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

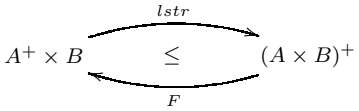
4.º Ano da LMCC (7008N2) + LESI (5308P3)
Ano Lectivo de 2005/06

Exame (1.ª chamada) — 21 de Junho de 2006
09h30
Salas 1316 a 1318

NB: *Esta prova consta de **8** alíneas todas com a mesma cotação.*

PROVA SEM CONSULTA (2 horas)

Questão 1 Atente na lei de refinamento



e defina a representação $lstr$ e a abstracção F em notação VDM-SL.

RESOLUÇÃO:

```
lstr[@A,@B] : seq1 of @A * @B -> seq1 of (@A * @B)
lstr(s,b) == [ mk_(s(i),b) | i in set inds s ];

lstrInv[@A,@B] : seq1 of @A * @B -> seq1 of (@A * @B)
lstrInv(s) == mk_([ s(i).#1 | i in set inds s ], s(1).#2);
```

□

Questão 2 Em muitos problemas de seriação, os indivíduos a seriar (ordenar) dependem de uma *avaliação* sua que é feita de acordo com algum critério pre-estabelecido (eg. aluno/nota de candidatura, professor/prioridade, professor/anos de serviço, etc.). Em todos os casos, essa avaliação pode ser encarada como uma função que observa o indivíduo e lhe atribui um *rank* numérico, tipicamente um número natural (eg. a nota de um candidato ao ensino superior é um natural de 0 a 200).

Assim sendo, faz sentido parametrizar o algoritmo de seriação com uma função f de avaliação que tome valores nos números naturais, ficando assumida a ordem linear destes para seriar as observações. Foi nesse sentido que um colega seu esboçou o seguinte rascunho, em VDM-SL:

```
serial[@A] : (@A -> nat) -> seq of @A -> seq of @A
serial(f)(s) == is not yet specified;
```

Antes, porém, de pensar concluir a sua especifiação, esse seu colega decidiu pedir-lhe ajuda na investigação de propriedades da função que estava a definir. Assim sendo,

1. Calcule-lhe o teorema grátis associado a

$$serial : a^\star \leftarrow a^\star \leftarrow (nat \leftarrow a))$$

2. Verificando-se que num processo de seriação de professores os dados aparecem em dois formatos diferentes,

```
FormatA = ProfId * Rank;
FormatB = Rank * ProfId;
```

onde `ProfId = seq of char` e `Rank = nat`, mostre que a seguinte propriedade escrita em notação VDM-SL,

```
forall s : seq of FormatA &
   let s' = serial[FormatA](classA)(s)
   in  [ swap(s'(i)) | i in set inds s' ] =
            serial[FormatB](classB)([ swap(s(i)) | i in set inds s  ]);
```

é uma consequência do teorema grátis acima calculado, onde

```
swap: FormatA -> FormatB
swap(mk_(a,b)) == mk_(b,a);

classA: FormatA -> Rank
classA(mk_(s,n)) == n;

classB: FormatB -> Rank
classB(mk_(n,b)) == n;
```

3. Uma possível implementação de $serial$ poderá ser

$$serial\ f \quad = \quad sort(\lambda(y,x).(f\ y) \leq (f\ x)) \tag{1}$$

onde $\leq$ é a ordem habitual em números naturais e

$$sort : a^\star \leftarrow a^\star \leftarrow (2 \leftarrow (a \times a))$$

é a função cujo teorema grátis foi calculado nas aulas práticas.

Mostre que a ordem que $sort$ usa para seriar valores (nessa implementação de $serial$) é sempre uma preordem (sugestão: prove que $f^\circ \cdot \leq \cdot f$ é sempre reflexiva e transitiva, qualquer que seja f de tipo adequado).

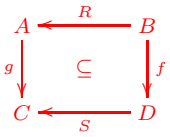
RESOLUÇÃO:

1. Recordando as regras

$$R_{t:=\mathsf{F}(t_1,\ldots,t_n)} \quad = \quad \mathsf{F}(R_{t_1},\ldots,R_{t_n}) \tag{2}$$
$$R_{t:=v} \quad = \quad R_v \tag{3}$$
$$R_{t:=t'\leftarrow t''} \quad = \quad R_{t'} \leftarrow R_{t''} \tag{4}$$

e a definição

$$g(S \leftarrow R)f \quad \equiv \quad g \cdot R \subseteq S \cdot f \qquad \text{cf. diagram} \tag{5}$$



ter-se-á

$$\begin{array}{ll} & serial(R_{(a^\star\leftarrow a^\star)\leftarrow(nat\leftarrow a)})serial \\ \equiv & \quad \{\ (2,3,4)\ ;\ R_{t:=2} = id \text{ (cf. functor constante)}\ \} \\ & serial((R^\star \leftarrow R^\star) \leftarrow (id \leftarrow R))serial \\ \equiv & \quad \{\ (5)\ \} \\ & serial \cdot (id \leftarrow R) \quad \subseteq \quad (R^\star \leftarrow R^\star) \cdot serial \\ \equiv & \quad \{\ \text{shunting}\ \} \\ & (id \leftarrow R) \quad \subseteq \quad serial^\circ \cdot (R^\star \leftarrow R^\star) \cdot serial \\ \equiv & \quad \{\ \text{introdução das variáveis } f \text{ e } g\ \} \\ & f(id \leftarrow R)g \quad \Rightarrow \quad (serial\ f)(R^\star \leftarrow R^\star)(serial\ g) \\ \equiv & \quad \{\ (2,4,5)\ \} \\ & f \cdot R \subseteq g \quad \Rightarrow \quad (serial\ f) \cdot R^\star \subseteq R^\star \cdot (serial\ g) \end{array}$$

2. Se convertermos o predicado dado para notação *pointfree*, obteremos

$$(serial\ classB) \cdot swap^\star = swap^\star \cdot (serial\ classA)$$

que é de facto o consequente da implicação do teorema calculado. O que aconteceu ao antecedente, que deveria ser

$$classB \cdot swap = classA \quad ?$$

Desaparece, pois — como $classB = \pi_1$, $classA = \pi_2$ e $swap = \langle \pi_2, \pi_1 \rangle$ — é imediato que $\pi_1 \cdot \langle \pi_2, \pi_1 \rangle = \pi_2 \equiv$ TRUE, por cancelamento-$\times$.

3. Reflexividade:

$$\begin{array}{ll} & id \subseteq f^\circ \cdot \leq \cdot f \\ \equiv & \quad \{\ \text{shunting}\ \} \\ & f \cdot id \subseteq \leq \cdot f \\ \equiv & \quad \{\ \text{natural-}id\ \} \\ & id \cdot f \subseteq \leq \cdot f \\ \Leftarrow & \quad \{\ (\cdot f)\ \text{é monótona}\ \} \\ & id \subseteq \leq \\ \equiv & \quad \{\ \leq \text{sobre os naturais é reflexiva}\ \} \\ & \text{TRUE} \end{array}$$

Transitividade:

$$\begin{array}{ll} & \text{TRUE} \\ \equiv & \quad \{\ f\ \text{é simples}\ \} \\ & f \cdot f^\circ \subseteq id \\ \Rightarrow & \quad \{\ \text{monotonia de} \leq \cdot \_ \cdot \leq\ ;\ \text{natural-}id\ \} \\ & \leq \cdot f \cdot f^\circ \cdot \leq \subseteq \leq \cdot \leq \\ \Rightarrow & \quad \{\ \leq\ \text{é transitiva}\ \} \\ & \leq \cdot f \cdot f^\circ \cdot \leq \subseteq \cdot \leq \\ \Rightarrow & \quad \{\ \text{monotonia de}\ f^\circ \cdot \_ \cdot f\ \} \\ & f^\circ \cdot \leq \cdot f \cdot f^\circ \cdot \leq \cdot f \subseteq f^\circ \cdot \leq \cdot f \end{array}$$

□

Questão 3 Recordando as leis

$$1 \underset{!}{\overset{!^\circ}{\underset{\leq}{\rightleftarrows}}} A \qquad (A \neq 0) \tag{6}$$

e

$$B \times A + C \times A \underset{distrl}{\overset{}{\underset{}{\rightleftarrows}}} (B + C) \times A \tag{7}$$

deduz-se

$$1 + A \times B \underset{f}{\overset{R = f^\circ}{\underset{\leq}{\rightleftarrows}}} (1 + A) \times B \qquad (B \neq 0) \tag{8}$$

Apresente justificações para os passos do seguinte cálculo da representação R:

$$
\begin{array}{rcl}
R & = & f^{\circ} \\
& = & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& & ((! + id) \cdot distrl)^{\circ} \\
& = & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& & distrl^{\circ} \cdot (!^{\circ} + id^{\circ}) \\
& = & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& & [i_1 \times id \,, i_2 \times id] \cdot (!^{\circ} + id) \\
& = & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& & [(i_1 \times id) \cdot !^{\circ} \,, i_2 \times id] \\
& = & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& & [(i_1 \times id) \cdot \langle id, !^{\circ} \rangle \,, i_2 \times id] \\
& = & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& & [\langle i_1, !^{\circ} \rangle \,, i_2 \times id]
\end{array}
$$

Questão 4 Pretende-se especificar (em VDM-SL) e implementar (numa linguagem imperativa a escolher mais tarde) um pacote estatístico elementar que calcule *modas*, *médias*, *medianas* etc. de dados organizados nos dois formatos habituais: listas de valores e tabelas de frequências (vulg. histogramas).

Nesse sentido, alguém definiu a seguinte função de conversão de um desses formatos no outro:

```
histog[@A] : seq of @A -> map @A to nat1
histog(l) ==
     cases l:
         []     -> {|->},
         others -> let h = [ l(i) | i in set inds l & l(i) =  hd l],
                       r = [ l(i) | i in set inds l & l(i) <> hd l]
                   in { hd l |-> len h } munion histog[@A](r)
     end;
```

1. Demonstre que $histog[token] = histWhile$, onde $histWhile$ é a operação iterativa

```
histWhile : seq of token ==> map token to nat1
histWhile(i) ==
     (dcl m: map token to nat1 := {|->},
          l: seq of token      := i;
          while (l <> []) do
             let h = [ l(i) | i in set inds l & l(i) =  hd l],
                 r = [ l(i) | i in set inds l & l(i) <> hd l]
             in (m := m munion { hd l |-> len h };
                 l := r);
     return m
);
```

2. Se comparar $histWhile$ com

```
histWhile' : seq of token ==> map token to nat1
histWhile'(i) ==
      (dcl m: map token to nat1 := {|->},
           l: seq of token      := i,
           n: nat,
           r: seq of token,
           k: nat;
           while (l <> []) do
```

```
                (n := 0; r := []; k := 0;
                while (k < len l) do
                      ( k := k + 1;
                        if l(k) = hd l
                        then n := n+1
                        else r := r ^ [l(k)]);
                m := m munion { hd l |-> n };
                l := r);
      return m
      );
```

verifica que nesta última operação se foi mais além na factorização iterativa. Quais das leis em anexo (e outras que conheça) lhe parecem ter sido usadas para se derivar $histWhile'$ de $histWhile$? Justifique informalmente.

RESOLUÇÃO: Resposta à segunda alínea:

A alteração tem a ver com o ciclo-while mais interno, que substitui o cálculo de r, de h e de $len\ h$.

Repare-se que h e r eram calculadas na versão anterior por duas compreensões de listas, que mais não são do que duas instâncias do catamorfismo $filter$ que filtra uma lista de acordo com um predicado de selecção (cf. exame da 2.ª chamada).

O que aconteceu foi:

1. o catamorfismo de cálculo do h fundiu com a função len, por fusão-cata — lei (12) em anexo — dando como resultado um novo catamorfismo de listas, com resultado em `nat`.
2. Este novo catamorfismo intercombinou-se com o de cálculo de r pela lei de *banana-split* — (17) do anexo — resultando daí um único acesso à lista de entrada, cujo resultado emparelha um `nat` (n) com uma `seq of token` (r)
3. Este último catamorfismo foi factorizado iterativamente num ciclo-while via lei (19).

□

Questão 5 Recorde a definição-PF da relação de satisfação (algorítmica) de uma especificação S por uma implementação R:

$$S \vdash R \quad \equiv \quad (\mathsf{dom}\ S \subseteq \mathsf{dom}\ R)\ \wedge\ (R \cdot \mathsf{dom}\ S \subseteq S) \tag{9}$$

Mostre que, sempre que a especificação S é simples e a implementação R é uma função f, então (9) reduz-se a

$$S \vdash f \quad \equiv \quad S \subseteq f \tag{10}$$

RESOLUÇÃO: Tem-se:

$$\begin{array}{ll} & S \vdash f \\ \equiv & \quad \{\ (9)\ \} \\ & (\mathsf{dom}\ S \subseteq \mathsf{dom}\ f)\ \wedge\ (f \cdot \mathsf{dom}\ S \subseteq S) \\ \equiv & \quad \{\ \mathsf{dom}\ f = id \text{ é a maior correflexiva ; shunting}\ \} \\ & \mathsf{dom}\ S \subseteq f^{\circ} \cdot S \\ \equiv & \quad \{\ \text{conversos}\ \} \\ & \mathsf{dom}\ S \subseteq S^{\circ} \cdot f \\ \equiv & \quad \{\ (20), \text{ pois } S \text{ é simples}\ \} \\ & S \subseteq f \end{array}$$

□

Anexo–Algumas leis de cálculo que podem ser úteis

Hilomorfismos:

$$[\![R, S]\!]^\circ = [\![S^\circ, R^\circ]\!] \tag{11}$$

$$V \cdot [\![S, R]\!] = [\![T, R]\!] \quad \Leftarrow \quad V \cdot S = T \cdot (\mathsf{F}\, V) \tag{12}$$

$$[\![S, R]\!] \cdot V = [\![S, U]\!] \quad \Leftarrow \quad R \cdot V = \mathsf{F}\, V \cdot U \tag{13}$$

$$[\![T, U]\!] \subseteq [\![R, S]\!] \quad \Leftarrow \quad T \subseteq R \ \wedge \ U \subseteq S \tag{14}$$

$$[\![R, S]\!] \subseteq T \quad \Leftarrow \quad R \cdot \mathsf{F}\, T \cdot S \subseteq T \tag{15}$$

Recursividade múltipla:

$$\left\{ \begin{array}{l} f \cdot in = h \cdot \mathsf{F}\, \langle f, g \rangle \\ g \cdot in = k \cdot \mathsf{F}\, \langle f, g \rangle \end{array} \right. \quad \equiv \quad \langle f, g \rangle = (\!| \langle h, k \rangle |\!) \tag{16}$$

$$\langle (\!| i |\!), (\!| j |\!) \rangle = (\!| (i \times j) \cdot \langle \mathsf{F}\, \pi_1, \mathsf{F}\, \pi_2 \rangle |\!) \tag{17}$$

Factorização iterativa: para θ associativa, tem-se

$$\begin{array}{rcl} \langle \mu\, f \ :: \ p \rightarrow b \ , \ \theta \cdot \langle d, f \cdot e \rangle \rangle & = & p \rightarrow b \ , \ \theta \cdot (id \times b) \cdot w \cdot \langle d, e \rangle \\ & & \mathit{onde} \\ & & w = \underline{\mathit{while}}\, (\neg \cdot p \cdot \pi_2)\ \underline{\mathit{do}}\ \langle \theta \cdot (id \times d), e \cdot \pi_2 \rangle \end{array} \tag{18}$$

Sendo (θ, u) um monoide, tem-se

$$\langle \mu\, f \ :: \ p \rightarrow \underline{u} \ , \ \theta \cdot \langle d, f \cdot e \rangle \rangle \quad = \quad \pi_1 \cdot w \cdot \langle \underline{u}, id \rangle \tag{19}$$

onde w é o mesmo que em (18).

Para S simples, tem-se:

$$S \cdot R \subseteq T \quad \equiv \quad (\mathsf{dom}\, S) \cdot R \quad \subseteq \quad S^\circ \cdot T \tag{20}$$

$$R \cdot S^\circ \subseteq T \quad \equiv \quad R \cdot \mathsf{dom}\, S \quad \subseteq \quad T \cdot S \tag{21}$$