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

4.º Ano da LMCC (7008N2) + LESI (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

$$\begin{tikzcd} \texttt{seq of A} \arrow[r, bend left, "g"] \arrow[r, phantom, "\leq"] & \texttt{map nat to A} \arrow[l, bend left, "f"] \end{tikzcd} \tag{1}$$

$$\begin{tikzcd} \texttt{map nat to A} \arrow[r, bend left, "i"] \arrow[r, phantom, "\leq"] & \texttt{seq of A} \arrow[l, bend left, "h"] \end{tikzcd} \tag{2}$$

   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`:

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

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 registe tudo na seguinte $\leq$-inequação:

$$\begin{tikzcd} \ldots \arrow[r, bend left, "\ldots"] \arrow[r, phantom, "\leq"] & \ldots \arrow[l, bend left, "\ldots"] \end{tikzcd}$$

Questão 3 A lei da representação estrutural do tipo $\begin{tikzcd} \mu\mathsf{F} \arrow[r, phantom, "\cong"] & \mathsf{F}\,\mu\mathsf{F} \arrow[l, bend left, "in_{\mu\mathsf{F}}"] \end{tikzcd}$ no tipo $\begin{tikzcd} \mu\mathsf{G} \arrow[r, phantom, "\cong"] & \mathsf{G}\,\mu\mathsf{G} \arrow[l, bend left, "in_{\mu\mathsf{G}}"] \end{tikzcd}$ é dada por

$$\begin{tikzcd} \mu\mathsf{F} \arrow[r, bend left, "r = \llparenthesis in_{\mu\mathsf{G}} \cdot s \rrparenthesis_{\mathsf{F}}"] \arrow[r, phantom, "\leq"] & \mu\mathsf{G} \arrow[l, bend left, "f = \llparenthesis in_{\mu\mathsf{F}} \cdot g \rrparenthesis_{\mathsf{G}}"] \end{tikzcd} \tag{4}$$

desde que

$$\begin{tikzcd} \mathsf{F}\,X \arrow[r, bend left, "s"] \arrow[r, phantom, "\leq"] & \mathsf{G}\,X \arrow[l, bend left, "g"] \end{tikzcd} \tag{5}$$

se verifique.

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

$$\begin{tikzcd} \mu\mathsf{F} \arrow[d, "r"'] & \mathsf{F}\,\mu\mathsf{F} \arrow[l, "in_{\mu\mathsf{F}}"'] \arrow[d, "\mathsf{F}\,r"] \\ \mu\mathsf{G} \arrow[d, "f"'] & \mathsf{F}\,\mu\mathsf{G} \arrow[l, "\alpha"] \arrow[d, "\mathsf{F}\,f"] \\ \mu\mathsf{F} & \mathsf{F}\,\mu\mathsf{F} \arrow[l, "in_{\mu\mathsf{F}} \cdot g"] \end{tikzcd}$$

$$\begin{array}{ll}
& f \cdot r = id \\
\Leftrightarrow & \quad \{ \text{ fazendo } r = \llparenthesis \alpha \rrparenthesis_{\mathsf{F}}\text{, passando } \alpha \text{ a ser a incógnita} \} \\
& f \cdot \llparenthesis \alpha \rrparenthesis_{\mathsf{F}} = id \\
\Leftrightarrow & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& f \cdot \llparenthesis \alpha \rrparenthesis_{\mathsf{F}} = \llparenthesis in_{\mu\mathsf{F}} \rrparenthesis_{\mathsf{F}} \\
\Leftarrow & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& f \cdot \alpha = in_{\mu\mathsf{F}} \cdot \mathsf{F}\,f \\
\Leftrightarrow & \quad \{ \text{ fazendo } \alpha = in_{\mu\mathsf{G}} \cdot \alpha' \text{ e passando } \alpha' \text{ a ser a incógnita} \} \\
& f \cdot in_{\mu\mathsf{G}} \cdot \alpha' = in_{\mu\mathsf{F}} \cdot \mathsf{F}\,f \\
\Leftrightarrow & \quad \{ \text{ definição de } f \} \\
& \llparenthesis in_{\mu\mathsf{F}} \cdot g \rrparenthesis_{\mathsf{G}} \cdot in_{\mu\mathsf{G}} \cdot \alpha' = in_{\mu\mathsf{F}} \cdot \mathsf{F}\,\llparenthesis in_{\mu\mathsf{F}} \cdot g \rrparenthesis_{\mathsf{G}} \\
\Leftrightarrow & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& in_{\mu\mathsf{F}} \cdot g \cdot \mathsf{G}\,\llparenthesis in_{\mu\mathsf{F}} \cdot g \rrparenthesis_{\mathsf{G}} \cdot \alpha' = in_{\mu\mathsf{F}} \cdot \mathsf{F}\,\llparenthesis in_{\mu\mathsf{F}} \cdot g \rrparenthesis_{\mathsf{G}} \\
\Leftrightarrow & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& in_{\mu\mathsf{F}} \cdot \mathsf{F}\,\llparenthesis in_{\mu\mathsf{F}} \cdot g \rrparenthesis_{\mathsf{G}} \cdot g \cdot \alpha' = in_{\mu\mathsf{F}} \cdot \mathsf{F}\,\llparenthesis in_{\mu\mathsf{F}} \cdot g \rrparenthesis_{\mathsf{G}} \\
\Leftarrow & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \alpha' = s
\end{array}$$

   Logo $\alpha = in_{\mu\mathsf{G}} \cdot s$, e assim se infere $r = \llparenthesis \alpha \rrparenthesis = \llparenthesis in_{\mu\mathsf{G}} \cdot s \rrparenthesis$.

2. Faça $\mathsf{F}\,X = 1 + A \times X$ (listas), $\mathsf{G}\,X = 1 + A \times (X \times X)$ (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 $A =$ `token`, $\mu\mathsf{F} =$ `seq of token` e modelando $\mu\mathsf{G}$ 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 `take` como um anamorfismo:

$$\begin{tikzcd}
\texttt{seq of @A} & 1 + \texttt{@A} \times \texttt{seq of @A} \arrow[l, "in"'] \\
\texttt{nat} \times \texttt{seq of @A} \arrow[u, "\texttt{take}"] \arrow[r, "h"'] & 1 + \texttt{@A} \times (\texttt{nat} \times \texttt{seq of @A}) \arrow[u, "id + id \times \texttt{take}"']
\end{tikzcd}$$

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 laboratorias 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.
