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

4.º Ano da LMCC (7008N2) + LESI (5308P3)
Ano Lectivo de 2002/03

Exame (época normal, 2ª chamada) — 12 de Julho 2003
09h30
Sala 2302

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

PROVA SEM CONSULTA (2 horas)

Questão 1 Os seguintes quatro exemplos de representação, em notação VDM-SL, da sequência $[a, b, c]$,

$$\begin{aligned} r_1 &= \{2 \mapsto b, 1 \mapsto a, 3 \mapsto c\} \\ r_2 &= \{a \mapsto 1, b \mapsto 2, c \mapsto 3\} \\ r_3 &= \{a, b, c\} \\ r_4 &= [(a, a), (b, b), (c, c)] \end{aligned}$$

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 um contra-exemplo (*i.é* uma sequência não representável) ou as respectivas funções de abstracção / representação.

Questão 2 Definindo, no cálculo relacional,

$$R + S = [i_1 \cdot R \, , i_2 \cdot S] \tag{1}$$

prove as seguintes propriedades da soma de relações:

$$(R + S)^\circ = R^\circ + S^\circ \tag{2}$$

$$[U \, , V] \cdot (R + S) = [U \cdot R \, , V \cdot S] \tag{3}$$

Questão 3 Seja dado o seguinte modelo em VDM-SL para *árvores genealógicas*:

```
AG = Ind | Fam ;          -- AG = arvore genealogica
                          -- Ind = individuo
                          -- Fam = familia
Fam :: c1: Ind            -- cabeça de casal
       c2: Ind            -- o outro conjuge
       desc: set of AG;   -- a descendencia do casal
```

1. Indique quantas tabelas relacionais necessitaria para implementar este tipo de dados em SQL, referindo as leis de refinamento de dados que são aplicadas em cada passo do seu raciocínio.

2. Em qual desses passos é necessária a formulação da relação de pertença estrutural $\in_{\mathsf{F}}$ associada ao functor $\mathsf{F}\,X = Ind + Ind \times Ind \times \mathcal{P}X$ que subjaz a definição de AG? E para que é necessária?
   Calcule a relação $\in_{\mathsf{F}}$ e exprima-a sob a forma de um predicado em notação VDM-SL com variáveis.

Questão 4 Na especificação formal de um sistema de gestão de conhecimento, escrita em VDM-SL, encontra-se o seguinte modelo para expressões sintácticas arbitrárias:

```
Exp    = Var | Term ;
Var    :: variable: Symbol ;
Term   :: operator: Symbol
          arguments: seq of Exp
          inv t == len t.arguments <= 20 ;
Symbol = seq of char
         inv s == len s <= 10 ;
```

Inspeccionando a sua implementação em SQL, verifica-se que a este fragmento de VDM-SL corresponde o seguinte código:

```
CREATE TABLE EXPRESSIONS (
       FatherId  NUMERIC (10) NOT NULL,
       ArgNr     NUMERIC (20) NOT NULL,
       ChildId   NUMERIC (10) NOT NULL
       CONSTRAINT EXPRESSIONS_pk
                  PRIMARY KEY (FatherId,ArgNr)
       );

CREATE TABLE OPERATORS (
       NodeId   NUMERIC (10) NOT NULL,
       Operator CHAR    (10) NOT NULL
       CONSTRAINT OPERATORS_pk
                  PRIMARY KEY(NodeId)
       );

CREATE TABLE VARIABLES (
       NodeId   NUMERIC (10) NOT NULL,
       Variable CHAR    (10) NOT NULL
       CONSTRAINT VARIABLES_pk
                  PRIMARY KEY(NodeId)
       );

ALTER TABLE EXPRESSIONS ADD CONSTRAINT EXPRESSIONS_fk1
      FOREIGN KEY (ChildId) REFERENCES OPERATORS(NodeId);
```

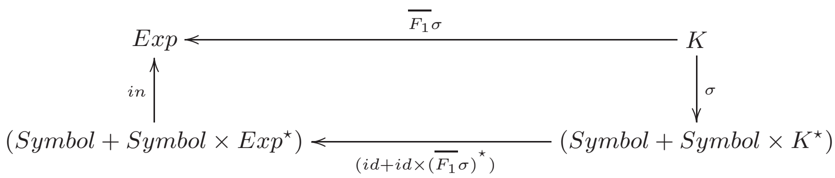
Analisando o manual de implementação, observam-se de início os seguintes dois passos de refinamento,

$$\begin{aligned} Exp &\leq_1 (K \rightharpoonup (Symbol + Symbol \times K^\star)) \times K \\ &\leq_2 ((K \rightharpoonup Symbol) \times (K \rightharpoonup Symbol \times K^\star)) \times K \end{aligned}$$

onde

$$\leq_1 = \begin{cases} R_1 \text{ não vem especificada} \\ \overline{F_1}\sigma = in \cdot (id + id \times (\overline{F_1}\sigma)^\star) \cdot \sigma \end{cases} \tag{4}$$

cf. diagrama



e

$$\leq_2 = \begin{cases} R_2 = uncojoin \times id \\ F_2 = cojoin \times id \end{cases}$$

Continuando a ler o manual de implementação, encontra-se a seguinte dedução da fusão destes passos de refinamento:

$$H = F_1 \cdot F_2$$

isto é,

$$\begin{array}{rcl} H((\sigma_1, \sigma_2), k) & = & F_1(cojoin(\sigma_1, \sigma_2), k) \\ & = & \overline{F_1}(cojoin(\sigma_1, \sigma_2))\ k \\ & = & G\ k \end{array}$$

onde G abrevia $\overline{F_1}(cojoin(\sigma_1, \sigma_2))$. Substituindo-se $\sigma = cojoin(\sigma_1, \sigma_2)$ em (4) e abreviando via G, obtem-se a equação

$$G = in \cdot (id + id \times G^\star) \cdot cojoin(\sigma_1, \sigma_2)$$

1. Complete a seguinte dedução de G como anamorfismo do tipo indutivo Exp:

$$\begin{array}{rl} G = & in \cdot (id + id \times G^\star) \cdot cojoin(\sigma_1, \sigma_2) \\ = & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & in \cdot (id + id \times G^\star) \cdot (i_1 \cdot \sigma_1 \cup i_2 \cdot \sigma_2) \\ = & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & in \cdot (id + id \times G^\star) \cdot [\sigma_1^\circ\ , \sigma_2^\circ]^\circ \\ = & \{ \ \text{lei (2) ;}\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & in \cdot (id^\circ + (id \times G^\star)^\circ)^\circ \cdot [\sigma_1^\circ\ , \sigma_2^\circ]^\circ \\ = & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & in \cdot ([\sigma_1^\circ\ , \sigma_2^\circ] \cdot (id + (id \times G^\star)^\circ))^\circ \\ = & \{ \ \text{lei (3) ;}\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & in \cdot [\sigma_1^\circ\ , \sigma_2^\circ \cdot (id \times G^\star)^\circ]^\circ \\ = & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & in \cdot [\sigma_1^\circ\ , ((id \times G^\star) \cdot \sigma_2)^\circ]^\circ \\ = & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & in \cdot (i_1 \cdot \sigma_1 \cup i_2 \cdot (id \times G^\star) \cdot \sigma_2) \\ = & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & in \cdot (i_1 \cdot \sigma_1 \cdot (\mathsf{dom}\,\sigma_1) \cup i_2 \cdot (id \times G^\star) \cdot \sigma_2 \cdot (\mathsf{dom}\,\sigma_2)) \\ = & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & \text{mk-}Var \cdot \sigma_1 \cdot \mathsf{dom}\,\sigma_1 \cup \text{mk-}Term \cdot (id \times G^\star) \cdot \sigma_2 \cdot (id - \mathsf{dom}\,\sigma_1) \\ = & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & \mathsf{dom}\,\sigma_1 \rightarrow \text{mk-}Var \cdot \sigma_1\ ,\ \text{mk-}Term \cdot (id \times G^\star) \cdot \sigma_2 \end{array}$$

2. Com base no raciocínio anterior, complete a seguinte definição de H após a sua conversão para notação VDM-SL com variáveis:

```
H(mk_(s1,s2),k) ==
      if k in set dom s1 then mk_Var(s1(k))
      else let o = s2(k).#1
               l = s2(k).#2
           in ..................
```

Questão 5 Recorde a lei geral de refinamento algorítmico

$$S \vdash R \quad \equiv \quad R \cdot \mathsf{dom}\, S \subseteq S \ \wedge\ \mathsf{dom}\, S \subseteq \mathsf{dom}\, R \tag{5}$$

que exprime o facto de que R implementa S.

Demonstre ou refute

$$LessThan \vdash Pred2$$

onde

```
LessThan(n: nat) r: nat
post r < n ;
```

e

```
Pred2 : nat -> nat
Pred2(n) == n - 2
pre n > 1 ;
```
