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

4.º Ano da LMCC (7007N2) + LESI (5307P6)
Ano Lectivo de 2003/04

Exame (época de recurso) — 18 de Fevereiro 2004
14h00
Sala 2301

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

PROVA SEM CONSULTA (2 horas)

Questão 1 Na compilação das notas de uma disciplina com componente prática, a informação regista-se segundo o modelo de dados que a seguir se esboça, em notação VDM-SL:

```
BdNotas :: inscritos: set of NrAluno
           teoricas: Classif
           praticas: Classif ;

Classif = map NrAluno to Registo;

Registo = map Atributo to Valor;

NrAluno = token;

Atributo = seq of char;

Valor = seq of char | real ;
```

1. Acrescente ao modelo um invariante que garanta que: (a) apenas alunos inscritos podem comparecer a exame e/ou realizar trabalhos práticos; (b) se o atributo `"Nota"` ocorre num dado `Registo`, então ele só toma valores em `real` e nunca em `seq of char`.
2. Especifique a operação de integração das notas teóricas e práticas,

   ```
   integra : (Registo * Registo -> Registo) -> BdNotas -> Classif
   integra(f)(bd) == ... ;
   ```

   de forma paramétrica em `f`, a função que agrega e calcula as classificações finais (NB: a sua especificação deverá garantir que todo o aluno avaliado consta do resultado de `integra`).
3. Use `integra` na especificação da operação

   ```
   intTcomP : BdNotas -> Classif
   intTcomP(bd) == ... integra(f1)(bd) ... ;
   ```

   que deve obedecer aos requisitos seguintes:

   > `f1(t,p)` deverá juntar os dois registos `p` e `t` sem qualquer perda de informação, acrescentando-lhe ainda um atributo adicional que conterá a nota pesada `0.6 * t["Nota"] + 0.4 * p["Nota"]` no caso (e apenas no caso) de tal nota poder ser calculada.

   NB: pode e deve impôr pré-condições onde achar conveniente.

Questão 2 Como sabe, a igualdade no cálculo relacional define-se pela "regra de ping-pong":

$$R = S \quad \equiv \quad R \subseteq S \wedge S \subseteq R \tag{1}$$

Prova-se de seguida que, no caso de R e S serem *relações simples* (funções parciais), a regra (1) se torna menos exigente também:

$$\sigma = \tau \quad \equiv \quad \sigma \subseteq \tau \wedge dom\ \tau \subseteq dom\ \sigma \tag{2}$$

Complete a seguinte prova de (2), em que apenas o passo $\Leftarrow$ é relevante (o passo $\Rightarrow$ é trivial; porquê?):

$\sigma \subseteq \tau \wedge dom\ \tau \subseteq dom\ \sigma$

$\equiv$ { ..... }

$\sigma \subseteq \tau \wedge (\tau \subseteq \tau \cdot dom\ \sigma)$

$\equiv$ { ..... }

$\sigma \subseteq \tau \wedge (\tau \subseteq \tau \cdot (id \cap \sigma^{\circ} \cdot \sigma))$

$\Rightarrow$ { ..... }

$\sigma \subseteq \tau \wedge (\tau \subseteq \tau \cdot (id \cap \tau^{\circ} \cdot \sigma))$

$\equiv$ { regra ..... para τ simples é uma igualdade }

$\sigma \subseteq \tau \wedge (\tau \subseteq \tau \cap \sigma)$

$\equiv$ { ..... }

$\sigma \subseteq \tau \wedge \tau \subseteq \sigma$

$\equiv$ { ..... }

$\sigma = \tau$

Questão 3 Use a propriedade universal do operador de divisão de relações $R \setminus S$ para mostrar que:

- a divisão $R \setminus S$ é reflexiva se e só se $R \subseteq S$;
- R é transitiva se e só se R está contida em $R \setminus R$.

Questão 4 Na modelação formal em VDM-SL de um sistema de reserva de lugares numa rede de transportes (eg. comboio, camionete ou outros) entendeu-se por *viagem* uma sequência de paragens,

```
Journey = seq of Station;
```

e por *reserva* um *segmento* de uma viagem (eg. da segunda à quinta paragem):

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

NB: admita que, num segmento `mk_Segment(i,j)`, o ocupante entra na estação `i` e sai na estação `j` (quer dizer, o lugar já está vago em `j`).
Complete a seguinte especificação da operação de intersecção de dois segmentos de viagem:

```
sint: Segment * Segment -> [Segment]
sint(s,r) == ... if ... then nil else mk_Segment(...) ... ;
```

NB: `nil` deverá modelar o segmento vazio.

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

Operator	Name	Semantics description
`s <: m`	Domain restrict to	Creates the map consisting of the elements in `m` whose key is in `s`. `s` need not be a subset of `dom m`.

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

$$X \mathbin{<:} \sigma \quad = \quad \sigma \cdot [\![X]\!] \tag{3}$$

onde, para $X \subseteq A$, $[\![X]\!] \subseteq id_A$ é a correflexiva $b[\![X]\!]a \quad = \quad b = a \wedge a \in X$. Com base em (3), prove

$$X \mathbin{<:} (Y \mathbin{<:} \sigma) \quad = \quad (X \cap Y) \mathbin{<:} \sigma \tag{4}$$

$$\emptyset \mathbin{<:} \sigma \quad = \quad \{\mapsto\} \tag{5}$$

e formule (sem provar) o equivalente a (3) e (4 , 5) para outro operador de VDM-SL que conhece

Operator	Name	Semantics description
`m <-: s`	Domain restricted by	Creates the map consisting of the elements in `m` whose key is not in `s`. `s` need not be a subset of `dom m`.