

Cálculo de Programas

2.º ano das Licenciaturas em
Engenharia Informática e Ciências da Computação
UNIVERSIDADE DO MINHO

2013/14 - Ficha nr.º 3

1. Tal como acontece com o combinador $\langle f, g \rangle$, também os cálculos sobre o combinador $[f, g]$ se baseiam todos na sua *propriedade universal*,

$$k = [f, g] \equiv \begin{cases} k \cdot i_1 = f \\ k \cdot i_2 = g \end{cases}$$

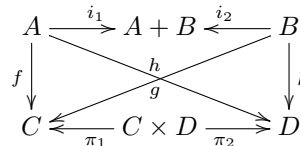
que encontra no formulário disponível na página *web* da disciplina. Use-a para demonstrar a lei

$$[f, g] = [h, k] \equiv \begin{cases} f = h \\ g = k \end{cases}$$

que também consta desse formulário sob a designação *Eq-+*.

2. A *lei da troca* (ver formulário) permite-nos exprimir determinadas funções sob duas formas alternativas, conforme desenhado no respectivo diagrama:

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



Analiticamente, esta lei demonstra-se facilmente resolvendo em ordem a x a equação

$$\langle \langle f, g \rangle, \langle h, k \rangle \rangle = x \tag{1}$$

Faça essa demonstração a partir do esboço seguinte:

$$\begin{aligned} x &= \langle \langle f, g \rangle, \langle h, k \rangle \rangle \\ &\equiv \{ \dots \} \\ &\vdots \\ &\equiv \{ \dots \} \\ &\begin{cases} \pi_1 \cdot x = [f, h] \\ \pi_2 \cdot x = [g, k] \end{cases} \\ &\equiv \{ \dots \} \\ x &= \langle [f, h], [g, k] \rangle \end{aligned}$$

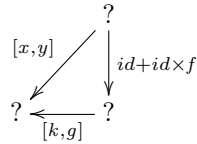
3. Uma função diz-se *constante* sempre que o seu resultado é o mesmo, qualquer que seja o argumento. Por isso se designa uma tal função sublinhando o valor do seu resultado: se este for k , por exemplo, ter-se-á a função $\underline{k} :: a \rightarrow b$, para k um valor de b , que satisfaz sempre a propriedade

$$\underline{k} \cdot f = \underline{k} \tag{2}$$

qualquer que seja k e f . Mostre que $[k, k] = k$.

NB: A função k escreve-se `const k` em Haskell.

4. Complete os “?” no diagrama



e resolva a equação nele implícita em ordem a x e y .

5. Recorra à lei Eq+ (entre outras) para mostrar que a definição que conhece da função factorial,

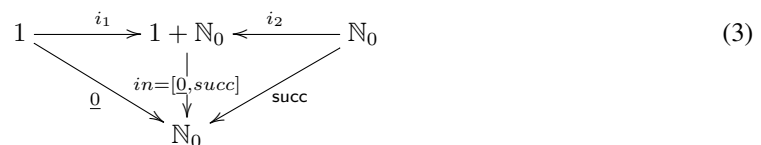
$$\begin{aligned}
 fac\ 0 &= 1 \\
 fac\ (n + 1) &= (n + 1) * fac\ n
 \end{aligned}$$

é equivalente à equação seguinte, em fac

$$fac \cdot [0, succ] = [1, mul \cdot \langle succ, fac \rangle].$$

onde $succ\ n = n + 1$ e $mul\ (a, b) = a * b$.

6. No contexto da questão anterior, considere a função $in = [0, succ]$ que exprime a forma como os números naturais são gerados a partir do número 0, de acordo com o diagrama seguinte:



O tipo 1 coincide com o tipo `()` em Haskell e é habitado por um único elemento, também designado por `()`.

(a) Calcule a inversa de in ,

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

resolvendo em ordem a out a equação $out \cdot in = id$ e introduzindo variáveis.

(b) Mostre que a função $for\ b\ i$, onde

$$\begin{aligned}
 for\ b\ i\ 0 &= i \\
 for\ b\ i\ (n + 1) &= b\ (for\ b\ i\ n)
 \end{aligned}$$

é solução da equação seguinte, em x :

$$x \cdot in = [i, b] \cdot (id + x) \tag{4}$$

Sugestão: proceda como anteriormente, para fac .

7. É dada a equação $[id, i_2] = mu$.

(a) Calcule, resolvendo a equação dada em ordem mu , a seguinte implementação em Haskell:

$$\begin{aligned}
 mu &:: \text{Either } (\text{Either } a\ b)\ b \rightarrow \text{Either } a\ b \\
 mu\ (i_1\ x) &= x \\
 mu\ (i_2\ y) &= i_2\ y
 \end{aligned}$$

desenhando um diagrama explicativo do tipo de mu .

(b) Mostre (por cálculo analítico) que mu satisfaz a propriedade

$$mu \cdot mu = mu \cdot (mu + id)$$