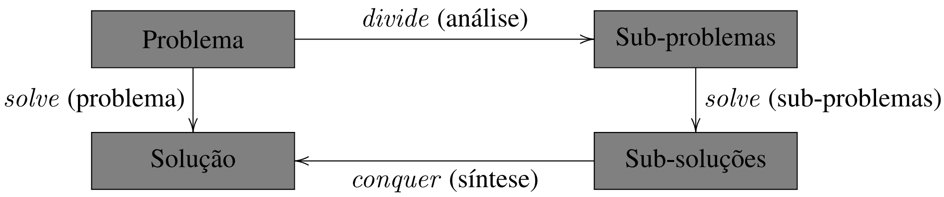
Cálculo de Programas

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

2016/17 - Ficha nr.° 10

1. O desenho que se segue



descreve aquela que é talvez a principal competência de um bom programador: a capacidade de dividir um problema complexo em partes e a de saber juntar as respectivas sub-soluções para assim resolver o problema inicial.

No Cálculo de Programas, o esquema desenhado acima é captado pelo conceito de *hilomorfismo*,

$$solve = conquer \cdot (\mathsf{F}\ solve) \cdot divide \qquad \begin{tikzcd} A \arrow[r, "divide"] \arrow[d, "solve"'] & \mathsf{F}\ A \arrow[d, "\mathsf{F}\ solve"] \\ B & \mathsf{F}\ B \arrow[l, "conquer"] \end{tikzcd} \tag{F1}$$

que normalmente se escreve $solve = [\![conquer, divide]\!]$ e se factoriza na composição do anamorfismo $[\!(divide)\!]$ com o catamorfismo $(\!| conquer |\!)$,

$$[\![conquer, divide]\!] = (\!| conquer |\!) \cdot [\!(divide)\!] \qquad \begin{tikzcd} A \arrow[r, "divide"] \arrow[d, "{[\!(divide)\!]}"] \arrow[dd, bend right=60, "solve"'] & \mathsf{F}\ A \arrow[d, "\mathsf{F}{[\!(divide)\!]}"'] \arrow[dd, bend left=60, "\mathsf{F}\ solve"] \\ \mathsf{T} \arrow[r, bend left, "\mathsf{out}"] \arrow[d, "{(\!| conquer |\!)}"] \arrow[r, phantom, "\cong"] & \mathsf{F}\ \mathsf{T} \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "\mathsf{F}{(\!| conquer |\!)}"'] \\ B & \mathsf{F}\ B \arrow[l, "conquer"] \end{tikzcd} \tag{F2}$$

mediados por uma estrutura de dados do tipo T associado ao functor F.[1]

[1] Esta estrutura intermédia é designada normalmente por estrutura de dados **virtual** por "não ser ver" quando o algoritmo executa, ficando escondida no 'heap' do sistema de 'run-time' da linguagem recursiva que está a ser utilizada.

Apresente justificações para os passos seguintes da demonstração do princípio da *hilo-factorização*, isto é, da equivalência entre (F1) e (F2):

$$
\begin{array}{ll}
& solve = \llbracket conquer, divide \rrbracket \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& solve = (\!| conquer |\!) \cdot [\!(divide)\!] \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& solve = (conquer \cdot \mathsf{F}\,(\!| conquer |\!) \cdot \mathsf{out}) \cdot (\mathsf{in} \cdot \mathsf{F}\,[\!(divide)\!] \cdot divide) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& solve = conquer \cdot \mathsf{F}\,(\!| conquer |\!) \cdot \mathsf{F}\,[\!(divide)\!] \cdot divide \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& solve = conquer \cdot \mathsf{F}\,((\!| conquer |\!) \cdot [\!(divide)\!]) \cdot divide \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& solve = conquer \cdot \mathsf{F}\, solve \cdot divide
\end{array}
$$

2. A função

$$f = p \rightarrow g, (h \cdot f \cdot k)$$

é o hilomorfismo $f = \llbracket [g\ , h], p \rightarrow i_1, (i_2 \cdot k) \rrbracket$. Desenhe o diagrama (F2) correspondente, identificando o tipo intermédio T.

3. Um ciclo-while é um hilomorfismo:

$$
\begin{array}{l}
\textbf{while} :: (a \rightarrow \mathbb{B}) \rightarrow (a \rightarrow a) \rightarrow a \rightarrow a \\
\textbf{while}\ p\ f = w\ \textbf{where}\ w = [id\ , id] \cdot (w + id) \cdot (f + id) \cdot p?
\end{array}
$$

Repita o exercício anterior para este combinador e verifique que se tem

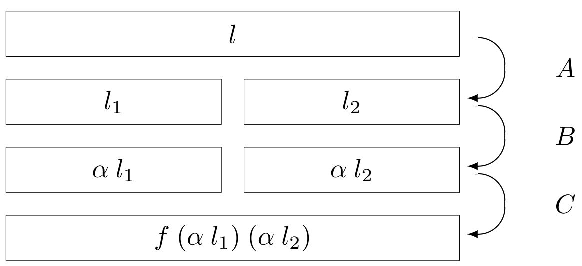
$$\textbf{while}\ p\ f\ x = \textbf{if}\ p\ x\ \textbf{then}\ \textbf{while}\ p\ f\ (f\ x)\ \textbf{else}\ x$$

4. Mostre que a função mirror da ficha anterior se pode definir como o anamorfismo

$$\mathsf{mirror} = [\!((id + \mathsf{swap}) \cdot \mathsf{out})\!] \qquad \text{(F3)}$$

onde out é a conversa de in e volte a demonstrar (F2) dessa ficha, desta vez recorrendo à lei de fusão dos anamorfismos.

5. O desenho abaixo pretende descrever graficamente um algoritmo de ordenação $A^* \xrightarrow{\alpha} A^*$ que conhece:



(a) Identifique α, bem como as suas fases A, B e C e a função f. Codifique esta última em Haskell.

(b) Descreva o mesmo algoritmo sob a forma de um hilomorfismo de um particular tipo indutivo estudado nas aulas desta disciplina.