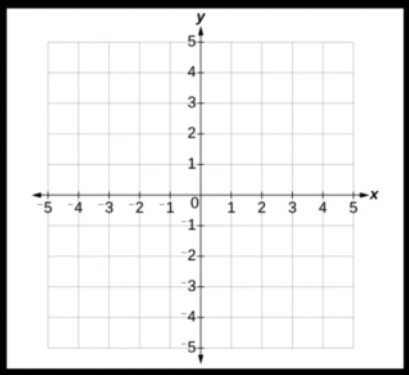
Aula T4 - Anotações

9 de outubro de 2023 21:19

"Structs"

Pontos no plano cartesiano:



Haskell:

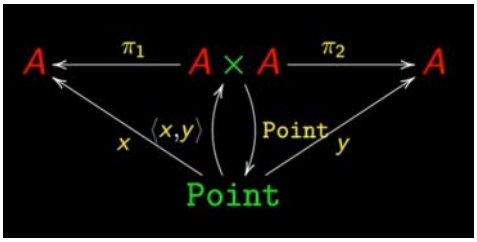
```
data Point a = Point {
   x :: a,
   y :: a
}
```

$out = \langle x,y \rangle$ — Par de coordenadas

$Point \cong A \times A$

$in = Point$

Point — Concreto; $A \times A$ — Abstrato



"Não é exatamente assim"

$Point :: a \to a \to Point$

$\neq$ ⇒ Exponenciação

$Point :: (a, a) \to Point$

Variações de notação

2 + 3	notação infixa
+ 2 3	notação prefixa
2 3 + (machine code)	notação posfixa
+ (2, 3)	prefixa "uncurried"

↳ Tuplos

→ "curried" f a b

Notação Prefixa

$\pi_1 : A \times B \to A$

$\pi_1\ (a, b) = a$

↳ Tuplo

"Uncurried"

$\pi_1' = curry\ \pi_1$

$\pi_1 = uncurry\ \pi_1'$

Importante:

- curry (e uncurry) aceitam funções como argumentos
- curry (e uncurry) dão funções como resultados

↳ FUNÇÕES DE ORDEM SUPERIOR

$curry :: ((a, b) \to c) \to (a \to b \to c)$

$uncurry :: (a \to b \to c) \to (a, b) \to c$

curry f a b = f (a,b)

uncurry g (a,b) = g a b

$uncurry\ g\ (a, b) = g\ a\ b$

$uncurry\ (curry\ f)\ (a, b) = curry\ f\ a\ b$

$uncurry\ (curry\ f)\ (a, b) = f\ (a, b)$ ⇒ (uncurry ∘ curry) f = f

↳ composição

∘ Introduzir uma função

↳ composição

→ (uncurry ∘ curry) f = f

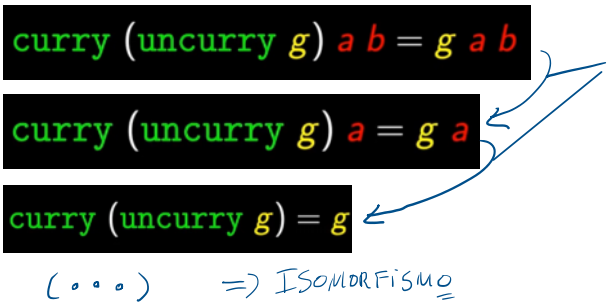
∘ Introduzir uma função

⇒ (uncurry . curry) f = id f

x = id x

⇒ (uncurry ∘ curry) f = id f , f x = g x <=> f = g

<=> uncurry ∘ curry = id ⇒ ISOMORFISMO



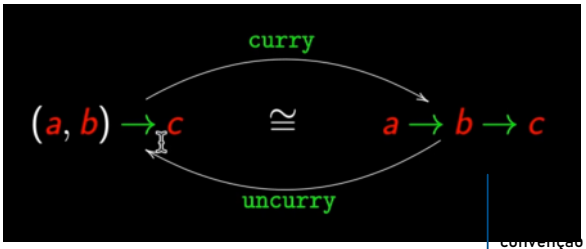
(∘ ∘ ∘) ⇒ ISOMORFISMO

f x = g x <=> f = g

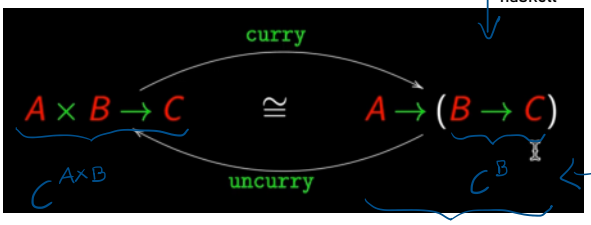
h a b = k a b

<=> h a = k a

<=> h = k



convenção do haskell



Tipo B^A -> é um tipo habitado por funções

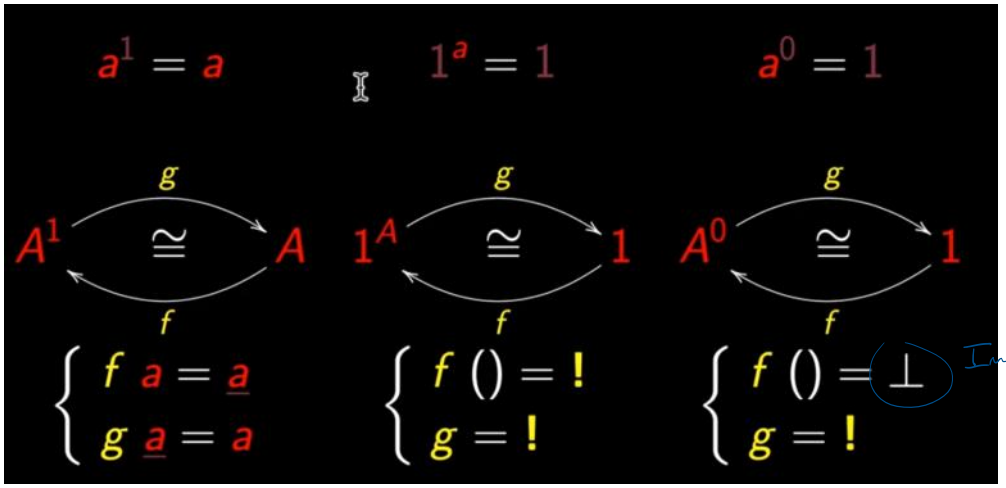
$= \{ f \mid f : A \to B \}$

-> qualquer função de A para B (do expoente para a base)

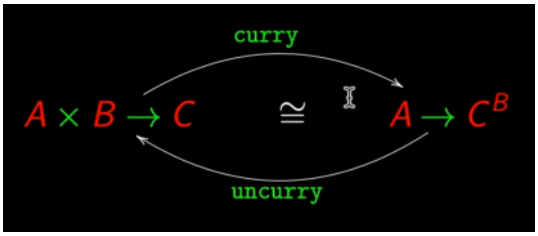
Potências numéricas

$C^{ab} = (C^b)^a$

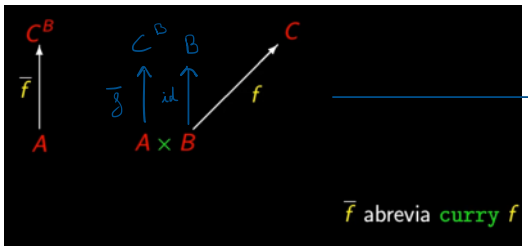
Tipo que contém funções

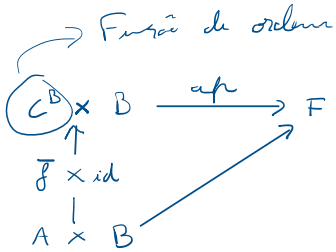


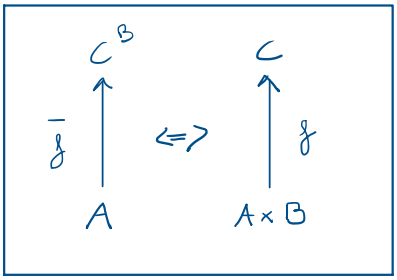
Indefinido

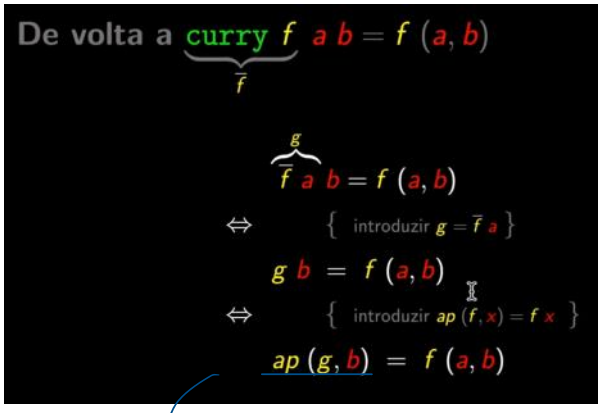


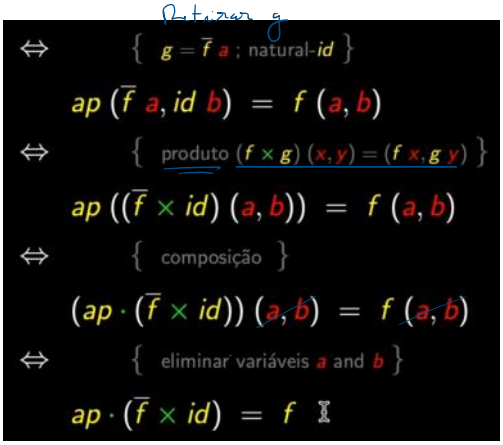
Todas as funções que têm pares como argumentos podem transformar-se em funções que não têm esses pares como argumentos. O segundo argumento foi diferido para a saída. Multiplicar por B na entrada das funções corresponde a elevar a B na saída das funções.
"Se avaliar o A depois quando tiver o B hei de dar um C"

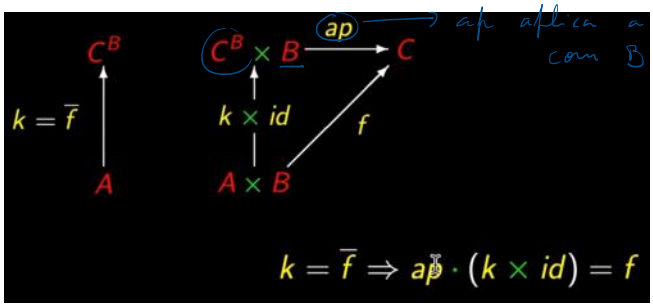






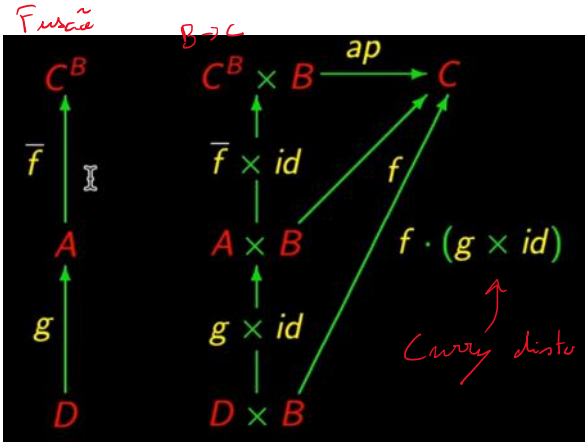


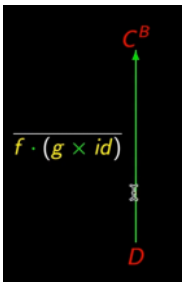




$ap \cdot (\overline{k} \times id)\ (a,b)$
$\Leftarrow ap\ (\overline{k}\ a,\ b)\ (a,b)$
$\Leftarrow (\overline{k}\ a)\ (b)$

$B \to C$
A





$$\overline{f} \cdot g = \overline{f \cdot (g \times id)}$$

Fusão $\overline{f} \cdot g = \overline{f \cdot (g \times id)}$

$(\text{curry } f) \cdot g = \text{curry } (f \cdot (g \times id))$

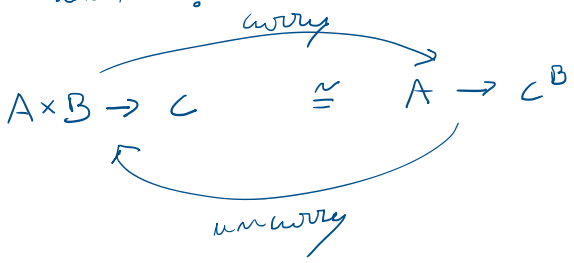
Fusão $\overline{f} \cdot g = \overline{f \cdot (g \times id)}$

$$
\begin{aligned}
&(\text{curry } f) \cdot g = \text{curry } (f \cdot (g \times id)) \\
\Leftrightarrow &\quad \{\ \text{uncurry} \cdot \text{curry} = id\ \} \quad \text{Aplicar uncurry nos dois lados} \\
&\text{uncurry } (\text{curry } f \cdot g) = f \cdot (g \times id) \\
\Leftrightarrow &\quad \{\ \text{introduzir variáveis } d \text{ e } b\ \} \\
&\text{uncurry } (\text{curry } f \cdot g)\ (d, b) = f\ (g\ d, b) \\
\Leftrightarrow &\quad \{\ \text{definição de uncurry}\ \} \\
&(\text{curry } f \cdot g)\ d\ b = f\ (g\ d, b) \\
\Leftrightarrow &\quad \{\ \text{composição de funções}\ \} \\
&\text{curry } f\ (g\ d)\ b = f\ (g\ d, b)
\end{aligned}
$$

Parte 2

$unDistL = [i_1 \times id, i_2 \times id]$; $\text{curry } f = \overline{f}$

$distL = ?$ $\quad \text{uncurry } f = \hat{f}$



isomorfismo $\Rightarrow$ $f = g \Leftrightarrow \overline{f} = \overline{g}$, $k = h \Leftrightarrow \overline{k} = \overline{h}$ } Não se perde informação

$$distL \cdot unDistL = id$$

$$\Leftarrow distL \cdot [i_1 \times id, i_2 \times id] = id$$

$$\Leftarrow \begin{cases} distL \cdot (i_1 \times id) = i_1 \\ distL \cdot (i_2 \times id) = i_2 \end{cases} \quad \Leftarrow \begin{cases} \overline{distL \cdot (i_1 \times id)} = \overline{i_1} \\ \overline{distL \cdot (i_2 \times id)} = \overline{i_2} \end{cases}$$

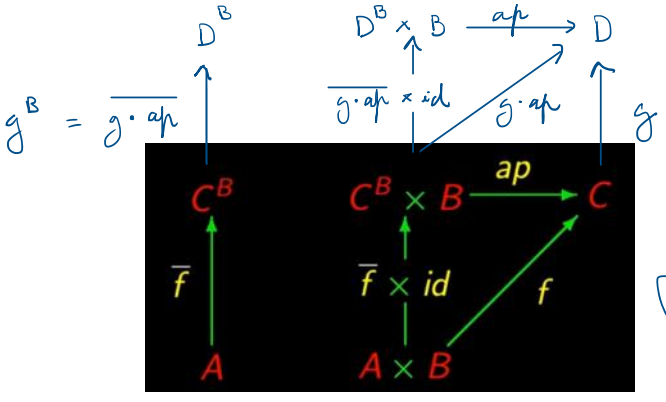
FUSÃO - EXP

$$\Leftarrow \begin{cases} \overline{distL} \cdot i_1 = \overline{i_1} \\ \overline{distL} \cdot i_2 = \overline{i_2} \end{cases}$$

Universal +

$$\Leftarrow \overline{distL} = [\overline{i_1}, \overline{i_2}]$$

$$\Leftarrow distL = \overline{[\overline{i_1}, \overline{i_2}]}$$



Mesmo tipo de entrada/ domínio

$g^B : \overline{C^B \longrightarrow D^B}$

É aplicada a uma função, e dá uma função como resultado.

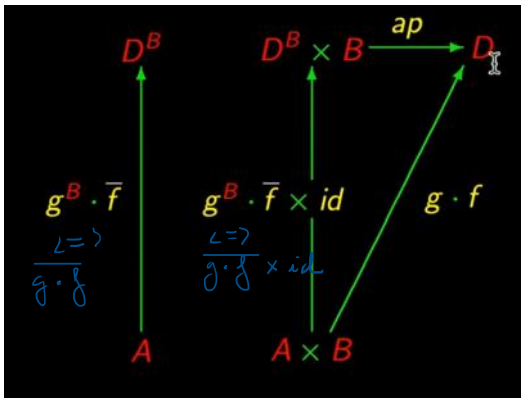
$$g^B = \overline{g \cdot ap}$$

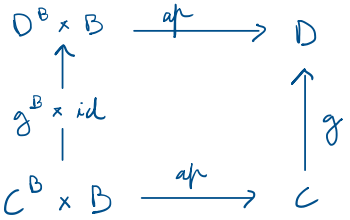
$\overline{g} = \overline{g \cdot ap}$

$$
\begin{aligned}
& g^B = \overline{g \cdot ap} \\
\Leftrightarrow & \quad \{ \text{ introduzir variáveis } \} \\
& g^B\ f\ b = \overline{g \cdot ap}\ f\ b \\
\Leftrightarrow & \quad \{ \text{ definição de } \overline{f} \text{ ("pointwise") } \} \\
& g^B\ f\ b = (g \cdot ap)\ (f, b) \\
\Leftrightarrow & \quad \{ \ ap\ (f, x) = f\ x \ \} \\
& g^B\ f\ b = g\ (f\ b) \\
\Leftrightarrow & \quad \{ \text{ composição de funções } \} \\
& g^B\ f = g \cdot f \quad \blacksquare
\end{aligned}
$$

$g^B f : B \to C$

↳ confere com $g : C \to D$





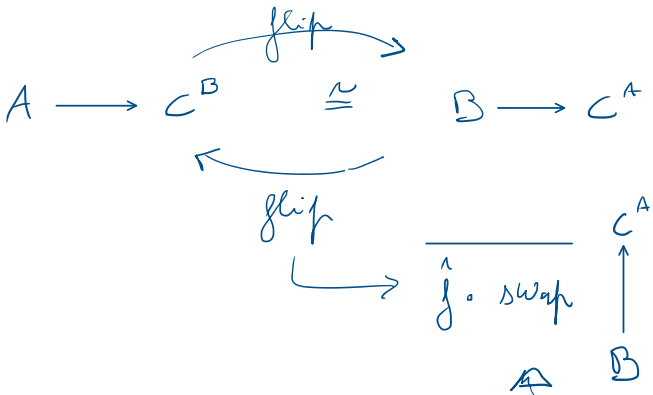
$$g \cdot ap = ap \cdot (g^B \times id)$$

$$
\begin{aligned}
\Leftrightarrow \quad & g \cdot ap\ (a, b) = ap \cdot (g^B \times id)\ (a, b) \\
\Leftrightarrow \quad & g \cdot a\ (b) = g^B\ (a, b) \\
\Leftrightarrow \quad & g \cdot a\ (b) = g^B\ a\ b \\
\Leftrightarrow \quad & g \cdot a\ (b) = g \cdot a\ (b)
\end{aligned}
$$

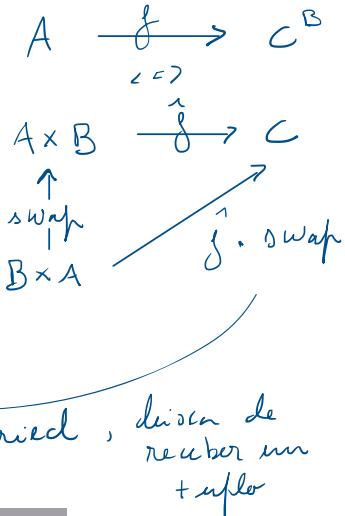
```
ghci> twice g = g . g
ghci>
ghci> f = twice(3+)
ghci> f 10
16
```

FLIP

$$flip : (A \to (B \to C)) \to (B \to (A \to C))$$



"Type oriented programming"



curried, deixa de receber um tuplo

```
Prelude> unJust (Just x) = x
Prelude> l2f l = unJust . (flip lookup l)
Prelude>
Prelude> data Pessoa = Maria | John deriving (Eq,Show)
Prelude> data Pais = Portugal | England deriving (Eq,Show)
```

```
Prelude> unJust (Just x) = x
Prelude> l2f l = unJust . (flip lookup l)
Prelude>
Prelude> data Pessoa = Maria | John deriving (Eq,Show)
Prelude> data Pais = Portugal | England deriving (Eq,Show)
Prelude> data Cidade = Braga | London deriving (Eq,Show)
Prelude>
Prelude> r = [(Maria,Braga),(John,London)]
Prelude> s = [(Braga,Portugal),(London,England)]
Prelude> h = [(Portugal,10),(England,55)]
Prelude>
Prelude> :t l2f
l2f :: Eq a => [(a, c)] -> a -> c
```

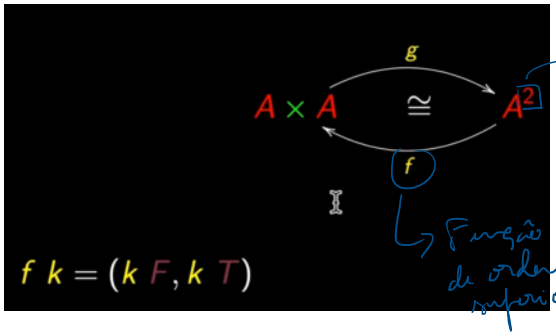
```
Prelude> res = l2f r
Prelude> pais = l2f s
Prelude> hab = l2f h
Prelude>
Prelude> :t res
res :: Pessoa -> Cidade
Prelude> :t pais
pais :: Cidade -> Pais
Prelude> :t +d hab
hab :: Pais -> Integer
Prelude>
Prelude> (hab . pais . res) Maria
10
```

Sem FLIP:

```
ghci> l2fv2 l = unJust . lookup l
ghci> res = l2f r
ghci> res Maria
Braga
ghci> resv2 = l2fv2 r
ghci>
ghci> resv2 Maria

<interactive>:27:7: error: [GHC-83865]
    * Couldn't match expected type '[[[(Pessoa, Cidade)], c)]'
                  with actual type 'Pessoa'
    * In the first argument of 'resv2', namely 'Maria'
      In the expression: resv2 Maria
      In an equation for 'it': it = resv2 Maria
    * Relevant bindings include it :: c (bound at <interactive>:27:1)
```

$$flip = \overline{\hat{g} \circ swap}$$



Booleano

$K : 2 \to A$

$gK = (KF, KT)$

Função de ordem superior

$$\begin{cases} g(a_1, a_2)\ F = a_1 \\ g(a_1, a_2)\ T = a_2 \end{cases}$$

$$\begin{cases} g(h,k) = [h,k] \\ gK = (K \cdot i_1, K \cdot i_2) \end{cases} \quad \longleftrightarrow \quad C^{A+B} \overset{g}{\underset{\hat{g}}{\cong}} C^A \times C^B$$

$$\begin{cases} g(h,k) = \langle h,k \rangle \\ gK = (\pi_1 \cdot K, \pi_2 \cdot K) \end{cases} \quad \longleftrightarrow \quad (A \times B)^C \overset{g}{\underset{\hat{g}}{\cong}} A^C \times B^C$$

Sumário

$f \cdot g$	sequencial
$\langle f, g \rangle$, $f \times g$	paralela
$[f, g]$, $f + g$, $p \to f, g$	alternativa
$\overline{f}$, $exp\ f$	ordem superior

Programação composicional

Sumário

$A \times B$	registos ("structs")
$A + B$	registos variantes ("unions")
$1 + A$	"apontadores"
B^A	"arrays", "streams"

estruturação de dados

$$(1 + a)\,(1 + b) = 1 + a + b + a\,b$$

$$\begin{array}{ccc} (1+A)\times(1+B) & \xrightarrow{\alpha} & (1+A)\times 1 + (1+A)\times B \\ & & \downarrow \beta \\ (1+A)+(B+A\times B) & \xleftarrow{\gamma} & (1\times 1 + A\times 1)+(1\times B + A\times B) \\ \downarrow \delta & & \\ 1+((A+B)+A\times B) & & \end{array}$$