

Cálculo de Programas

3/2.º Ano de LEI/MiEI (Universidade do Minho)
Ano Lectivo de 2023/24

Exame de Recurso — 15 de Janeiro de 2024, 14h00–16h00
Salas 1.03 + 1.07 do Edifício 2.

PROVA PRESENCIAL INDIVIDUAL SEM CONSULTA (2h)

Importante — *Ler antes de iniciar a prova:*

- *Esta prova consta de 8 questões que valem, cada uma, 2.5 valores. O tempo médio estimado para resolução de cada questão é de 15 min.*
- *Recomenda-se que os alunos leiam a prova antes de decidirem por que ordem querem responder às questões que são colocadas.*

Questão 1 Recorra às leis que conhece dos produtos, coprodutos e funções constantes para demonstrar a igualdade:

$$[(f, k), (g, l)] = [f, g], l \quad (\text{E1})$$

Questão 2 Infira o tipo mais geral da função α definida por

$$\alpha = [(id, nil), id \times singl] \quad (\text{E2})$$

onde $nil _ = []$ e $singl x = [x]$. De seguida, derive desse tipo a propriedade grátis de α .

Questão 3 Seja dada uma função de ordem superior α que satisfaz a propriedade:

$$\widehat{\alpha f} = \widehat{f} \cdot swap \quad (\text{E3})$$

Mostre, usando as propriedades da exponenciação, que

$$\alpha (\alpha f) = f$$

se verifica.

Questão 4 A seguinte questão apareceu num teste anterior desta disciplina:

Considere a estrutura de dados que se segue, estudada nas aulas:

Árvores com informação de tipo A nos nós :

$$T = \text{BTree } A \quad \begin{cases} B(X, Y) = 1 + X \times Y^2 \\ B(g, f) = id + g \times f^2 \end{cases} \quad \text{in} = [\text{Empty}, \text{Node}]$$

Haskell: `data BTree a = Empty | Node (a, (BTree a, BTree a))`

Defina como um catamorfismo a função

$$f : \text{BTree } A \rightarrow A^* \\ f = (g)$$

isto é, identifique g tal que $f t$ seja o caminho a percorrer na árvore t para atingir o seu nó terminal mais à direita.

Uma das respostas que apareceram ao corrigir esse teste foi

$$g = [\text{nil}, \pi_1 \cdot \pi_2] \tag{E4}$$

Mostre que $f = \llbracket [\text{nil}, \pi_1 \cdot \pi_2] \rrbracket$ é uma função constante, calculando-a usando as leis do cálculo de programas.

Questão 5 Mostre que o catamorfismo

$$k = \llbracket (m \times n) \cdot \langle F \pi_2, F \pi_1 \rangle \rrbracket \tag{E5}$$

se pode decompor em duas funções f e g tal que:

$$\begin{cases} f \cdot \text{in} = m \cdot F g \\ g \cdot \text{in} = n \cdot F f \end{cases}$$

Sugestão: resolva em ordem a f e g a equação

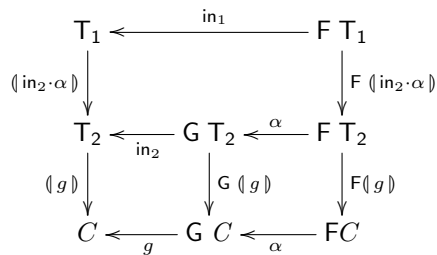
$$\langle f, g \rangle = k$$

onde k é o catamorfismo dado.

Questão 6 Considere a seguinte generalização de um resultado que foi abordado nas aulas teórico-práticas,

$$\llbracket g \rrbracket \cdot \llbracket \text{in}_2 \cdot \alpha \rrbracket = \llbracket g \cdot \alpha \rrbracket \iff G f \cdot \alpha = \alpha \cdot F f \tag{E6}$$

a que corresponde o diagrama:



Apresente uma demonstração analítica de (E6).

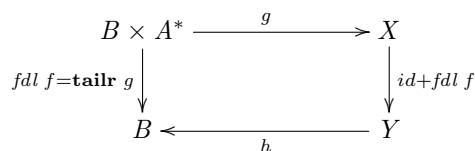
Questão 7 Conheça com certeza, de Programação Funcional, a função:

$$\begin{cases} \text{foldl} :: (b \rightarrow a \rightarrow b) \rightarrow b \rightarrow [a] \rightarrow b \\ \text{foldl } f \ z \ [] = z \\ \text{foldl } f \ z \ (h : t) = \text{foldl } f \ (f \ z \ h) \ t \end{cases}$$

Pode mostrar-se que $\widehat{\text{foldl}} f = \text{fdl } f$ onde

$$\begin{cases} \text{fdl } f \ (z, []) = z \\ \text{fdl } f \ (z, h : t) = \text{fdl } f \ (f \ z \ h, t) \end{cases}$$

Pretendendo-se agora mostrar que $\text{fdl } f$ é um “ciclo-while disfarçado”, para $f : B \rightarrow A \rightarrow B$, encontre g, h, X e Y no diagrama



por forma a se ter $\text{fdl } f = \text{tailr } g$. Justifique a sua resposta. **NB:** recorde a definição $\text{tailr } f = \llbracket [id, id], f \rrbracket$.

Questão 8 Nas aulas mostrou-se que qualquer functor de tipo $T A \cong A + F (T A)$ é um monade, para um dado F . Deu-se como exemplo o monade $LTree A$, para o qual $F X = X \times X$. Ora, dá-se o facto deste functor F ser também um monade, cf.

$$A \xrightarrow{u=\langle id, id \rangle} F A \xleftarrow{\mu=\pi_1 \times \pi_2} F (F A) \quad (E7)$$

e de se poder usar essa informação para extrair $F A$ a partir de $T A$, de forma imediata:

$$\begin{aligned} f &: T A \rightarrow F A \\ f &= ([\langle id, id \rangle, \pi_1 \times \pi_2]) \end{aligned} \quad (E8)$$

Com base nos detalhes que abaixo se dão sobre $LTree$, converta o catamorfismo f para Haskell sem quaisquer combinadores *pointfree* e diga, através de um exemplo, o que a função f “faz”.

Árvores com informação de tipo A nas folhas :

$$\begin{aligned} T = LTree A & \quad \begin{cases} B (X, Y) = X + Y \times Y \\ B (g, f) = g + f \times f \end{cases} \quad \text{in} = [Leaf, Fork] \\ \text{Haskell: data } LTree a &= Leaf a \mid Fork (LTree a, LTree a) \end{aligned}$$