Métodos Formais de Programação I +
Opção I - Métodos Formais de Programação I

4.º Ano da LMCC (7007N2) + LESI (5307P6)
Ano Lectivo de 2006/07

Exame (época especial) — 15 de Setembro 2007
09h30
Sala 1317

NB: *Esta prova consta de **8** alíneas todas com as mesma cotação.*

PROVA SEM CONSULTA (2 horas)

Questão 1 Seja dada uma especificação *pre/post* em VDM sobre um tipo de dados sujeito ao invariante inv. Converta a inclusão

$$\lceil pre \rceil \cdot \lceil inv \rceil \quad \subseteq \quad \top \cdot \lceil inv \rceil \cdot \lceil post \rceil \tag{1}$$

para notação com variáveis e identifique qual a *obrigação de prova* da metodologia VDM que (1) exprime.

Questão 2 Considere as leis seguintes do cálculo relacional:

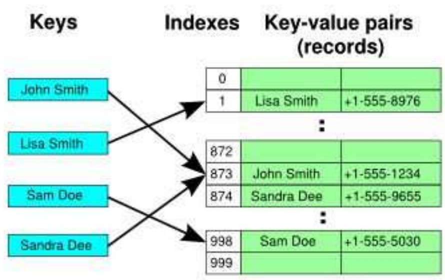
$$\langle R, S \rangle^\circ \cdot \langle X, Y \rangle \quad = \quad (R^\circ \cdot X) \cap (S^\circ \cdot Y) \tag{2}$$

$$[R\ , S] \cdot [T\ , U]^\circ \quad = \quad (R \cdot T^\circ) \cup (S \cdot U^\circ) \tag{3}$$

1. Mostre, a partir de (2), que o *split* de duas relações é inteiro se e só se ambas o forem.
2. Que facto semelhante pode deduzir de (3)? Formule-o e justifique-o formalmente.

Questão 3 Atente no seguinte fragmento de artigo da *Wikipedia* que aborda o conceito de 'open addressing' em gestão de tabelas de 'hashing':

> *"Open addressing hash tables store the records directly within the array. This approach is also called closed hashing. A hash collision is resolved by probing, or searching through alternate locations in the array (the probe sequence) until either the target record is found, or an unused array slot is found, which indicates that there is no such key in the table. Well known probe sequences include: (a)* linear probing *in which the interval between probes is fixed–often at 1; (b)* quadratic probing *in which the interval between probes increases linearly (hence, the indices are described by a quadratic function); (c)* double hashing *in which the interval between probes is fixed for each record but is computed by another hash function."*
> (`http://en.wikipedia.org/wiki/Hash_table`)



Considere agora o seguinte esboço de modelo em VDM para a inserção e procura numa tabela de *hashing* segundo a técnica de *open addressing* descrita nesse fragmento:

```
types

Index   = nat1;
OpenHT :: table: map Index to Record
          max: Index
          inv mk_OpenHT(t,m) == dom t subset {1,...,m};
Record :: key: Key data: Data;
Key    = token;
Data   = token;

functions

hash: Key -> Index
hash(k) == is not yet specified;

Insert : OpenHT * Record -> OpenHT
Insert(h,r) == let j = findix(h,hash(r.key))
               in mk_OpenHT(h.table munion { j |-> r } , h.max)
pre card dom h.table < h.max;

findix: OpenHT * Index -> Index
findix(h,i) == is not yet specified;

lookup : OpenHT * Key -> [ Data ]
lookup(h,k) == is not yet specified;
```

1. Acrescente ao modelo de dados um invariante que garanta que não há chaves (*Key*) repetidas numa tabela.
2. Complete a especificação da função *findix* por forma a garantir o funcionamento em regime de *linear probing*.
3. Especifique a função $lookup$.

Questão 4 A chamada *lei de Dedekind*

$$R \cdot S \cap P \quad \subseteq \quad R \cdot (S \cap R^\circ \cdot P) \tag{4}$$

é útil para raciocinar sobre determinados padrões de notação-PF. Use-a na justificação do facto

$$R \cap \Phi \cdot \top \quad = \quad \Phi \cdot R \tag{5}$$

completando

$$
\begin{array}{cl}
 & R \cap \Phi \cdot \top \quad \subseteq \quad \Phi \cdot R \\
\Leftarrow & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & \Phi \cdot (\top \cap \Phi^\circ \cdot R) \quad \subseteq \quad \Phi \cdot R \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & \Phi \cdot (\Phi \cdot R) \quad \subseteq \quad \Phi \cdot R \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & \Phi \cdot R \quad \subseteq \quad \Phi \cdot R
\end{array}
$$

e provando, de seguida, a inclusão em sentido contrário, isto é, $\Phi \cdot R \subseteq R \cap \Phi \cdot \top$.

Questão 5 O manual "on-line" de VDM-SL fornece a seguinte informação sobre um operador da linguagem:

Operator	Name	Semantics description
`m ** n`	Map iteration	yields the map where `m` is composed with itself `n` times. `n=0` yields the identity map where each element of `dom m` is mapped into itself; `n=1` yields `m` itself. For `n>1`, the range of `m` must be a subset of `dom m`.

Para raciocinarmos sobre este operador precisamos da seguinte semântica formal expressa no cálculo relacional:

$$\lceil \texttt{M ** } 0 \rceil = \delta \lceil \texttt{M} \rceil \tag{6}$$

$$\lceil \texttt{M ** } (n+1) \rceil = \lceil \texttt{M} \rceil \cdot \lceil \texttt{M ** } n \rceil \tag{7}$$

Calcule $\lceil \texttt{M ** } 1 \rceil$ e mostre que $M^\circ \subseteq \top \cdot M$ é a transformada-PF da condição que a semântica do manual VDM-SL exige para o caso $n > 1$.