

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

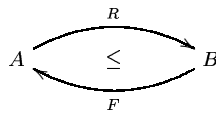
4.º Ano da LMCC (7008N2) + LES1 (5308P3)
Ano Lectivo de 2003/04

Exame (1.ª chamada) — 16 de Junho de 2004
09h30
Salas 2205, 2206

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

PROVA SEM CONSULTA (2 horas)

Questão 1 O cálculo de refinamento de estruturas de dados estudado nesta disciplina baseia-se em inequações da forma



onde F é uma relação de *abstracção* simples e sobrejectiva e R é uma relação de *representação* inteira e injectiva tal que $R \subseteq F^\circ$.

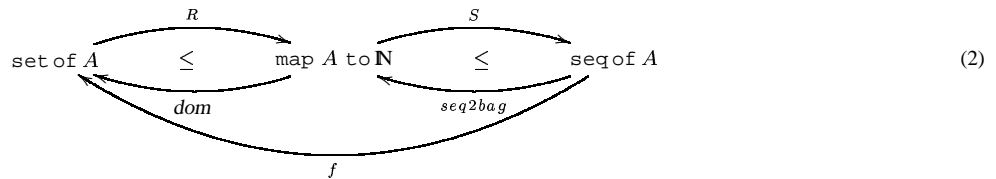
Em muitas situações F e R são funções f e r , respectivamente. Nessas situações, são válidas as seguintes propriedades de cancelamento, à direita e à esquerda, respectivamente:

$$T \cdot f \subseteq S \cdot f \equiv T \subseteq S \equiv r \cdot T \subseteq r \cdot S \quad (1)$$

Apresente justificações detalhadas para os passos da seguinte prova de (1):

$$\begin{aligned}
 & T \cdot f \subseteq S \cdot f \\
 \equiv & \{ \dots \} \\
 & T \cdot f \cdot f^\circ \subseteq S \\
 \equiv & \{ \dots \} \\
 & T \subseteq S \\
 \equiv & \{ \dots \} \\
 & T \subseteq id \cdot S \\
 \equiv & \{ \dots \} \\
 & T \subseteq (r^\circ \cdot r) \cdot S \\
 \equiv & \{ \dots \} \\
 & r \cdot T \subseteq r \cdot S
 \end{aligned}$$

Questão 2 O multiconjunto (“bag”) é uma noção intermédia entre o *conjunto* e a *lista*. É “mais do que um conjunto” na medida em que dá lugar à repetição de elementos, mas é “menos do que uma lista” na medida em que não regista qualquer ordem entre os seus elementos:



1. Proponha duas representações R e S que sejam funções, escrevendo-as em notação VDM-SL.
2. Sabendo que

$$seq2bag = ([[bnil, bcons]]) \quad (3)$$

onde

$$bnil = \{\mapsto\} \quad (4)$$

$$bcons = \oplus \cdot (singb \times id) \quad (5)$$

$$singb a = \{a \mapsto 1\} \quad (6)$$

especifique o operador \oplus (união de multiconjuntos) em notação VDM-SL.

3. Calcule (por fusão-cata) a abstracção f em (2) e identifique-a com uma função da notação VDM-SL que conhece. (**NB:** pode assumir propriedades *válidas* óbvias sobre \oplus sem ter que as provar.)

Questão 3 Na especificação formal de um sistema de gestão de conhecimento, escrita em VDM-SL, encontra-se o seguinte modelo para expressões sintáticas 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, verificamos 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);

```

Concorda com as decisões que foram tomadas nesta codificação? Se a sua resposta é afirmativa, exponha o respectivo processo de cálculo. Se é negativa, indique quais as deficiências encontradas e refaça o respectivo processo de cálculo. Em qualquer dos casos, indique as leis de refinamento de dados que foram aplicadas em cada passo do seu raciocínio, e as respectivas relações de abstracção/representação.

Questão 4 Recordando a lei geral de refinamento algorítmico

$$S \vdash R \equiv R \cdot dom S \subseteq S \wedge dom S \subseteq dom R \quad (7)$$

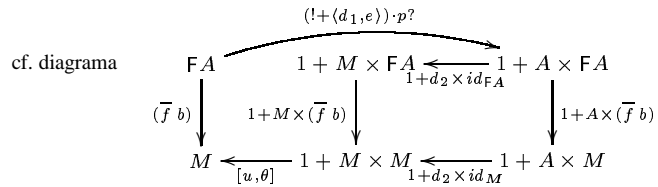
que exprime o facto de R implementar S , mostre que, sempre que S é uma função f , então $f \vdash R \equiv R = f$

Questão 5 Estudou-se nesta disciplina que, *sob certas condições*, toda a função f expressável sob a forma

$$\begin{aligned} \bar{f} &: B \rightarrow FA \rightarrow M \\ (\bar{f} b) &\stackrel{\text{def}}{=} p \rightarrow u, \theta \cdot \langle d_2 \cdot d_1, (\bar{f} b) \cdot e \rangle \end{aligned} \quad (8)$$

onde

$p : FA \rightarrow \text{Bool}$
 $e : FA \rightarrow FA$
 $d_1 : FA \rightarrow A$
 $d_2 : A \rightarrow M$
 $\theta : M \times M \rightarrow M$
 $u : 1 \rightarrow M$



é convertível num ciclo “while”. Mostre que a função f definida no fragmento de VDM-SL que se segue,

```

types

BTree = [Node];
Node :: item: int left: BTree right: BTree;

functions

f : int * BTree -> bool
f(b,t) == cases t:
    nil          -> false,
    mk_Node(x,l,r) -> (b = x) or if (b < x) then f(b,l) else f(b,r)
end;

```

está nessas condições. Justifique adequadamente a sua resposta identificando eventuais lei de cálculo que tenha utilizado e todos os parâmetros em jogo (p , e , θ , etc).

