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

4.º Ano da LMCC (7008N2) + LESI (5308P3)
Ano Lectivo de 2000/01

Exame (época especial) — 3 de Dezembro 2001
17h00
Sala 1214

NB*: Encontra anexa a esta prova a listagem de algumas leis de cálculo estudadas na disciplina.*

PROVA SEM CONSULTA (2h30m)

Questão 1 [4 valores] Os seguintes quatro exemplos de representação da sequência $[a, b, c]$,

$$\begin{aligned} r_1 &= \{2 \mapsto b, 1 \mapsto a, 3 \mapsto c\} \\ r_2 &= \{a, b, c\} \\ r_3 &= \{99 \mapsto c, 7 \mapsto a, 10 \mapsto b\} \\ r_4 &= \{a \mapsto 1, b \mapsto 2, c \mapsto 3\} \end{aligned}$$

sugerem outros tantos refinamentos possíveis para sequências. Defina o modelo de dados sugerido por cada exemplo e discuta a sua adequação. Em cada caso, indique ou um contra-exemplo (*i.é* uma sequência não representável) ou as respectivas funções de abstracção/representação.

Questão 2 [4 valores] Uma das operações conhecidas sobre listas é a da inversão:

```
invl[@A] : seq of @A -> seq of @A
invl(l) == if l = [] then l else invl[@A](tl l) ^ [hd l] ;
```

Use indução sobre `seq of @A` para mostar que $invl$ comuta com a concatenação por ordem inversa, isto é:

$$invl(l \,\hat{}\, r) = (invl\, r) \,\hat{}\, (invl\, l) \tag{1}$$

NB: assuma propriedades básicas sobre listas como, por exemplo,

$$\mathit{head}\ (s \,\hat{}\, r) = \mathit{head}\ s \tag{2}$$

$$\mathit{tail}\ (s \,\hat{}\, r) = (\mathit{tail}\ s) \,\hat{}\, r \tag{3}$$

para $s \neq []$.

Questão 3 [2 valores] Identifique qual das seguintes leis é válida e explique por que razão as outras duas o não são:

$$A \rightharpoonup (B \times C) \cong (A \rightharpoonup B) \times (A \rightharpoonup C) \tag{4}$$

$$(B \rightharpoonup C)^A \cong (A \times B) \rightharpoonup C \tag{5}$$

$$C^\star \times B^\star \leq (B \times C)^\star \tag{6}$$

Questão 4 [4 valores] A compreensão de conjuntos em VDM-SL

```
{ f(a) | a in set s & p(a) }
```

tem como semântica a operação de filtragem

$$filter(f, p) \stackrel{\text{def}}{=} \mathit{dunion} \cdot \mathcal{P}(p \rightarrow sings \cdot f, \underline{\emptyset}) \tag{7}$$

onde $\mathit{dunion} = (\![\underline{\emptyset}, \cup]\!)$ é a operação de união distribuída de conjuntos.

1. Investigue informalmente o resultado das expressões $filter(id, \underline{\text{TRUE}})$ e $filter(\underline{\text{FALSE}}, p)$.

2. Complete as reticências no raciocínio que se segue:

$$
\begin{array}{ll}
& filter(f, \underline{\mathrm{TRUE}}) = dunion \cdot \mathcal{P}(\underline{\mathrm{TRUE}} \rightarrow sings \cdot f, \underline{\emptyset}) \\
\equiv & \quad \{ \cdots\cdots\cdots\cdots \} \\
& filter(f, \underline{\mathrm{TRUE}}) = dunion \cdot \mathcal{P}(sings \cdot f) \\
\equiv & \quad \{ \cdots\cdots\cdots\cdots \} \\
& filter(f, \underline{\mathrm{TRUE}}) = dunion \cdot (\mathcal{P} sings) \cdot \mathcal{P} f \\
\equiv & \quad \{ \cdots\cdots\cdots\cdots \} \\
& filter(f, \underline{\mathrm{TRUE}}) = (\![[\underline{\emptyset}, \cup]]\!) \cdot (\mathcal{P} sings) \cdot \mathcal{P} f \\
\equiv & \quad \{ \cdots\cdots\cdots\cdots \} \\
& filter(f, \underline{\mathrm{TRUE}}) = (\![[\underline{\emptyset}, \cup] \cdot (id + sings \times id)]\!) \cdot \mathcal{P} f \\
\equiv & \quad \{ \cdots\cdots\cdots\cdots \} \\
& filter(f, \underline{\mathrm{TRUE}}) = (\![[\underline{\emptyset}, \cup \cdot (sings \times id)]]\!) \cdot \mathcal{P} f \\
\equiv & \quad \{ \cdots\cdots\cdots\cdots \} \\
& \cdots\cdots\cdots\cdots \\
\equiv & \quad \{ \cdots\cdots\cdots\cdots \} \\
& \cdots\cdots\cdots\cdots \\
\equiv & \quad \{ \cdots\cdots\cdots\cdots \} \\
& filter(f, \underline{\mathrm{TRUE}}) = \mathcal{P} f
\end{array}
$$

Questão 5 [6 valores] Em processos de migração de dados é vulgar recorrer-se a linguagens de "scripting" (*e.g.* Perl) sobre suporte textual, *e.g.* CSV ("comma separated values"). Para cada tabela relacional, listam-se os nomes dos atributos separados por vírgulas e, pela mesma ordem, os registos um a um, tendo o cuidado de não ignorar campos nulos. Uma especificação em VDM-SL para um tal formato poderá ser o que se segue:

```
CSV :: header: seq of String
       records: set of seq of String;
String = seq of char;
```

1. Acrescente a `CSV` o necessário invariante que garante que todos os registos têm o mesmo comprimento, que deverá ser exactamente o número de atributos da tabela original.
2. Partindo da seguinte especificação simplificada de uma base de dados relacional,

   ```
   Database = map FileName to Table;
   Table    = set of Record;
   Record   = map Attribute to Value;
   FileName = String;
   Attribute = String;
   Value = String;
   ```

   especifique em VDM-SL a função que representa uma `Table` em formato `CSV`, não esquecendo a preservação do invariante da alínea anterior.
3. Escreva a função inversa da anterior, isto é, aquela que recupera uma tabela (`Table`) a partir de um ficheiro em formato `CSV` válido.

Anexo

COMPOSIÇÃO

Natural-id	$f \cdot id = id \cdot f = f$	(8)
Associatividade	$(f \cdot g) \cdot h \quad = \quad f \cdot (g \cdot h)$	(9)

PRODUTO

Universal-$\times$	$k = \langle f, g \rangle \quad \Leftrightarrow \quad \left\{ \begin{array}{l} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{array} \right.$	(10)
Cancelamento-$\times$	$\pi_1 \cdot \langle f, g \rangle = f \quad , \quad \pi_2 \cdot \langle f, g \rangle = g$	(11)

Reflexão-$\times$	$\langle \pi_1, \pi_2 \rangle = id_{A\times B}$	(12)
Fusão-$\times$	$\langle g, h\rangle \cdot f = \langle g \cdot f, h \cdot f\rangle$	(13)
Absorção-$\times$	$(i \times j) \cdot \langle g, h\rangle = \langle i \cdot g, j \cdot h\rangle$	(14)
Functor-$\times$	$(g \cdot h) \times (i \cdot j) = (g \times i) \cdot (h \times j)$	(15)
Functor-id-$\times$	$id_A \times id_B = id_{A\times B}$	(16)

COPRODUTO

Universal-$+$	$k = [f, g] \Leftrightarrow \left\{ \begin{array}{l} k \cdot i_1 = f \\ k \cdot i_2 = g \end{array} \right.$	(17)
Cancelamento-$+$	$[g, h] \cdot i_1 = g \quad , \quad [g, h] \cdot i_2 = h$	(18)
Reflexão-$+$	$[i_1, i_2] = id_{A+B}$	(19)
Fusão-$+$	$f \cdot [g, h] = [f \cdot g, f \cdot h]$	(20)
Absorção-$+$	$[g, h] \cdot (i + j) = [g \cdot i, h \cdot j]$	(21)
Functor-$+$	$(g \cdot h) + (i \cdot j) = (g + i) \cdot (h + j)$	(22)
Functor-id-$+$	$id_A + id_B = id_{A+B}$	(23)

EXPONENCIAÇÃO

Universal	$k = \overline{f} \Leftrightarrow f = ap \cdot (k \times id)$	(24)
Cancelamento	$f = ap \cdot (\overline{f} \times id)$	(25)
Reflexão	$\overline{ap} = id_{B^A}$	(26)
Fusão	$\overline{g \cdot (f \times id)} = \overline{g} \cdot f$	(27)
Absorção	$f^A \cdot \overline{g} = \overline{f \cdot g}$	(28)
Functor	$(g \cdot h)^A = g^A \cdot h^A$	(29)
Functor-id	$id^A = id$	(30)

INDUÇÃO

Universal-cata	$k = (\!\vert \beta \vert\!) \Leftrightarrow k \cdot in = \beta \cdot (\mathsf{F}\, k)$	(31)
Cancelamento-cata	$(\!\vert \alpha \vert\!) \cdot in = \alpha \cdot \mathsf{F}\, (\!\vert \alpha \vert\!)$	(32)
Reflexão-cata	$(\!\vert in \vert\!) = id_{\mathsf{T}}$	(33)
Fusão-cata	$f \cdot (\!\vert \alpha \vert\!) = (\!\vert \beta \vert\!) \Leftarrow f \cdot \alpha = \beta \cdot (\mathsf{F}\, f)$	(34)
Absorção-cata	$(\!\vert g \vert\!) \cdot \mathsf{T}\, f = (\!\vert g \cdot \mathsf{B}(f, id) \vert\!)$	(35)

FUNCTORES

Functor-F	$\mathsf{F}(g \cdot h) = (\mathsf{F}\, g) \cdot (\mathsf{F}\, h)$	(36)
Functor-id-F	$\mathsf{F}\, id_A = id_{(\mathsf{F}\, A)}$	(37)
“Teorema grátis” de g	$(\mathsf{G}\, f) \cdot g = g \cdot (\mathsf{F}\, f)$	(38)

MISC.

Lei da troca	$[\langle f, g\rangle, \langle h, k\rangle] = \langle [f, h], [g, k]\rangle$	(39)
Fusão de predicado guardado	$p? \cdot f = (f + f) \cdot (p \cdot f)?$	(40)
1.ª Lei de fusão do condicional	$f \cdot (p \rightarrow g, h) = p \rightarrow f \cdot g, f \cdot h$	(41)
2.ª Lei de fusão do condicional	$(p \rightarrow f, g) \cdot h = (p \cdot h) \rightarrow (f \cdot h), (g \cdot h)$	(42)

‘POWERSETS’

Reflexão-$\mathcal{P}$	$\{\!\vert ins \vert\!\} = id$	(43)
Fusão-$\mathcal{P}$	$f \cdot \{\!\vert g \vert\!\} = \{\!\vert h \vert\!\} \Leftarrow f \cdot g = h \cdot (id + id \times f)$	(44)
Absorção-$\mathcal{P}$	$\{\!\vert g \vert\!\} \cdot (\mathcal{P} f) = \{\!\vert g \cdot (id + f \times id) \vert\!\} \Leftarrow \{\!\vert g \vert\!\} \cdot ins = g \cdot (id + id \times \{\!\vert g \vert\!\})$	(45)

ISOMORFISMOS BÁSICOS

$$A \rightharpoonup B \cong (B + 1)^A \tag{46}$$

$$\mathcal{P} A \cong A \rightharpoonup 1 \tag{47}$$