

Cálculo de Programas

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

2014/15 - Ficha nr.º 8

1. Considere o tipo genérico das árvores binárias com informação nas folhas

`data LTree a = Leaf a | Fork (LTree a, LTree a)`

e a função

`mirror (Leaf a) = Leaf a`
`mirror (Fork (x, y)) = Fork (mirror y, mirror x)`

que “espelha” árvores binárias desse tipo.

(a) Mostre que

$$\text{mirror} = (\text{inLTree} \cdot (\text{id} + \text{swap})) \quad (\text{F1})$$

para $\text{inLTree} = [\text{Leaf}, \text{Fork}]$ e desenhe o digrama que representa este catamorfismo.

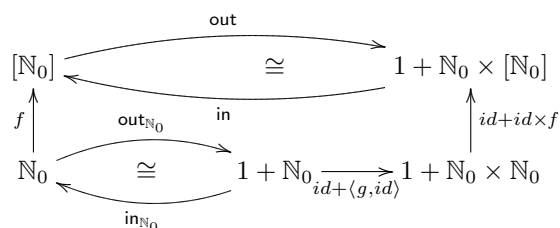
(b) Tal como swap , mirror é um isomorfismo de árvores pois é a sua própria inversa:

$$\text{mirror} \cdot \text{mirror} = \text{id} \quad (\text{F2})$$

Complete a seguinte demonstração de (F2):

$$\begin{aligned} & \text{mirror} \cdot \text{mirror} = \text{id} \\ \equiv & \quad \{ \dots \} \\ & \text{mirror} \cdot (\text{inLTree} \cdot (\text{id} + \text{swap})) = (\text{inLTree}) \\ \Leftarrow & \quad \{ \dots \} \\ & \text{mirror} \cdot \dots = \dots \\ \dots & \quad \{ \dots \} \\ & (\text{etc}) \end{aligned}$$

2. Assumindo as definições $\text{in}_{\mathbb{N}_0} = [\underline{0}, \text{succ}]$, $\text{in} = [\text{nil}, \text{cons}]$, $\text{succ } n = n + 1$, $\text{nil} = []$ e $\text{cons } (h, t) = h : t$, mostre que o diagrama



define o *anamorfismo* de listas

$$f = \llbracket (id + \langle g, id \rangle) \cdot out_{\mathbb{N}_0} \rrbracket$$

e diga o que faz este anamorfismo para $g = succ \cdot (2*)$.

3. A função

$$\begin{aligned} \text{map } f \ [] &= [] \\ \text{map } f \ (h : t) &= (f \ h) : \text{map } f \ t \end{aligned}$$

é o catamorfismo de listas $\text{map } f = \llbracket in \cdot (id + f \times id) \rrbracket$, para $in = [\text{nil}, \text{cons}]$. Mostre que a mesma função se pode escrever como um anamorfismo:

$$\text{map } f = \llbracket (id + f \times id) \cdot out \rrbracket \tag{F3}$$

4. Mostre que o anamorfismo de listas

$$\text{suffixes} = \llbracket g \rrbracket \text{ where } g = (id + \langle \text{cons}, \pi_2 \rangle) \cdot out$$

é a função

$$\begin{aligned} \text{suffixes} \ [] &= [] \\ \text{suffixes} \ (h : t) &= (h : t) : \text{suffixes } t \end{aligned}$$

5. O algoritmo da divisão inteira,

$$\begin{aligned} m \div n \\ | \ m < n = 0 \\ | \ \text{otherwise} = 1 + (m - n) \div n \end{aligned}$$

corresponde ao anamorfismo de naturais $(\div n) = \llbracket g \ n \rrbracket$ em que $g \ n = (\lt n) \rightarrow (i_1 \cdot !), (i_2 \cdot (-n))$. É de esperar que o algoritmo dado satisfaça a propriedade $(m * n) \div n = m$, isto é, $(\div n) \cdot (*n) = id$. Complete a prova seguinte dessa propriedade, que se baseia na lei de fusão dos anamorfismos (identifique-a no formulário):

$$\begin{aligned} &(\div n) \cdot (*n) = id \\ \equiv &\{ \dots \} \\ &\llbracket g \ n \rrbracket \cdot (*n) = \llbracket out \rrbracket \\ \Leftarrow &\{ \dots \} \\ &(g \ n) \cdot (*n) = (id + (*n)) \cdot out \\ \equiv &\{ \dots \} \\ &((\lt n) \cdot (*n)) \rightarrow (i_1 \cdot !), (i_2 \cdot (-n)) \cdot (*n) = (id + (*n)) \cdot out \\ \equiv &\{ \dots \} \\ & (= 0) \rightarrow (i_1 \cdot !), (i_2 \cdot (*n) \cdot pred) = (id + (*n)) \cdot out \\ \equiv &\{ \dots \} \\ &(id + (*n)) \cdot out = (id + (*n)) \cdot out \\ \equiv &\{ \dots \} \\ &true \end{aligned}$$

6. Mostre que a função *mirror* da primeira questão desta ficha se pode definir como o anamorfismo

$$\text{mirror} = \llbracket (id + \text{swap}) \cdot outLTree \rrbracket \tag{F4}$$

onde *outLTree* é a conversa de *inLTree* e volte a demonstrar (F2), desta vez recorrendo à lei de fusão dos anamorfismos.