Métodos Formais de Programação I +
Opção I - Métodos Formais de Programação I

4.º Ano da LMCC (7007N2) + LESI (5307P2)
Ano Lectivo de 191999/2000

Exame (2.ª chamada) — 9 de Fevereiro de 2000
09h30
Sala 206

NB: *Esta prova consta de **10** alíneas que valem, cada uma, 2 valores. Para sua consulta, encontra anexa a esta prova a listagem das principais leis de cálculo estudadas na disciplina.*

PROVA SEM CONSULTA (3 $horas$)

Questão 1 Considere a seguinte assinatura associada a filas de espera ('queues'):

$$\Sigma = \left\{ \begin{array}{rcll} start & : & \longrightarrow Queue & \textit{/*inicialização de uma fila */} \\ enq & : & Elem \times Queue \longrightarrow Queue & \textit{/*chegada de um elemento à fila */} \\ next & : & Queue \longrightarrow Elem & \textit{/*próximo elemento a ser atendido */} \\ deq & : & Queue \longrightarrow Queue & \textit{/*retirada de um elemento para ser atendido */} \\ empty? & : & Queue \longrightarrow Bool & \textit{/*teste se a fila está vazia */} \end{array} \right.$$

No seguinte modelo de Σ, uma fila é representada por um par $\langle r, f \rangle$ de sequências, onde f representa a parte da frente da fila e r representa a rectaguarda da fila, listada por ordem inversa:

$$\mathcal{A} = \left\{ \begin{array}{rcl} Queue & = & Elem^\star \times Elem^\star \\ startx() & \stackrel{\text{def}}{=} & \ldots \\ enq(e, \langle r, f \rangle) & \stackrel{\text{def}}{=} & \langle cons(e, r), f \rangle \\ next\, q & \stackrel{\text{def}}{=} & \ldots \\ deq\, q & \stackrel{\text{def}}{=} & \ldots \\ empty?\, \langle r, f \rangle & \stackrel{\text{def}}{=} & \textit{length}\ (f \frown r) = 0 \end{array} \right.$$

1. Mostre que a definição de enq se pode escrever, em notação sem variáveis, da forma seguinte: $enq \stackrel{\text{def}}{=} (cons \times id) \cdot assocl$.
2. Complete as rectic̨ências nas definições de $start$, $next$ e de deq.

Questão 2 Na linguagem de programação LISP existe um único suporte para representação de informação, designado por *expressão-S* — abreviatura de "expressão simbólica" — de acordo com a sintaxe seguinte:

$$\langle\text{Expressão-S}\rangle \quad ::= \quad \langle\text{Átomo}\rangle \mid (\langle\text{Expressão-S}\rangle^\star) \tag{1}$$

onde se considera um $\langle$Átomo$\rangle$ toda a unidade de informação indivisível, não-estruturada (*i.é* "atómica"). Por exemplo, são átomos os inteiros e os 'strings' alfanuméricos, *e.g.* `10`, `-5`, `x12`, `Sun3@160`. Dão-se a seguir exemplos de $\langle$Expressão-S$\rangle$ não atómicas:

```
()
(1)
(1 um 2 dois)
(1 (2 (3 (4))))
```

1. Descreva $\langle$Expressão-S$\rangle$ como um tipo indutivo paramétrico em $\langle$Átomo$\rangle$ e construa o diagrama genérico dos seus catamorfismos.

2. Indique qual o "gene" g do catamorfismo $(\!|g|\!)$ que calcula o número de átomos que ocorrem numa $\langle$Expressão-S$\rangle$, isto é, que é tal que, por exemplo, $(\!|g|\!)$`(1 um 2 dois)` $= 4$.

Questão 3 Mostre como converter a seguinte expressão num único `if-then-else` aplicando leis de cálculo que conhece:

```
k(if p(x) then f(x) else h(x), if p(x) then g(x) else i(x))
```

Questão 4 Considere a definição seguinte da função factorial

$$\begin{array}{rcl} fac\,0 & = & 1 \\ fac(n+1) & = & (n+1) \times fac\,n \end{array}$$

Repare que o padrão recursivo desta função, habitualmente designado por *recursão primitiva*, é um tudo nada mais elaborado do que o habitual catamorfismo sobre $I\!N$.

1. Verifique que o seguinte diagrama comuta:

$$\begin{tikzcd} I\!N & 1 + (I\!N \times I\!N) \arrow[l, "f"'] \\ I\!N \arrow[u, "fac"] & 1 + I\!N \arrow[l, "in"] \arrow[u, "id+\langle id, fac\rangle"'] \end{tikzcd}$$

   onde $in = [\underline{0}, succ]$, $f = [\underline{1}, mul \cdot (succ \times id)]$ e $succ\,n \stackrel{\mathrm{def}}{=} n+1$.

2. A função fac é um caso particular de um *paramorfismo*. No caso geral, dado um tipo indutivo $\mathsf{T} \xleftarrow{in} \mathsf{F\,T}$, o paramorfismo de f relativamente ao functor F, designado por $\{\!|f|\!\}$ é o único morfismo que faz comutar o diagrama seguinte:

$$\begin{tikzcd} C & \mathsf{F}\,(\mathsf{T} \times C) \arrow[l, "f"'] \\ \mathsf{T} \arrow[u, "\{\!|f|\!\}"] & \mathsf{F\,T} \arrow[l, "in"] \arrow[u, "\mathsf{F}\,\langle id, \{\!|f|\!\}\rangle"'] \end{tikzcd}$$

   que traduz a seguinte propriedade universal:

$$h = \{\!|f|\!\} \quad \Leftrightarrow \quad h \cdot in = f \cdot \mathsf{F}\,\langle id, h\rangle$$

   A partir desta propriedade universal deduza a regra de **reflexão-para**:

$$id = \{\!|in \cdot \mathsf{F}\pi_1|\!\}$$

3. Analise a função f que a seguir se especifica, em notação CAMILA:

```
f(l) = if l== <> then true
       else let (h=hd(l), t=tl(l))
            in (if t== <> then true else h >= hd(t)) /\ f(t)
```

   Explique o seu significado por palavras suas e exprima-a sob a forma de um paramorfismo de sequências de naturais ($\mathsf{F}\,X = 1 + I\!N \times X$).

Questão 5 Seja $\mathsf{T} \xleftarrow{in} \mathsf{F\,T}$ um tipo indutivo e f e g duas funções mutuamente recursivas

$$\left\{ \begin{array}{l} f \cdot in = h \cdot \mathsf{F}\,\langle f, g\rangle \\ g \cdot in = k \cdot \mathsf{F}\,\langle f, g\rangle \end{array} \right. \tag{2}$$

paramétricas em h e k, cf os diagramas:

$$\begin{tikzcd} \mathsf{T} \arrow[d, "f"'] & \mathsf{F\,T} \arrow[l, "in"'] \arrow[d, "\mathsf{F}\,\langle f,g\rangle"] \\ A & \mathsf{F}\,(A \times B) \arrow[l, "h"] \end{tikzcd} \qquad \begin{tikzcd} \mathsf{T} \arrow[d, "g"'] & \mathsf{F\,T} \arrow[l, "in"'] \arrow[d, "\mathsf{F}\,\langle f,g\rangle"] \\ B & \mathsf{F}\,(A \times B) \arrow[l, "k"] \end{tikzcd}$$

1. Justifique os passos do seguinte raciocínio que mostra que o 'split' $\langle f, g\rangle$ se pode escrever como o catamorfismo $(\!|\langle h, k\rangle|\!)$ sobre T:

$$
\begin{array}{cl}
 & \langle f, g\rangle \cdot in = \langle h, k\rangle \cdot \mathsf{F}\,\langle f, g\rangle \\
\equiv & \qquad \{ \ldots \text{(justificar)} \ldots \} \\
 & \langle f, g\rangle \cdot in = \langle h \cdot \mathsf{F}\,\langle f, g\rangle, k \cdot \mathsf{F}\,\langle f, g\rangle\rangle \\
\equiv & \qquad \{ \ldots \text{(justificar)} \ldots \} \\
 & \langle f, g\rangle \cdot in = \langle f \cdot in, g \cdot in\rangle \\
\equiv & \qquad \{ \ldots \text{(justificar)} \ldots \} \\
 & \langle f, g\rangle \cdot in = \langle f, g\rangle \cdot in \\
\equiv & \qquad \{ \ldots \text{(justificar)} \ldots \} \\
 & V
\end{array}
$$

cf. o diagrama:

$$
\begin{tikzcd}
\mathsf{T} \arrow[d, "\langle f,g\rangle"'] & \mathsf{F}\,\mathsf{T} \arrow[l, "in"'] \arrow[d, "\mathsf{F}\,\langle f,g\rangle"] \\
A \times B & \mathsf{F}\,(A \times B) \arrow[l, "\langle h,k\rangle"]
\end{tikzcd}
$$

2. Considere o seguinte par de funções mutuamente recursivas, sobre $\mathbb{N}_0$, escritas em notação CAMILA:

```
f(n) = if n == 0 then 0 else g(n .- 1);
g(n) = if n == 0 then 1 else f(n .- 1) .+ g(n .- 1);
```

Aplique a $\langle f, g\rangle$ o resultado da alínea anterior. (Pode observar-se que g não é mais que a função que calcula o n-ésimo número de Fibonacci.)

Anexo–Cálculo de Funções

COMPOSIÇÃO

Natural-id	$f \cdot id = id \cdot f = f$	(3)
Associatividade	$(f \cdot g) \cdot h \quad = \quad f \cdot (g \cdot h)$	(4)

PRODUTO

Universal-$\times$	$k = \langle f, g\rangle \quad \Leftrightarrow \quad \left\{ \begin{array}{l} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{array} \right.$	(5)
Cancelamento-$\times$	$\pi_1 \cdot \langle f, g\rangle = f \quad , \quad \pi_2 \cdot \langle f, g\rangle = g$	(6)
Reflexão-$\times$	$\langle \pi_1, \pi_2\rangle \quad = \quad id_{A\times B}$	(7)
Fusão-$\times$	$\langle g, h\rangle \cdot f = \langle g \cdot f, h \cdot f\rangle$	(8)
Absorção-$\times$	$(i \times j) \cdot \langle g, h\rangle = \langle i \cdot g, j \cdot h\rangle$	(9)
Functor-$\times$	$(g \cdot h) \times (i \cdot j) \quad = \quad (g \times i) \cdot (h \times j)$	(10)
Functor-id-$\times$	$id_A \times id_B \quad = \quad id_{A\times B}$	(11)

COPRODUTO

Universal-$+$	$k = [f, g] \quad \Leftrightarrow \quad \left\{ \begin{array}{l} k \cdot i_1 = f \\ k \cdot i_2 = g \end{array} \right.$	(12)
Cancelamento-$+$	$[g, h] \cdot i_1 = g \quad , \quad [g, h] \cdot i_2 = h$	(13)
Reflexão-$+$	$[i_1, i_2] \quad = \quad id_{A+B}$	(14)
Fusão-$+$	$f \cdot [g, h] = [f \cdot g, f \cdot h]$	(15)
Absorção-$+$	$[g, h] \cdot (i + j) = [g \cdot i, h \cdot j]$	(16)
Functor-$+$	$(g \cdot h) + (i \cdot j) = (g + i) \cdot (h + j)$	(17)
Functor-id-$+$	$id_A + id_B = id_{A+B}$	(18)

EXPONENCIAÇÃO

Universal	$k = \overline{f} \quad \Leftrightarrow \quad f = ap \cdot (k \times id)$	(19)
Cancelamento	$f = ap \cdot (\overline{f} \times id)$	(20)

Reflexão	$\overline{ap} = id_{B^A}$	(21)
Fusão	$\overline{g \cdot (f \times id)} = \overline{g} \cdot f$	(22)
Absorção	$f^A \cdot \overline{g} = \overline{f \cdot g}$	(23)
Functor	$(g \cdot h)^A = g^A \cdot h^A$	(24)
Functor-id	$id^A = id$	(25)

INDUÇÃO

Universal-cata	$k = (\!|\alpha|\!) \quad \Leftrightarrow \quad k \cdot in = \alpha \cdot (\mathsf{F}\,k)$	(26)
Cancelamento-cata	$(\!|\alpha|\!) \cdot in = \alpha \cdot \mathsf{F}\,(\!|\alpha|\!)$	(27)
Reflexão-cata	$(\!|in|\!) = id_{\mathsf{T}}$	(28)
Fusão-cata	$f \cdot (\!|\alpha|\!) = (\!|\beta|\!) \quad \text{if} \quad f \cdot \alpha = \beta \cdot (\mathsf{F}\,f)$	(29)

MISC.

Lei da troca	$[\langle f, g\rangle, \langle h, k\rangle] \quad = \quad \langle [f, h], [g, k]\rangle$	(30)