

Cálculo de Programas

Lic. C. Computação (2º ano)
Lic./Mest. em Engenharia Informática (3º ano)
UNIVERSIDADE DO MINHO

2021/22 - Ficha nr.º 3

1. Considere o isomorfismo

$$(A + B) + C \begin{array}{c} \xrightarrow{\text{coassocr}} \\ \cong \\ \xleftarrow{\text{coassocl}} \end{array} A + (B + C)$$

onde $\text{coassocr} = [id + i_1, i_2 \cdot i_2]$. Calcule a sua conversa resolvendo em ordem a coassocl a equação,

$$\text{coassocl} \cdot \text{coassocr} = id$$

isto é

$$\text{coassocl} \cdot \underbrace{[id + i_1, i_2 \cdot i_2]}_{\text{coassocr}} = id$$

etc. Finalmente, exprima coassocl sob a forma de um programa em Haskell *não recorra* ao combinador “either”.

2. O combinador

```
const :: a -> b -> a
const a b = a
```

está disponível em Haskell para construir funções constantes, sendo habitual designarmos $\text{const } k$ por \underline{k} , qualquer que seja k . Demonstre a igualdade

$$\underline{(b, a)} = \langle \underline{b}, \underline{a} \rangle \tag{F1}$$

a partir da propriedade universal do produto e das propriedades das funções constantes que constam do formulário.

3. Sabendo que uma dada função xr satisfaz a propriedade

$$xr \cdot \langle \langle f, g \rangle, h \rangle = \langle \langle f, h \rangle, g \rangle \tag{F2}$$

para todo o f, g e h , derivar de (F2) a definição de xr :

$$xr = \langle \pi_1 \times id, \pi_2 \cdot \pi_1 \rangle \tag{F3}$$

4. Use a lei da troca para exprimir o isomorfismo $\text{undistl} = [i_1 \times id, i_2 \times id]$ sob a forma de um ‘split’ de alternativas.

5. No Cálculo de Programas, as definições condicionais do tipo $h\ x = \text{if } p\ x \text{ then } f\ x \text{ else } g\ x$ são escritas usando o combinador ternário $p \rightarrow f, g$, conhecido pelo nome de *condicional de McCarthy*, cuja definição

$$p \rightarrow f, g = [f, g] \cdot p?$$

vem no formulário. Baseando-se em leis deste combinador que constam também do formulário, demonstre a chamada 2ª-lei do condicional de McCarthy:

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

6. Sabendo que as igualdades

$$p \rightarrow k, k = k \tag{F4}$$

$$(p? + p?) \cdot p? = (i_1 + i_2) \cdot p? \tag{F5}$$

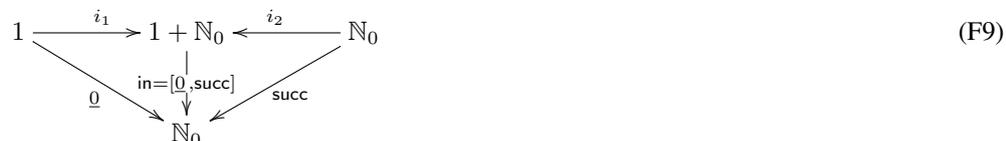
se verificam, demonstre as seguintes propriedades do mesmo combinador:

$$\langle (p \rightarrow f, h), (p \rightarrow g, i) \rangle = p \rightarrow \langle f, g \rangle, \langle h, i \rangle \tag{F6}$$

$$\langle f, (p \rightarrow g, h) \rangle = p \rightarrow \langle f, g \rangle, \langle f, h \rangle \tag{F7}$$

$$p \rightarrow (p \rightarrow a, b), (p \rightarrow c, d) = p \rightarrow a, d \tag{F8}$$

7. Considere a função $\text{in} = [0, \text{succ}]$ que exprime a forma como os números naturais são gerados a partir do número 0, de acordo com o diagrama seguinte,



onde $\text{succ } n = n + 1$. Sabendo que o tipo 1 coincide com o tipo `()` em Haskell e é habitado por um único elemento, também designado por `()`, calcule a inversa de in ,

$$\begin{aligned} \text{out } 0 &= i_1 () \\ \text{out } (n + 1) &= i_2 n \end{aligned}$$

resolvendo em ordem a out a equação $\text{out} \cdot \text{in} = \text{id}$ e introduzindo variáveis. (NB: poderá deste cálculo inferir que in e out são isomorfismos? Justifique.)