

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

4.º Ano da LMCC (7007N2) + LES1 (5307P6)  
Ano Lectivo de 2002/03

Exame (2.ª chamada) — 31 de Janeiro de 2003  
14h00  
Salas 2110,2111

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

PROVA SEM CONSULTA (2 horas)

**Questão 1** Sejam dados, em notação VDM-SL, os tipos enumerados

```

People = <Manuel> | <Maria> ;
Company = <Tap> | <Iberia> ;
Flight = <TP123> | <IB987> | <KL555> ;
    
```

sobre os quais se definem duas relações finitas (=conjuntos finitos de pares)

```

R = {
    mk_(<TP123>, <Manuel>),
    mk_(<KL555>, <Manuel>),
    mk_(<IB987>, <Maria>)
};

S = {
    mk_(<TP123>, <Tap>),
    mk_(<TI123>, <Iberia>),
    mk_(<IB987>, <Iberia>),
    mk_(<KL555>, <KLM>)
};
    
```

com o significado seguinte:

$f R p = o$  voo  $f$  transporta o passageiro  $p$

$f S c = f$  é voo da companhia  $c$

1. Comece por calcular, manualmente, a partir de  $R$  e de  $S$ , a relação  $T$  com o significado seguinte:

$p T c = o$  passageiro  $p$  é “fiel” à companhia  $c$ , isto é, apenas voa em voos de  $c$ .

Verificando que essa relação poderia ter sido realizado à custa do operador genérico de “divisão” de relações descrito pelo diagrama

$$\begin{array}{ccc}
 A & \xleftarrow{T=R \setminus S} & C \\
 & \searrow R & \swarrow S \\
 & & B
 \end{array}
 \quad a(R \setminus S)c \equiv (\forall b. bRa \Rightarrow bSc)
 \tag{1}$$

complete a seguinte especificação em VDM-SL desse operador para o caso de relações finitas enumeradas como conjuntos de pares:

```

div[@A,@B,@C]: set of (@B*@A) * set of (@B*@C) -> set of (@A*@C)
div(r,s) ==
    { mk_(p,c) | p in set .....
      c in set .....
      & .....
    };
    
```

confirmando que, de facto,  $T = \text{div}[\text{People}, \text{Flight}, \text{Company}](R, S)$ .

2. A definição “point-free” de  $R \setminus S$  é dada pela propriedade universal

$$X \subseteq R \setminus S \equiv R \cdot X \subseteq S \tag{2}$$

Mostre, através de uma prova indirecta, que a seguinte propriedade se verifica:

$$R \setminus (S \setminus T) = (S \cdot R) \setminus T \tag{3}$$

---

**Questão 2** Estabeleça a diferença entre as funções  $f = \{g\}$  e  $f' = \langle g \rangle$ , para um dado  $g$ . Se fizer  $g = ins$ , qual das funções  $f$  ou  $f'$  é a identidade? Converta essa função (polimórfica) para notação VDM-SL convencional e acompanhe a sua resposta com diagramas explicativos.

---

**Questão 3** Considere a seguinte definição em notação VDM-SL de uma função sua conhecida:

```
fib: nat -> nat
fib(n) == if n < 2 then 1 else fib(n-1)+fib(n-2);
```

Complete as especificações das funções parciais  $R$  e  $S$  seguintes,

```
R : [nat*nat] -> nat
R(x) ==
  cases x:
    nil      -> .....
    mk_(n,m) -> .....
  end
pre cases x:
  nil      -> .....
  mk_(n,m) -> .....
end ;

S : [nat*nat] -> nat
S(x) ==
  cases x:
    nil      -> .....
    mk_(n,m) -> .....
  end
pre cases x:
  nil      -> .....
  mk_(n,m) -> .....
end ;
```

por forma a validar a hilo-factorização

$$fib = \langle R \rangle \cdot \langle S \rangle^\circ$$

**NB:** acompanhe a sua resposta com um diagrama explicativo.

---

**Questão 4** Considere o catamorfismo

$$f = \langle in \cdot (id + swap) \rangle$$

definido sobre a estrutura

$$LTree A \cong A + LTree A \times LTree A$$

conhecida pelo nome de *árvore com folhas*.

1. Que “faz” a função  $f$ ? Desenhe-a sob a forma de um diagrama e converta-a para notação VDM-SL, bem como o tipo  $LTree$ .
2. Preencha as reticências na seguinte prova de idempotência de  $f$ :

$$\begin{aligned} & f \cdot f = id \\ \equiv & \{ \dots \} \\ & f \cdot \langle in \cdot (id + swap) \rangle = id \end{aligned}$$

$$\begin{aligned}
&\equiv \{ \dots \} \\
&\quad f \cdot ((in \cdot (id + swap))) = (in) \\
&\Leftarrow \{ \dots \} \\
&\quad f \cdot (in \cdot (id + swap)) = in \cdot (id + f \times f) \\
&\equiv \{ \dots \} \\
&\quad ((in \cdot (id + swap)) \cdot in) \cdot (id + swap) = in \cdot (id + f \times f) \\
&\equiv \{ \dots \} \\
&\quad (in \cdot (id + swap)) \cdot (id + f \times f) \cdot (id + swap) = in \cdot (id + f \times f) \\
&\equiv \{ \dots \} \\
&\quad in \cdot (id + swap \cdot (f \times f) \cdot swap) = in \cdot (id + f \times f) \\
&\equiv \{ \dots \} \\
&\quad in \cdot (id + swap \cdot swap \cdot (f \times f)) = in \cdot (id + f \times f) \\
&\equiv \{ \dots \} \\
&\quad in \cdot (id + f \times f) = in \cdot (id + f \times f) \\
&\equiv \{ \dots \} \\
&\quad \text{TRUE}
\end{aligned}$$

**Questão 5** Neste exercício aborda-se a modelação formal em VDM-SL de um sistema de reserva de lugares numa rede de transportes (eg. comboio, camionete ou outros). O modelo toma como primitivos os tipos que descrevem estações, paragens ou apeadeiros,

Station = token;

os identificadores do meio de transporte em si

TransId = token;

os números de lugar,

SeatNo = token;

e os códigos de reserva de lugar:

ResId = token;

Uma viagem não é mais do que uma sequência de paragens:

Journey = seq of Station;

Uma reserva é feita para um *segmento* de uma viagem (eg. da segunda à quinta paragem):

```

Segment :: origin      : nat1
          destination  : nat1
          inv s == s.origin < s.destination;

```

Para cada comboio (camionete, etc), regista-se a sua rota (as sucessivas estações onde pára) e o conjunto de lugares disponíveis:

```

TransInfo :: route : Journey
           seats  : set of SeatNo;

```

A cada reserva deverá estar associada a informação seguinte: o lugar reservado, em que comboio (camionete, etc) e qual o segmento afectado. É possível ter um dado lugar reservado da paragem 3 à 5 por um dado passageiro, e reservado por um outro passageiro da paragem 7 à 9, por exemplo.

O sistema de reservas é então modelado por duas funções parciais finitas:

```

System :: trains : map TransId to TransInfo
        res      : map ResId to ResInfo

```

1. Defina as seguintes cláusulas de um invariante sobre System:

- (a) todos os comboios com reservas são comboios conhecidos no sistema;
- (b) não há colisão de reservas (i.e., não há duas reservas distintas do mesmo lugar no mesmo comboio para a mesma parte de uma viagem). **NB:** Há aqui que definir o que se entende por sobreposição de segmentos de viagem, por exemplo definindo uma função auxiliar.

2. Especifique as funções seguintes sobre System:

- (a) função que faz uma reserva de um lugar num dado comboio (camionete, etc);
  - (b) função que sugere lugares disponíveis: dado um comboio (camionete, etc) e um segmento de viagem, informa sobre os lugares disponíveis (i.e., lugares sem quaisquer reservas nesse segmento).
-