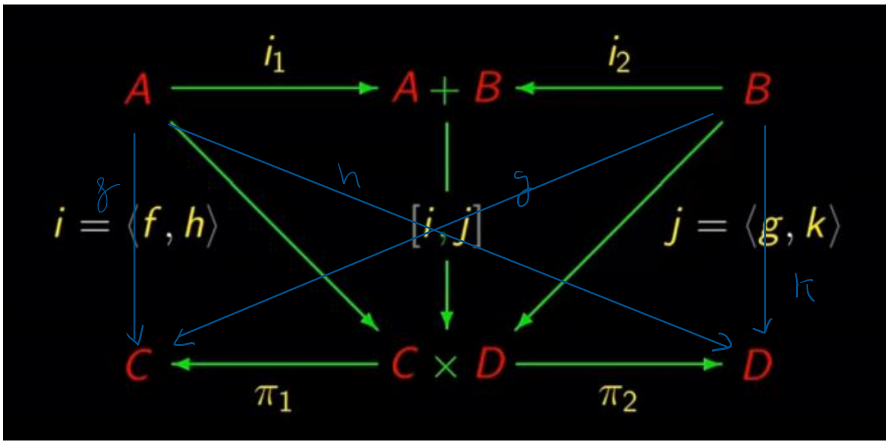
obtém-se a **lei da troca**:

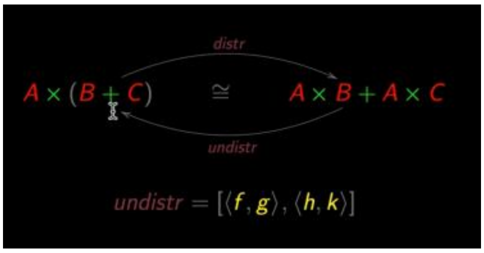
$$\langle [f, g], [h, k] \rangle = [\langle f, h \rangle, \langle g, k \rangle]$$

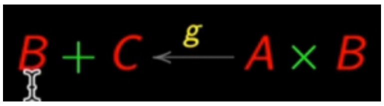
Split

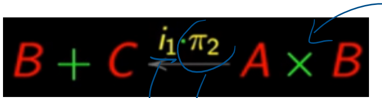
Split







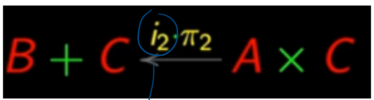




Par A,B

Excluir o A

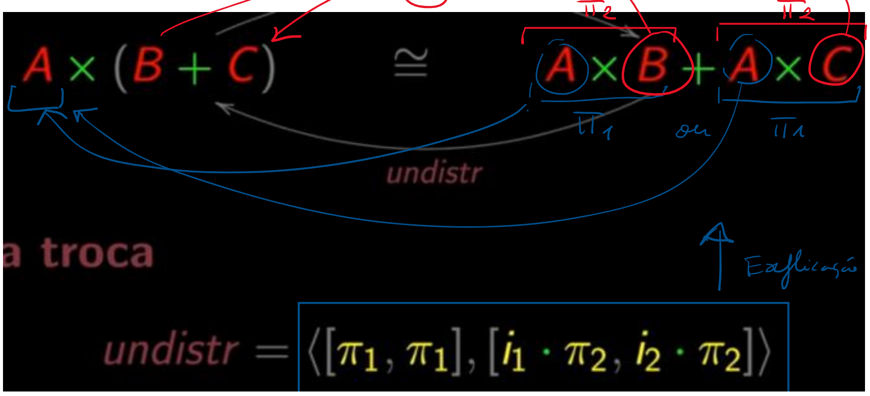
injetar o B no lado da etiqueta i1



injeta C com etiqueta i2

i_1 i_2

π_2 π_2



Explicação

$$undistr = \langle [\pi_1, \pi_1], [i_1 \cdot \pi_2, i_2 \cdot \pi_2] \rangle$$

Lei da troca:

$$\langle [f, g], [h, k] \rangle = [\langle f, h \rangle, \langle g, k \rangle]$$

"tenho um split e dava-me jeito uma alternativa..."

e vice-versa

Em Haskell:

() :: ()

Só tem um único elemento

Logo $1 = \{()\}$ } O Tipo 1

$$\begin{tikzcd} A \times 1 \arrow[r, bend left, "\pi_1"] & A \arrow[l, bend left, "\langle id, ! \rangle"] \end{tikzcd}$$

Funções que envolvam o tipo 1 são necessariamente constantes

$$A \xrightarrow{!} 1$$

"Bang"

$! = ()$

"Points" $1 \xrightarrow{\underline{a}} A$

$\underline{a}\ () = a, \quad a \in A$

$\boxed{\text{O Tipo 0}}$

data Zero

$0 = \{\}$ → não é habitado: $\neg(a \in 0)\ \forall a$

Impossível $f : A \longrightarrow 0, \quad A \neq 0$

$$\begin{tikzcd} A \times 0 \arrow[r, bend left, "id"] & 0 \arrow[l, bend left, "id"] \end{tikzcd}$$

$A \times 0 = 0$ → $A \times 0 = \underbrace{\{(a,b) \mid a \in A \wedge \underbrace{b \in 0}_{\text{sempre FALSO}}\}}_{\text{FALSO}}$

$\Rightarrow A \times 0 = \{\} = 0$

$$\begin{tikzcd} 1 \arrow[r, "i_1"] \arrow[dr, "b_0"'] & 1 + A \arrow[d, "{[b_0, f]}"] & A \arrow[l, "i_2"'] \arrow[dl, "f"] \\ & B & \end{tikzcd}$$

CONSTANTE

"Apontador para A"

Apontador Nulo/Vazio

↳ b_0 correto ex.: 0xFFFF

Obter uma morada de um funcionário público, sabendo que este se pode identificar através do seu número do cartão de cidadão (CC) ou do seu número de identificação fiscal (NIF).

cidadão (CC) ou do seu número de identificação fiscal (NIF).

Em Haskell (Portugal):

```
data Iden = CC Int | NIF Int
```

Em Haskell (Alemanha):

```
data Iden = IDK Int | SZN Int
```

$$
\begin{tikzcd}
Int \arrow[r, "i_1"] \arrow[dr, "CC"'] & Int + Int \arrow[d, "{[CC, NIF]}"] & Int \arrow[l, "i_2"'] \arrow[dl, "NIF"] \\
& Iden &
\end{tikzcd}
$$

```
data Iden = CC Int | NIF Int
```

$$
\begin{tikzcd}
Iden \arrow[rr, bend left, "out = in^{\circ}"] \arrow[rr, phantom, "\cong"] & & Int + Int \arrow[ll, bend left, "{in = [CC, NIF]}"]
\end{tikzcd}
$$

Sintaxe concreta (Iden) — Sintaxe abstrata (Int + Int)

$$
\begin{tikzcd}
1 \arrow[r, "i_1"] \arrow[dr, "Nothing"'] & 1 + A \arrow[d, "{[Nothing, Just]}"] & A \arrow[l, "i_2"'] \arrow[dl, "Just"] \\
& Maybe\ A &
\end{tikzcd}
$$

Nothing → constante

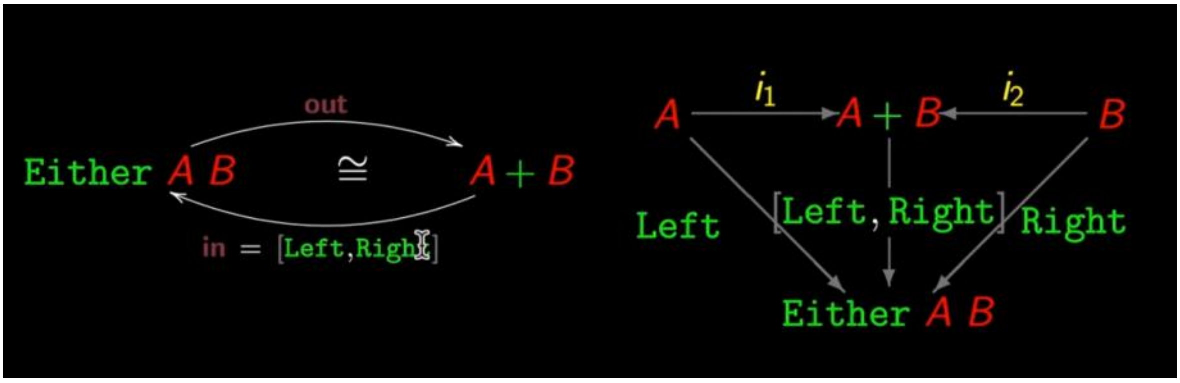
```
data Maybe a = Nothing | Just a
```

$$
\begin{tikzcd}
Maybe\ A \arrow[rr, bend left, "out = in^{\circ}"] \arrow[rr, phantom, "\cong"] & & 1 + A \arrow[ll, bend left, "{in = [Nothing, Just]}"]
\end{tikzcd}
$$

$$out \cdot in = id$$

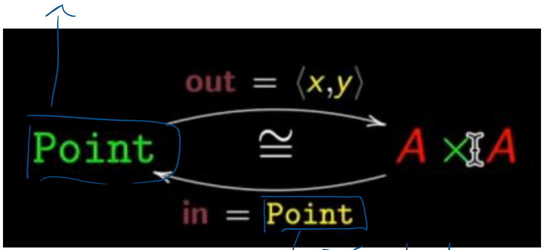
$$
\begin{aligned}
out \cdot in = id \quad &\Leftrightarrow \quad out \cdot [Nothing, Just] = id \\
&\Leftrightarrow \quad [out \cdot Nothing, out \cdot Just] = id \\
&\Leftrightarrow \quad \begin{cases} out \cdot Nothing = i_1 \\ out \cdot Just = i_2 \end{cases} \\
&\Leftrightarrow \quad \begin{cases} (out \cdot Nothing)\ () = i_1\ () \\ (out \cdot Just)\ a = i_2\ a \end{cases} \\
&\Leftrightarrow \quad \begin{cases} out\ Nothing = i_1\ () \\ out\ (Just\ a) = i_2\ a \end{cases}
\end{aligned}
$$

$$\underline{Nothing}\ \underline{1} \longrightarrow Maybe\ A$$



STRUCTS

Tipo de Dados



Construtor de tipos

```
data Point a = Point {
      x :: a,
      y :: a
}
```

$$
\begin{tikzcd}
A & A \times A \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "Point"] & A \\
 & Point \arrow[ul, "x"] \arrow[ur, "y"'] &
\end{tikzcd}
$$