

...Machine Learning...

Artur Ezequiel Nelson

Universidade do Minho

26 de Abril

Indice

- 1 Nelson
- 2 Ezequiel
- 3 Fork e Join
- 4 Operacoes Numericas
- 5 Exemplos
- 6 Generalizar

Nelson

Ezequiel

Fork e Join

Operacoes Numericas

Exemplos

Generalizar

titulo

Nelson

Ezequiel

Fork e Join

Operacoes Numericas

Exemplos

Generalizar

titulo

Fork e Join

- $(\triangle) :: \text{Cartesian } k \Rightarrow (a \text{ 'k' } c) \rightarrow (a \text{ 'k' } d) \rightarrow (a \text{ 'k' } (c \times d))$
- $(\nabla) :: \text{Cartesian } k \Rightarrow (c \text{ 'k' } a) \rightarrow (d \text{ 'k' } a) \rightarrow ((c \times d) \text{ 'k' } a)$

instancia de \rightarrow^+

newtype $a \rightarrow^+ b = \text{AddFun } (a \rightarrow b)$

instance Category (\rightarrow^+) **where**

type Obj (\rightarrow^+) = Additive

id = AddFun id

AddFun $g \circ \text{AddFun } f = \text{AddFun } (g \circ f)$

instance Monoidal (\rightarrow^+) **where**

AddFun $f \times \text{AddFun } g = \text{AddFun } (f \times g)$

instance Cartesian (\rightarrow^+) **where**

exl = AddFun exl

exr = AddFun exr

dup = AddFun dup

instancia de \rightarrow^+

instance Cocartesian (\rightarrow^+) **where**

inl = AddFun inlF

inr = AddFun inrF

jam = AddFun jamF

inlF :: Additive b \Rightarrow a \rightarrow a \times b

inrF :: Additive a \Rightarrow b \rightarrow a \times b

jamF :: Additive a \Rightarrow a \times a \rightarrow a

inlF = $\lambda a \rightarrow (a, 0)$

inrF = $\lambda b \rightarrow (0, b)$

jamF = $\lambda(a, b) \rightarrow a + b$

definição de NumCat

```
class NumCat k a where  
  negateC :: a 'k' a  
  addC :: (a × a) 'k' a  
  mulC :: (a × a) 'k' a  
  ...
```

```
instance Num a ⇒ NumCat (→) a where  
  negateC = negate  
  addC = uncurry (+)  
  mulC = uncurry (·)  
  ...
```


$$D (\text{negate } u) = \text{negate } (D u)$$

$$D (u + v) = D u + D v$$

$$D (u \cdot v) = u \cdot D v + v \cdot D u$$

- Impreciso na natureza de u e v .
- Algo mais preciso seria definir a diferenciação das operações em si.

class Scalable k a **where**

scale :: a \rightarrow (a 'k' a)

instance Num a \Rightarrow Scalable (\rightarrow^+) a **where**

scale a = AddFun ($\lambda da \rightarrow a \cdot da$)

instance NumCat D **where**

negateC = linearD negateC

addC = linearD addC

mulC = D ($\lambda(a, b) \rightarrow (a \cdot b, \text{scale } b \nabla \text{scale } a)$)

Exemplos

Generalizar