

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

4.º Ano da LMCC (7008N2) + LES1 (5308P3)  
Ano Lectivo de 2001/02

Exame (época de recurso) — 7 de Setembro 2002  
9h30  
Sala 2212

**NB:** Esta prova consta de 7 alíneas todas com a mesma cotação.

PROVA SEM CONSULTA (2 horas)

**Questão 1** Considere a seguinte especificação de uma função em VDM-SL:

```
seq2ff[@A] : seq of @A -> map nat to @A
--
-- seq2ff(l) converts sequence l into a finite mapping keeping track
-- of the original element positions.
--
seq2ff(l) == { i |-> l(i) | i in set inds l };
```

1. Indique quais dos factos

$$\begin{array}{ccc}
 & g & \\
 \text{seq of A} & \xrightarrow{\quad} & \text{map nat to A} \\
 & \leq & \\
 & f & \\
 \end{array} \tag{1}$$

$$\begin{array}{ccc}
 & i & \\
 \text{map nat to A} & \xrightarrow{\quad} & \text{seq of A} \\
 & \leq & \\
 & h & \\
 \end{array} \tag{2}$$

se verifica, e qual a posição  $f$ ,  $g$ ,  $h$  ou  $i$  que `seq2ff` aí ocupa. Justifique a sua resposta.

2. Prove a seguinte propriedade de `seq2ff`:

$$\text{dom seq2ff}(l) = \text{inds } l \tag{3}$$

**Questão 2** Considere a seguinte definição incompleta, em VDM-SL, de uma função de abstracção:

```
f(l) == { x |-> card { i | i in set (inds l) & l(i) = x }
          | x in set elems l
        };
```

Determine a assinatura desta função, especifique (em VDM-SL) uma sua inversa à direita (função de representação) e registre tudo na seguinte  $\leq$ -inequação:

$$\begin{array}{ccc}
 \dots & & \dots \\
 \dots & \xrightarrow{\quad} & \dots \\
 & \leq & \\
 \dots & \xleftarrow{\quad} & \dots \\
 & & \dots
 \end{array}$$

**Questão 3** A lei da representação estrutural do tipo  $\mu F \xrightarrow{\cong} F \mu F$  no tipo  $\mu G \xrightarrow{\cong} G \mu G$  é dada por

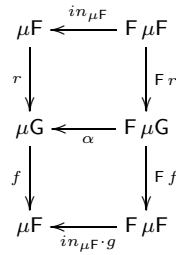
$$\begin{array}{ccc}
 \mu F & \xrightarrow{\cong} & F \mu F \\
 \text{\scriptsize } in_{\mu F} \curvearrowright & & \text{\scriptsize } in_{\mu G} \curvearrowleft \\
 & & \\
 & \xrightarrow{r = (in_{\mu G} \cdot s)_F} & \\
 \mu F & \xrightarrow{\leq} & \mu G \\
 & \text{\scriptsize } f = (in_{\mu F} \cdot g)_G \curvearrowleft & 
 \end{array} \tag{4}$$

desde que

$$\begin{array}{ccc}
 & \xrightarrow{s} & \\
 F X & \xrightarrow{\leq} & G X \\
 & \text{\scriptsize } g \curvearrowleft & 
 \end{array} \tag{5}$$

se verifique.

1. Preencha as reticências nas justificações do seguinte cálculo de  $r$  como inversa à direita de  $f = (in_{\mu F} \cdot g)_G$ , moldada sobre o diagrama:



- $$\begin{array}{l}
 f \cdot r = id \\
 \Leftrightarrow \{ \text{fazendo } r = (in_{\mu F} \cdot g)_G, \text{ passando } \alpha \text{ a ser a incógnita} \} \\
 f \cdot (in_{\mu F} \cdot g)_G = id \\
 \Leftrightarrow \{ \dots\dots\dots \} \\
 f \cdot (in_{\mu F} \cdot g)_G = (in_{\mu F})_F \\
 \Leftarrow \{ \dots\dots\dots \} \\
 f \cdot \alpha = in_{\mu F} \cdot F f \\
 \Leftrightarrow \{ \text{fazendo } \alpha = in_{\mu G} \cdot \alpha' \text{ e passando } \alpha' \text{ a ser a incógnita} \} \\
 f \cdot in_{\mu G} \cdot \alpha' = in_{\mu F} \cdot F f \\
 \Leftrightarrow \{ \text{definição de } f \} \\
 (in_{\mu F} \cdot g)_G \cdot in_{\mu G} \cdot \alpha' = in_{\mu F} \cdot F (in_{\mu F} \cdot g)_G \\
 \Leftrightarrow \{ \dots\dots\dots \} \\
 in_{\mu F} \cdot g \cdot G (in_{\mu F} \cdot g)_G \cdot \alpha' = in_{\mu F} \cdot F (in_{\mu F} \cdot g)_G \\
 \Leftrightarrow \{ \dots\dots\dots \} \\
 in_{\mu F} \cdot F (in_{\mu F} \cdot g)_G \cdot g \cdot \alpha' = in_{\mu F} \cdot F (in_{\mu F} \cdot g)_G \\
 \Leftarrow \{ \dots\dots\dots \} \\
 \alpha' = s
 \end{array}$$

Logo  $\alpha = in_{\mu G} \cdot s$ , e assim se infere  $r = (in_{\mu F} \cdot g)_G = (in_{\mu G} \cdot s)_F$ .

2. Faça  $F X = 1 + A \times X$  (listas),  $G X = 1 + A \times (X \times X)$  (árvores binárias) e  $g = id + id \times \pi_1$ . Calcule  $f$  de acordo com a lei (4) acima e transcreva essa função para notação VDM-SL, fazendo  $A = \text{token}$ ,  $\mu F = \text{seq of token}$  e modelando  $\mu G$  da forma seguinte:

```

BTree = [Node];
Node  :: item: token
      subtrees: STrees;
STrees :: left: BTree
        right: BTree;

```

---

**Questão 4** Considere a seguinte especificação em VDM-SL da função que retira de uma sequência os seus  $n$ -primeiros elementos, se existirem:

```

take[@A] : nat * seq of @A -> seq of @A
take(n,l) == if n=0 or l=[] then []
            else [hd l] ^ take[@A](n-1,tl l);

```

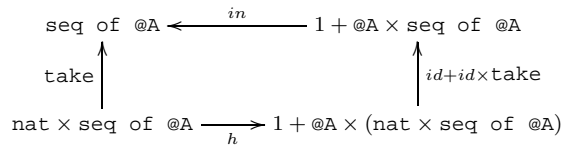
Caracterize o método de indução estrutural

1. Caso de base: *para* .....  
*provar* .....

2. Caso indutivo: *seja* .....  
     (a) Hipótese de indução: *assumir* .....  
     (b) Salto indutivo: *provar* .....

□

aplicável em provas envolvendo esta função, identificando a ordem bem-fundada  $\prec_h$  associada à co-álgebra  $h$  no diagrama que a seguir representa  $\text{take}$  como um anamorfismo:



**Questão 5** Aplique as leis de refinamento de tipos de dados estudados nesta disciplina ao cálculo da representação relacional (tabelas normalizadas) de um modelo que conhece das aulas laboratoriais desta disciplina:

```

PPD :: S: Stock
      P: Pricelist
      E: EquipDb;
Stock = map Unit to Quantity;
Unit = Equip | Comp;
Quantity = nat;
Equip  :: K: token;
Comp  :: K: token;
Pricelist = map Comp to Currency;
Currency = real;
EquipDb = map Equip to map Unit to Quantity;

```

Após o processo de cálculo, descreva o resultado em notação VDM-SL.

---