Cálculo de Programas

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

2013/14 - Ficha nr.° 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*,

$$k = [f, g] \quad \equiv \quad \left\{ \begin{array}{l} k \cdot i_1 = f \\ k \cdot i_2 = g \end{array} \right.$$

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

$$[f, g] = [h, k] \quad \equiv \quad \left\{ \begin{array}{l} f = h \\ g = k \end{array} \right.$$

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:

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

$$\begin{tikzcd} A \arrow[r, "i_1"] \arrow[d, "f"'] \arrow[drr, "h" near start] & A + B & B \arrow[l, "i_2"'] \arrow[d, "k"] \arrow[dll, "g" near start] \\ C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D \end{tikzcd}$$

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

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

Faça essa demonstração a partir do esboço seguinte:

$$\begin{array}{cl}
 & x = [\langle f, g\rangle, \langle h, k\rangle] \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & \vdots \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & \left\{ \begin{array}{l} \pi_1 \cdot x = [f, h] \\ \pi_2 \cdot x = [g, k] \end{array} \right. \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & x = \langle [f, h], [g, k]\rangle
\end{array}$$

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

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

qualquer que seja k e f. Mostre que $[\underline{k}, \underline{k}] = \underline{k}$.
NB: A função $\underline{k}$ escreve-se `const k` em Haskell.

4. Complete os "?" no diagrama

$$\begin{tikzcd} & ? \arrow[dl, "{[x,y]}"'] \arrow[d, "id+id\times f"] \\ ? & ? \arrow[l, "{[k,g]}"] \end{tikzcd}$$

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,

$$fac\ 0 = 1$$
$$fac\ (n+1) = (n+1) * fac\ n$$

é equivalente à equação seguinte, em fac

$$fac \cdot [\underline{0}, \mathsf{succ}] = [\underline{1}, mul \cdot \langle \mathsf{succ}, fac \rangle].$$

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

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

$$\begin{tikzcd} 1 \arrow[r, "i_1"] \arrow[dr, "\underline{0}"'] & 1 + \mathbb{N}_0 \arrow[d, "{in=[\underline{0},succ]}"] & \mathbb{N}_0 \arrow[l, "i_2"'] \arrow[dl, "\mathsf{succ}"] \\ & \mathbb{N}_0 & \end{tikzcd} \tag{3}$$

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,

$$out\ 0 = i_1\ ()$$
$$out\ (n+1) = i_2\ n$$

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

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

$$for\ b\ i\ 0 = i$$
$$for\ b\ i\ (n+1) = b\ (for\ b\ i\ n)$$

é solução da equação seguinte, em x:

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

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:

$$mu :: \mathsf{Either}\ (\mathsf{Either}\ a\ b)\ b \rightarrow \mathsf{Either}\ a\ b$$
$$mu\ (i_1\ x) = x$$
$$mu\ (i_2\ y) = i_2\ y$$

desenhando um diagrama explicativo do tipo de mu.

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

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