Cálculo de Programas

2.° ano
Lic. Ciências da Computação e Mestrado Integrado em Engenharia Informática
UNIVERSIDADE DO MINHO

2018/19 - Ficha nr.° 5

1. Sejam dadas, em geral, duas funções f e g satisfazendo a propriedade

$$f\ y = x \quad \equiv \quad y = g\ x \tag{F1}$$

Mostre que f e g são inversas uma da outra — isto é, que as igualdades

$$\left\{ \begin{array}{l} id = g \cdot f \\ f \cdot g = id \end{array} \right.$$

se verificam — e que, portanto, são ambas isomorfismos.

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

$$\begin{tikzcd} 1 \arrow[r, "i_1"] \arrow[dr, "\underline{0}"'] & 1+\mathbb{N}_0 \arrow[d, "\mathsf{in}=[\underline{0}\ ,\mathsf{succ}]"] & \mathbb{N}_0 \arrow[l, "i_2"'] \arrow[dl, "\mathsf{succ}"] \\ & \mathbb{N}_0 & \end{tikzcd} \tag{F2}$$

onde $\mathsf{succ}\ n = n + 1$. 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,

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

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

3. Os diagramas seguintes representam as **propriedades universais** que definem o combinador **catamorfismo** para dois tipos de dados — números naturais $\mathbb{N}_0$ à esquerda e listas finitas A^* à direita:

$$\begin{tikzcd} \mathbb{N}_0 \arrow[d, "(\!|g|\!)"'] & 1+\mathbb{N}_0 \arrow[l, "\mathsf{in}"'] \arrow[d, "id+(\!|g|\!)"] \\ B & 1+B \arrow[l, "g"] \end{tikzcd} \qquad \begin{tikzcd} A^* \arrow[d, "(\!|g|\!)"'] & 1+A\times A^* \arrow[l, "\mathsf{in}"'] \arrow[d, "id+id\times(\!|g|\!)"] \\ B & 1+A\times B \arrow[l, "g"] \end{tikzcd}$$

$$\left\{ \begin{array}{l} \mathsf{in} = [\mathsf{zero}\ , \mathsf{succ}] \\ \mathsf{zero}\ \_ = 0 \\ \mathsf{succ}\ n = n+1 \end{array} \right. \qquad \left\{ \begin{array}{l} \mathsf{in} = [\mathsf{nil}\ , \mathsf{cons}] \\ \mathsf{nil}\ \_ = [\,] \\ \mathsf{cons}\ (a, x) = a : x \end{array} \right.$$

$$k = (\!|g|\!) \ \equiv\ k \cdot \mathsf{in} = g \cdot (id + k) \qquad k = (\!|g|\!) \ \equiv\ k \cdot \mathsf{in} = g \cdot (id + id \times k)$$

$$\mathsf{for}\ b\ i = (\!|[\underline{i}\ , b]|\!) \qquad \mathsf{foldr}\ f\ u = (\!|[\underline{u}\ , \widehat{f}]|\!)$$

onde $\widehat{f}$ abrevia $\mathsf{uncurry} f$.

(a) Mostre, usando a propriedade universal dada em cima, à esquerda, que o catamorfismo de naturais $(a+) = \mathsf{for}\ \mathsf{succ}\ a = (\![\underline{a}\ , \mathsf{succ}]\!)$ se converte na definição[1]

$$a + 0 = a$$
$$a + (n + 1) = 1 + (a + n)$$

(b) Considere agora a operação $a \ominus n$ de subtracção entre um inteiro a e um número natural n:

$$a \ominus 0 = a$$
$$a \ominus (n + 1) = (a \ominus n) - 1$$

Encontre k e g tal que $(a\ominus) = (\![\underline{k}\ , g]\!)$. **Sugestão**: apoie a sua resolução num diagrama.

(c) Use a propriedade universal da esquerda para demonstrar a lei de reflexão $\mathsf{for}\ \mathsf{succ}\ 0 = id$ e a propriedade $\mathsf{for}\ \mathsf{succ}\ 1 = \mathsf{succ}$.

(d) Usando agora a propriedade universal da direita, mostre que $(\![\mathsf{nil}\ , \mathsf{cons}]\!) = id$.

(e) Mostre que as funções

$$f = \mathsf{for}\ id\ i$$

e

$$g = \mathsf{for}\ \underline{i}\ i$$

são a mesma função. (Qual?)

4. Identifique como catamorfismos de listas as funções seguintes, indicando o gene g para cada caso:[2]

(a) k é a função que multiplica todos os elementos de uma lista

(b) $k = reverse$

(c) k é a função $\mathsf{map}\ f$, para um dado $f : A \to B$

(d) k é a função que calcula o máximo de uma lista de números naturais ($\mathbb{N}_0{}^*$).

(e) $k = \mathsf{filter}\ p$ onde

$$\begin{array}{l} \mathsf{filter}\ p\ [\,] = [\,] \\ \mathsf{filter}\ p\ (h : t) = x +\!\!+\ \mathsf{filter}\ p\ t \\ \quad \mathbf{where}\ x = \mathbf{if}\ (p\ h)\ \mathbf{then}\ [h]\ \mathbf{else}\ [\,] \end{array}$$

5. Poderá a definição da função factorial

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

ser vista *directamente* como um catamorfismo de números naturais (i.e. ciclo-for)? Justifique.

6. A função $k = \mathsf{for}\ f\ i$ pode ser codificada em sintaxe C escrevendo

```
int k(int n) {
  int r=i;
  int j;
  for (j=1;j<n+1;j++) {r=f(r);}
  return r;
};
```

Escreva em sintaxe C as funções $(a+)$ da alínea 2a acima, e f e g da alínea 2e acima.

[1] Repare que esta função mais não faz do que usar duas propriedades da adição de números – quais?

[2] Apoie a sua resolução com diagramas.