Cálculo de Programas

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

2014/15 - Ficha nr.° 5

1. Seja dada uma função ∇ da qual só sabe duas propriedades: $\nabla \cdot i_1 = id$ e $\nabla \cdot i_2 = id$. Mostre que, necessariamente, ∇ satisfaz também a propriedade natural $f \cdot \nabla = \nabla \cdot (f + f)$.

2. Faça a inferência do tipo polimórfico principal (isto é, mais geral) da função $\langle \pi_1 + \pi_1, [\pi_2\ , \pi_2] \rangle$ através de um diagrama e deduza daí a respectiva propriedade natural.

3. Calcule analiticamente a propriedade natural da função $twist = \mathsf{swap} \cdot (id \times \mathsf{swap})$ recorrendo explicitamente às propriedades naturais de id e de swap.

4. 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:

$$\begin{tikzcd} \mathbb{N}_0 \arrow[d, "\llparenthesis g \rrparenthesis"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathsf{in}"'] \arrow[d, "id + \llparenthesis g \rrparenthesis"] \\ B & 1 + B \arrow[l, "g"] \end{tikzcd} \qquad \begin{tikzcd} {[A]} \arrow[d, "\llparenthesis g \rrparenthesis"'] & 1 + A \times [A] \arrow[l, "\mathsf{in}"'] \arrow[d, "id + id \times \llparenthesis g \rrparenthesis"] \\ B & 1 + A \times B \arrow[l, "g"] \end{tikzcd}$$

$$\mathsf{in} = [\mathsf{zero}\ , \mathsf{succ}] \qquad \mathsf{in} = [\mathsf{nil}\ , \mathsf{cons}]$$
$$\mathsf{zero}\ \_ = 0 \qquad \mathsf{nil}\ \_ = [\,]$$
$$\mathsf{succ}\ n = n + 1 \qquad \mathsf{cons}\ (a, x) = a : x$$

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

$$\mathsf{for}\ b\ i = \llparenthesis [\underline{i}\ , b] \rrparenthesis \qquad \mathsf{foldr}\ f\ u = \llparenthesis [\underline{u}\ , \widehat{f}] \rrparenthesis$$

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

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

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

(c) Mostre que as funções

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

e

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

são a mesma função. (Qual?)

(d) Identifique como catamorfismos de listas as funções seguintes, indicando o gene g para cada caso:

- f é a função que multiplica todos os elementos de uma lista
- $f = reverse$
- f é a função que implementa o algoritmo de ordenação de listas por inserção ('insertion sort').
- A função $f = \mathsf{filter}\ p$ tal que

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

5. Poderá a função que calcula o valor *máximo* de uma lista de inteiros ser escrita da forma $(\!|g|\!)$, para um dado g a definir?

6. Mostre que $(\!|[\underline{i}\ , \pi_2]|\!) = \underline{i}$. Qual das duas propriedades universais acima usou no seu raciocínio?

7. Mostre que

$$\begin{array}{l}\mathsf{map} :: (a \to b) \to [a] \to [b]\\ \mathsf{map}\, f\ [\,] = [\,]\\ \mathsf{map}\, f\ (h : t) = (f\ h) : \mathsf{map}\, f\ t\end{array}$$

se reduz à equação

$$(\mathsf{map}\, f) \cdot \mathsf{in} \quad = \quad \mathsf{in} \cdot (id + f \times id) \cdot (id + id \times (\mathsf{map}\, f))$$

e que, portanto, $\mathsf{map}\, f = (\!|\mathsf{in} \cdot (id + f \times id)|\!)$.

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

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

Escreva em C as funções f e g da alínea 4c acima, bem como aquela que decorre da lei de reflexão da alínea 4a.

9. Considere a função, em Haskell

$$\begin{array}{l}g\ (\mathsf{Leaf}\ a) = \mathsf{Leaf}\ (\mathsf{succ}\ a)\\ g\ (\mathsf{Fork}\ (x, y)) = \mathsf{Fork}\ (g\ x, g\ y)\end{array}$$

definida sobre uma instância do tipo de dados

$$\mathbf{data}\ \mathsf{LTree}\ a = \mathsf{Leaf}\ a \mid \mathsf{Fork}\ (\mathsf{LTree}\ a, \mathsf{LTree}\ a)$$

do qual se infere a existência da função

$$inLTree = [\mathsf{Leaf}\ , \mathsf{Fork}]$$

É possível mostrar que g satisfaz a equação

$$g \cdot inLTree = inLTree \cdot k \cdot (id + g \times g)$$

para uma dada função k. Calcule k.
Sugestão: retire as variáveis a, x e y à definição dada, converta-a numa igualdade *pointfree* e compare o resultado como a equação acima.