Cálculo de Programas

2.° ano da Licenciatura em Engenharia Informática da Universidade do Minho

2010/11 - 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*,

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

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* permite-nos exprimir determinadas funções sob duas formas alternativas, conforme desenhado no respectivo diagrama:

$$[\langle f,g\rangle,\langle h,k\rangle] = \langle [f,h],[g,k]\rangle \\ A \xrightarrow{i_1} A + B \xrightarrow{i_2} B \\ f \downarrow k \\ C \xleftarrow{\pi_1} C \times D \xrightarrow{\pi_2} D$$

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

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

Faça essa demonstração a partir do esboço seguinte:

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 $\underline{k} :: a \to k$, que satisfaz sempre a propriedade

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

qualquer que seja k e f. Mostre que $[\underline{k}, \underline{k}] = \underline{k}$.

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

$$\begin{array}{l} \mathit{fac} \ 0 = 1 \\ \mathit{fac} \ (n+1) = (n+1) * \mathit{fac} \ n \end{array}$$

é equivalente à equação seguinte, em fac

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

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

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

$$1 \xrightarrow{i_1} 1 + \mathbb{N}_0 \xleftarrow{i_2} \mathbb{N}_0$$

$$\underbrace{i_1 = [0, succ]}_{\mathbb{N}_0} \text{succ}$$

$$(3)$$

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) Recordando o combinador iterativo

for
$$b \ i \ 0 = i$$

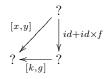
for $b \ i \ (n+1) = b \ (for \ b \ i \ n)$

mostre que, dados b e i, a função $for \ b \ i$ é solução da equação seguinte, em x:

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

Sugestão: proceda como anteriormente, para fac.

6. Complete os "?" no diagrama



e resolva a equação nele implícita em ordem a $x \in y$.

- 7. É dada a equação $[id, i_2] = mu$.
 - (a) Calcule, resolvendo a equação dada em ordem mu, a seguinte implementação em Haskell:

$$mu:$$
 Either (Either $a\ b)\ b o$ 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

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