

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

4.º Ano da LMCC (7007N2) + LESI (5307P6)
Ano Lectivo de 2005/06

Exame (época especial) — 3 de Outubro de 2006
10h00
Sala 3101

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

PROVA SEM CONSULTA (2 horas)

Questão 1 Complete o cálculo que se segue da lei distributiva da divisão relacional

$$(R \cup R') \setminus S = (R \setminus S) \cap (R' \setminus S) \tag{1}$$

completando as necessárias justificações:

$$\begin{aligned} & X \subseteq (R \cup R') \setminus S \\ \equiv & \{ \dots\dots\dots \} \\ & (R \cup R') \cdot X \subseteq S \\ \equiv & \{ \dots\dots\dots \} \\ & (R \cdot X) \cup (R' \cdot X) \subseteq S \\ \equiv & \{ \dots\dots\dots \} \\ & R \cdot X \subseteq S \wedge R' \cdot X \subseteq S \\ \equiv & \{ \dots\dots\dots \} \\ & X \subseteq R \setminus S \wedge X \subseteq R' \setminus S \\ \equiv & \{ \dots\dots\dots \} \\ & X \subseteq (R \setminus S) \cap (R' \setminus S) \\ \therefore & \{ \dots\dots\dots \} \\ & (R \cup R') \setminus S = (R \setminus S) \cap (R' \setminus S) \end{aligned}$$

Questão 2 Uma fila com prioridades (*priority queue*) é uma estrutura tipo FIFO (=‘First-In-First-Out’) em que elementos com prioridade mais elevada são servidos primeiro. Por exemplo, se numa urgência hospitalar estiverem à espera três ocorrências a , b e c , tais que a chegou primeiro e tem prioridade (=gravidade) 0, b chegou a seguir com prioridade 1, e finalmente c chegou por último mas tem prioridade 4, então c será atendido primeiro, b de seguida e finalmente a .

Considere os seguintes dois modelos formais alternativos (em VDM-SL) para *fila com prioridades*,

$PQueueA = \text{seq of } (X * \text{nat})$

e

`PQueueB = map nat to seq of X;`

onde X — o tipo dos objectos a enfileirar — é sinónimo de `token` e a ordem das sequências representa a ordem de chegada (ie., índices menores correspondem a chegadas mais cedo).

1. Um dos modelos acima carece de um invariante. Qual? Identifique-o e defina-o em VDM-SL.
2. Especifique — sobre o modelo `PQueueA` ou `PQueueB`, à sua escolha — a função que modela a chegada de um novo elemento e a uma fila l , com prioridade p .
3. Especifique — sobre o modelo que não escolheu na alínea anterior — a função que modela a saída do próximo elemento com mais prioridade de uma fila l .

Questão 3 Mostre que a definição relacional do condicional de McCarthy,

$$P \rightarrow R, S = (R \cdot \delta P) \cup (S \cdot \neg(\delta P)) \quad (2)$$

(onde, para Φ correflexiva, $\neg\Phi$ abrevia $id - \Phi$) decorre da propriedade universal:

$$P \rightarrow R, S \subseteq X \equiv R \cdot \delta P \subseteq X \wedge S \cdot \neg(\delta P) \subseteq X \quad (3)$$

Questão 4 Considere o operador de “actualização selectiva” de uma função finita, em notação VDM-SL:

```
selUp[@A,@B]: set of @A * (@B -> @B) * map @A to @B -> map @A to @B
selUp(S,f,M) == M ++ fmap[@A,@B](f)(S <: M);
```

onde `selUp(S, f, M)` designa o ‘mapping’ que se obtém de M quando a função f actualiza todas as imagens no do contradomínio de M cujos objectos estão no conjunto S , e só essas.

1. Complete a definição da função

`fmap[@A,@B]: (@B -> @B) -> map @A to @B -> map @A to @B`

que é usada na definição dada.

2. Com base na transformada-PF de `selUp`,

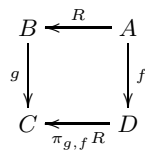
$$[[selUp S f M]] \stackrel{\text{def}}{=} M \dagger (f \cdot M \cdot [S]) \quad (4)$$

mostre que, se S for o conjunto vazio, então, qualquer que seja f , o resultado do operador `selUp S f M` é o próprio M .

NB: recorde que o operador relacional de sobreposição \dagger é um caso particular do condicional de McCarthy,

$$M \dagger N = N \rightarrow N, M \quad (5)$$

Questão 5 Dada uma relação binária $B \xleftarrow{R} A$ e duas funções f, g tal como no diagrama que se segue,



diz-se que R satisfaz a *dependência funcional* $f \rightarrow g$ se e só se

$$\ker (f \cdot R^{\circ}) \subseteq \ker g \quad (6)$$

se verifica.

Se R a relação coreflexiva que representa relacionalmente um dado conjunto de pares T , i.é $R = \llbracket T \rrbracket$, mostre, introduzindo variáveis em (6) e fazendo os cálculos convenientes, que a dependência $\pi_1 \xrightarrow{\llbracket T \rrbracket} \pi_2$ se expande no seguinte predicado, escrito em VDM-SL:

```
forall t,t' in set T & t.#1 = t'.#1 => t.#2 = t'.#2
```
