

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

4.^º Ano da LMCC (7007N2) + LESI (5307P6)
Ano Lectivo de 2001/02

Exame (1.^a chamada) — 16 de Janeiro de 2002
09h30
Salas 2206 e 2207

PROVA SEM CONSULTA (2 horas)

Questão 1 Considere o catamorfismo

$$f = (\lambda in \cdot (id + swap))$$

definido sobre a estrutura

$$\mathbf{LTree}\ A \cong A + \mathbf{LTree}\ A \times \mathbf{LTree}\ A$$

conhecida pelo nome de *árvore com folhas*.

1. Que “faz” a função f ? Desenhe-a sob a forma de um diagrama e converta \mathbf{LTree} e f para notação VDM-SL.
2. Preencha as reticências “...A...” a “...I...” na seguinte prova de idempotência de f :

$$\begin{aligned}
 f \cdot f &= id \\
 &\equiv \{ \dots A \dots \} \\
 f \cdot (\lambda in \cdot (id + swap)) &= id \\
 &\equiv \{ \dots B \dots \} \\
 f \cdot (\lambda in \cdot (id + swap)) &= (\lambda in) \\
 &\Leftarrow \{ \dots C \dots \} \\
 f \cdot (in \cdot (id + swap)) &= in \cdot (id + f \times f) \\
 &\equiv \{ \dots D \dots \} \\
 ((\lambda in \cdot (id + swap)) \cdot in) \cdot (id + swap) &= in \cdot (id + f \times f) \\
 &\equiv \{ \dots E \dots \} \\
 (in \cdot (id + swap)) \cdot (id + f \times f) \cdot (id + swap) &= in \cdot (id + f \times f) \\
 &\equiv \{ \dots F \dots \} \\
 in \cdot (id + swap \cdot (f \times f) \cdot swap) &= in \cdot (id + f \times f) \\
 &\equiv \{ \dots G \dots \} \\
 in \cdot (id + swap \cdot swap \cdot (f \times f)) &= in \cdot (id + f \times f) \\
 &\equiv \{ \dots H \dots \} \\
 in \cdot (id + f \times f) &= in \cdot (id + f \times f) \\
 &\equiv \{ \dots I \dots \} \\
 &\quad V
 \end{aligned}$$

3. Caracterize o *método de indução estrutural* associado ao tipo \mathbf{LTree} , isto é, complete as reticências em:

<p><i>Para provar o facto</i></p> $\forall x \in \text{LTree } A : \phi x$ <p><i>proceder da forma seguinte:</i></p>
<p>(a) Caso de base: <i>para provar</i></p>
<p>(b) Caso induutivo: <i>seja</i></p>
<p style="margin-left: 2em;">i. Hipótese de indução: <i>assumir</i></p>
<p style="margin-left: 2em;">ii. Salto induutivo: <i>provar</i></p>
<p style="margin: 0;">\square</p>

NB: Justifique a sua resposta através do cálculo da relação bem-fundada $\prec_{out\text{-}\text{LTree}}$.

Questão 2 O registo de chamadas não atendidas em telemóveis é vulgarmente organizado de forma a que os últimos números que chamaram estejam imediatamente acessíveis. Essa estrutura corresponde à de um “stack” cuja operação de “push” se redefine da forma seguinte:

```
npush : int * seq of int -> seq of int
-- push int or move it to the front
npush(n, s) == [n] ^ [ s(i) | i in set inds s & n <> s(i) ] ;
```

1. Formule em notação “pointfree” as seguintes propriedades que são preservadas por *npush*:

- Não há números repetidos numa sequência de chamadas não atendidas.
- Sempre que a última chamada se repete várias vezes seguidas, *npush* deixa apenas um único registo seu.

2. Redefina *npush* de forma a preservar apenas as últimas 10 chamadas.

3. Considere agora a estrutura

```
TLM :: registo: seq of int
agenda: map int to seq of char ;
```

que especifica, não só o registo de chamadas não atendidas, mas a agenda que o telemóvel tem indicando, para alguns números de telefone, os nomes respectivos. Especifique em VDM-SL a operação de “display no ecrã” das chamadas não atendidas:

```
display : TLM -> seq of seq of char
```

(como é habitual, só vai poder mostrar um nome se o respetivo número estiver na agenda.)

Questão 3 Recorde o *Problema 6* das sessões laboratoriais desta disciplina, onde se abordou a operação

```
mseCup[@A] : map @A to nat * map @A to nat -> map @A to nat
```

Partindo da definição formal deste operador que foi estudada nas aulas, demonstre ou refute (por contra-exemplo), os factos seguintes:

$$mseCup[@A](m, m) = m \tag{1}$$

$$m = mseCup[@A](m, \{\mid\rightarrow\}) \tag{2}$$

Questão 4 Considere o seguinte modelo VDM-SL que especifica, abstractamente, a estrutura de um sistema de informação baseado no ‘World Wide Web’:

```
WWW = map Ref to URL;           -- (URL=Universal Resource Location)
URL = seq of Unit;
Unit = PlainText | HyperLink;
PlainText = seq of Word;
Word = seq of char;
HyperLink :: link: Ref
          txt: PlainText;      -- "underlined text"
Ref = token;
```

Pretende-se especificar um motor de pesquisa sobre WWW baseado em inversão de texto, isto é, numa função

`invert : WWW -> map Word to set of Ref`

que calcula, para cada palavra, as URLs que a mencionam. Especifique `invert`.

Anexo–Cálculo de Funções

COMPOSIÇÃO

$$f \cdot id = id \cdot f = f \quad (3)$$

$$(f \cdot g) \cdot h = f \cdot (g \cdot h) \quad (4)$$

PRODUTO

$$\text{Universal-}\times \quad k = \langle f, g \rangle \Leftrightarrow \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases} \quad (5)$$

$$\text{Cancelamento-}\times \quad \pi_1 \cdot \langle f, g \rangle = f, \quad \pi_2 \cdot \langle f, g \rangle = g \quad (6)$$

$$\text{Reflexão-}\times \quad \langle \pi_1, \pi_2 \rangle = id_{A \times B} \quad (7)$$

$$\text{Fusão-}\times \quad \langle g, h \rangle \cdot f = \langle g \cdot f, h \cdot f \rangle \quad (8)$$

$$\text{Absorção-}\times \quad (i \times j) \cdot \langle g, h \rangle = \langle i \cdot g, j \cdot h \rangle \quad (9)$$

$$\text{Functor-}\times \quad (g \cdot h) \times (i \cdot j) = (g \times i) \cdot (h \times j) \quad (10)$$

$$\text{Functor-id-}\times \quad id_A \times id_B = id_{A \times B} \quad (11)$$

COPRODUTO

$$\text{Universal-}+ \quad k = [f, g] \Leftrightarrow \begin{cases} k \cdot i_1 = f \\ k \cdot i_2 = g \end{cases} \quad (12)$$

$$\text{Cancelamento-}+ \quad [g, h] \cdot i_1 = g, \quad [g, h] \cdot i_2 = h \quad (13)$$

$$\text{Reflexão-}+ \quad [i_1, i_2] = id_{A+B} \quad (14)$$

$$\text{Fusão-}+ \quad f \cdot [g, h] = [f \cdot g, f \cdot h] \quad (15)$$

$$\text{Absorção-}+ \quad [g, h] \cdot (i + j) = [g \cdot i, h \cdot j] \quad (16)$$

$$\text{Functor-}+ \quad (g \cdot h) + (i \cdot j) = (g + i) \cdot (h + j) \quad (17)$$

$$\text{Functor-id-}+ \quad id_A + id_B = id_{A+B} \quad (18)$$

EXPONENCIAÇÃO

$$\text{Universal} \quad k = \bar{f} \Leftrightarrow f = ap \cdot (k \times id) \quad (19)$$

$$\text{Cancelamento} \quad f = ap \cdot (\bar{f} \times id) \quad (20)$$

$$\text{Reflexão} \quad \overline{ap} = id_{B^A} \quad (21)$$

$$\text{Fusão} \quad \overline{g \cdot (f \times id)} = \bar{g} \cdot f \quad (22)$$

$$\text{Absorção} \quad f^A \cdot \bar{g} = \overline{f \cdot g} \quad (23)$$

$$\text{Functor} \quad (g \cdot h)^A = g^A \cdot h^A \quad (24)$$

$$\text{Functor-id} \quad id^A = id \quad (25)$$

INDUÇÃO

$$\text{Universal-cata} \quad k = (\alpha) \Leftrightarrow k \cdot in = \alpha \cdot F k \quad (26)$$

$$\text{Cancelamento-cata} \quad (\alpha) \cdot in = \alpha \cdot F (\alpha) \quad (27)$$

$$\text{Reflexão-cata} \quad (\in) = id_{\mu F} \quad (28)$$

$$\text{Fusão-cata} \quad f \cdot (\alpha) = (\beta) \quad \text{if } f \cdot \alpha = \beta \cdot F f \quad (29)$$