

Cálculo de Programas

2.º ano

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

2020/21 - Ficha nr.º 10

1. O formulário desta disciplina apresenta duas definições alternativas para o functor $T f$ de um tipo indutivo, uma como *catamorfismo* e outra como *anamorfismo*. Identifique-as e acrescente justificações à seguinte prova de que essas definições são equivalentes:

$$\begin{aligned} T f &= \llbracket \text{in} \cdot B(f, id) \rrbracket \\ \equiv & \{ \dots \} \\ T f \cdot \text{in} &= \text{in} \cdot B(f, id) \cdot F(T f) \\ \equiv & \{ \dots \} \\ T f \cdot \text{in} &= \text{in} \cdot B(id, T f) \cdot B(f, id) \\ \equiv & \{ \dots \} \\ \text{out} \cdot T f &= F(T f) \cdot B(f, id) \cdot \text{out} \\ \equiv & \{ \dots \} \\ T f &= \llbracket B(f, id) \cdot \text{out} \rrbracket \\ \square \end{aligned}$$

2. Mostre que o catamorfismo de listas $\text{length} = \llbracket [\text{zero}, \text{succ} \cdot \pi_2] \rrbracket$ é a mesma função que o *anamorfismo* de naturais $\llbracket (id + \pi_2) \cdot \text{out}_{\text{List}} \rrbracket$.
3. Mostre que o anamorfismo que calcula os sufixos de uma lista

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

é a função

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

4. Mostre que a função *mirror* da ficha nr.º 7 se pode definir como o anamorfismo

$$\text{mirror} = \llbracket (id + \text{swap}) \cdot \text{out} \rrbracket \tag{F1}$$

onde *out* é a conversa de *in*. Volte a demonstrar a propriedade $\text{mirror} \cdot \text{mirror} = id$, desta vez recorrendo à lei de fusão dos anamorfismos.

5. Nas aulas teóricas viu-se que, sempre que um ciclo-*while* termina, ele pode ser definido por

$$\mathbf{while} \ p \ f \ g = \mathbf{tailr} \ ((g + f) \cdot (\neg \cdot p)?) \tag{F2}$$

recorrendo ao combinador de “tail recursion” $\mathbf{tailr} \ f = \llbracket \nabla, f \rrbracket$, que é um hilomorfismo de base $B \ (X, Y) = X + Y$, para $\nabla = [id, id]$.

- (a) Derive a definição *pointwise* de $\mathbf{while} \ p \ f \ g$, sabendo que qualquer $h = \llbracket f, g \rrbracket$ é tal que $h = f \cdot F \ h \cdot g$.
- (b) Complete a demonstração da lei de fusão de \mathbf{tailr} ¹

$$(\mathbf{tailr} \ g) \cdot f = \mathbf{tailr} \ h \iff (id + f) \cdot h = g \cdot f$$

que se segue:

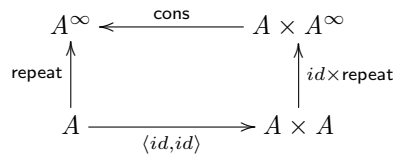
$$\begin{aligned} & (\mathbf{tailr} \ g) \cdot f = \mathbf{tailr} \ h \\ \equiv & \quad \{ \dots\dots\dots \} \\ & (\nabla) \cdot \llbracket g \rrbracket \cdot f = (\nabla) \cdot \llbracket h \rrbracket \\ \Leftarrow & \quad \{ \dots\dots\dots \} \\ & \llbracket g \rrbracket \cdot f = \llbracket h \rrbracket \\ \Leftarrow & \quad \{ \dots\dots\dots \} \\ & g \cdot f = (id + f) \cdot h \\ & \square \end{aligned}$$

6. O algoritmo de “quick-sort” foi definido nas aulas teóricas como o hilomorfismo $qSort = (\llbracket inord \rrbracket) \cdot \llbracket qsep \rrbracket$ sobre árvores BTree, cujo catamorfismo recorre à função:

$$inord = [nil, f] \ \mathbf{where} \ f \ (x, (l, r)) = l \ ++ \ [x] \ ++ \ r$$

Para um certo isomorfismo α , $h = (\llbracket inord \cdot \alpha \rrbracket) \cdot \llbracket qsep \rrbracket$ ordenará a lista de entrada por ordem inversa. Identifique α , justificando informalmente.

7. Mostre que o anamorfismo $repeat = \llbracket \langle id, id \rangle \rrbracket$ definido pelo diagrama



é a função:

$$repeat \ a = a : repeat \ a$$

De seguida, recorrendo às leis dos anamorfismos mostre que, apesar de não terminar², $repeat$ satisfaz a propriedade:³

$$\mathbf{map} \ f \cdot \mathbf{repeat} = \mathbf{repeat} \cdot f \tag{F3}$$

¹**NB:** Assume-se que $(\mathbf{tailr} \ g) \cdot f$ termina.
²Por isso usamos, no diagrama, A^∞ em vez de A^* , para incluir também as listas infinitas.
³“Verifique” este facto comparando, por exemplo, $(take \ 10 \cdot \mathbf{map} \ succ \cdot \mathbf{repeat}) \ 1$ com $(take \ 10 \cdot \mathbf{repeat} \cdot \mathbf{succ}) \ 1$.