Cálculo de Programas

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

2020/21 - Ficha nr.° 5

1. Considere a função $\mathsf{iso} = \langle ! + !, [id\ , id] \rangle$, onde $! : A \to 1$ designa a única função constante que habita o tipo $A \to 1$.[1]

   (a) Identifique o isomorfismo que ela testemunha, desenhando-a sob a forma de um diagrama de tipos.

   (b) Derive a partir desse diagrama a propriedade (dita *grátis*) de iso,

   $$(id \times f) \cdot \mathsf{iso} \quad = \quad \mathsf{iso} \cdot (f + f) \tag{F1}$$

   (c) Confirme, por cálculo analítico, essa propriedade.

   (d) Derive uma definição em Haskell *pointwise* de iso.

2. Deduza o tipo mais geral da função $\alpha = (id + \pi_1) \cdot i_2 \cdot \pi_2$ e infira a respectiva propriedade *grátis* (*natural*) através de um diagrama.

3. Identifique, apoiando a sua resolução num diagrama, qual é a definição da função polimórfica α cuja propriedade natural ("grátis") é

   $$(f + h) \cdot \alpha \quad = \quad \alpha \cdot (f + g \times h)$$

4. Para o caso de um *isomorfismo* α, têm-se as equivalências:

   $$\alpha \cdot g = h \ \equiv \ g = \alpha^\circ \cdot h \tag{F2}$$
   $$g \cdot \alpha = h \ \equiv \ g = h \cdot \alpha^\circ \tag{F3}$$

   Recorra a essas propriedades para mostrar que a igualdade

   $$h \cdot \mathsf{distr} \cdot (g \times (id + \alpha)) = k$$

   é equivalente à igualdade

   $$h \cdot (g \times id + g \times \alpha) = k \cdot \mathsf{undistr}$$

   (**Sugestão:** não ignore a propriedade natural (i.e. *grátis*) do isomorfismo distr.)

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

[1] A função $! : A \to 1$ costuma-se designar-se também por função *"bang"*.

6. O diagrama seguinte representa o combinador *catamorfismo* (de naturais) que se começou a estudar na última aula teórica, onde a notação $\llparenthesis\, g \,\rrparenthesis$ abrevia $cata\ g$ então usada:

$$\begin{tikzcd} \mathbb{N}_0 \arrow[d, "\llparenthesis\, g \,\rrparenthesis"'] & 1 + \mathbb{N}_0 \arrow[l, "\mathsf{in}"'] \arrow[d, "id + \llparenthesis\, g \,\rrparenthesis"] \\ B & 1 + B \arrow[l, "g"] \end{tikzcd} \quad \text{onde} \quad \left\{ \begin{array}{l} \mathsf{in} = [\mathsf{zero}\ , \mathsf{succ}] \\ \mathsf{zero}\ \_ = 0 \\ \mathsf{succ}\ n = n + 1 \end{array} \right.$$

Assumindo a seguinte propriedade universal desse combinador,

$$k = \llparenthesis\, g \,\rrparenthesis \ \equiv\ k \cdot \mathsf{in} = g \cdot (id + k) \tag{F4}$$

mostre que o combinador "ciclo-for"definido por

$$\mathsf{for}\ b\ i = \llparenthesis\, [\underline{i}\ , b] \,\rrparenthesis$$

se converte na seguinte versão "pointwise":

$$\begin{array}{l} \mathsf{for}\ b\ i\ 0 = i \\ \mathsf{for}\ b\ i\ (n+1) = b\ (\mathsf{for}\ b\ i\ n) \end{array}$$

7. Mostre, usando (F4), que o catamorfismo de naturais $(a+) = \mathsf{for}\ \mathsf{succ}\ a = \llparenthesis\, [\underline{a}\ , \mathsf{succ}] \,\rrparenthesis$ se converte na definição[2]

$$\begin{array}{l} a + 0 = a \\ a + (n+1) = 1 + (a + n) \end{array}$$

8. Mostre que as funções $f = \mathsf{for}\ id\ i$ e $g = \mathsf{for}\ \underline{i}\ i$ são a mesma função. (Qual?)

9. Sabendo que $\mathsf{for}\ f\ i = \llparenthesis\, [\underline{i}\ , f] \,\rrparenthesis$ para $\mathsf{F}\ f = id + f$ (naturais), recorra à lei de fusão-cata para demonstrar a propriedade:[3]

$$f \cdot (\mathsf{for}\ f\ i) = \mathsf{for}\ f\ (f\ i) \tag{F5}$$

10. 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 $(a*) = \mathsf{for}\ (a+)\ 0$ e outros catamorfismos de naturais de que se tenha falado nas aulas da disciplina.

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

[3] Como complemento desta questão, escreva em sintaxe C os programas correspondentes aos dois lados da igualdade e compare-os informalmente.