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

4.º Ano da LMCC (7008N2) + LESI (5308P3)
Ano Lectivo de 2003/04

Exame (época especial) — 11 de Setembro de 2004
09h30
Sala 1314

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

PROVA SEM CONSULTA (2 horas)

Questão 1 Recordando as leis

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

e

$$B \times A + C \times A \underset{distrl}{\cong} (B + C) \times A \tag{2}$$

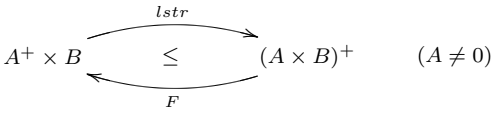
deduz-se

$$1 + A \times B \underset{f}{\overset{R = f^\circ}{\leq}} (1 + A) \times B \tag{3}$$

Complete o seguinte cálculo da representação R:

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

Questão 2 Com respeito à lei



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

Questão 3 Um algoritmo conhecido para compressão de sequências redu-las a sequências de comprimento par registando os extremos dos respectivos intervalos, omitindo todos os valores entre os extremos de uma série de valores consecutivos. Exemplificando, a sequência

$$[4, 5, 6, 10, 20, 19, 18, 23, 24, 25, 26]$$

será reduzida a

$$[4, 6, 10, 10, 20, 18, 23, 26]$$

1. Complete a seguinte especificação em VDM-SL de uma função que deverá realizar tal compressão, onde o tipo de dados `int` é uma simplificação de um qualquer outro tipo paramétrico totalmente ordenado:

```
compress : seq of int -> seq of int
compress(l) == ... g(...) ... ;

g : int * seq of int -> seq of int
g(a,s)  == if s = []
           then [a,a]
           else ... ;
```

2. Investigue o comportamento da sua versão de `compress` quando a lista de entrada tem elementos repetidos. Poderá `compress` ser considerada uma **função** de abstracção? E de representação? Justifique.
3. Especifique a conversa $\texttt{compress}^\circ$ em notação VDM-SL.

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 <= 10 ;
Symbol = seq of char
         inv s == len s <= 20 ;
```

Inspeccionando a sua implementação em SQL, verificamos que a este fragmento de VDM-SL correspondem apenas **duas** tabelas:

```
CREATE TABLE SYMBOLS (
       Symbol   CHAR    (20) NOT NULL,
       NodeId   NUMERIC (10) NOT NULL,
       IfVar    BOOLEAN      NOT NULL
       CONSTRAINT SYMBOLS_pk
                  PRIMARY KEY(NodeId,IfVar)
       );

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

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

Apresente o processo de cálculo que conduziu a esta implementação, indicando as leis de refinamento de dados que foram aplicadas em cada passo do seu raciocínio.

Questão 5 Recorde a lei de refinamento algorítmico

$$S \vdash f \quad \equiv \quad f \cdot \mathit{dom}\, S \subseteq S \tag{4}$$

Comece por completar o cálculo da sua versão com variáveis,

$S \vdash f$

$\equiv$ { definição (4) }

. . . .

$\equiv$ { Galois }

. . . .

$\equiv$ { . . . . }

$y(\mathit{dom}\, S)x \quad \Rightarrow \quad y(f^{\circ} \cdot S)x$

$\equiv$ { . . . . }

$y = x \wedge x \in \mathit{dom}\, S \quad \Rightarrow \quad (f\ y)\ S\ x$

$\equiv$ { . . . . }

$x \in \mathit{dom}\, S \quad \Rightarrow \quad (f\ x)\ S\ x$

aplicando de seguida essa versão à verificação do facto $S \vdash id$ onde S é a especificação que se segue, escrita em VDM-SL:

```
S(n: real) r: real
pre n > 1
post r*r + 2*n*n = 3*n*r;
```

Será id a única implementação functional de S? Justifique informalmente.
