

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

4.º Ano da LMCC (7008N2) + LESI (5308P3)
Ano Lectivo de 2005/06

Exame (época de recurso) — 22 de Julho de 2006
09h30
Salas 2210, 2211

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

PROVA SEM CONSULTA (2 horas)

Questão 1 Recorde uma lei de refinamento de dados que estudou nesta disciplina, em duas versões: a *pointfree*

$$\begin{array}{ccc}
 & \xrightarrow{\langle id \rightarrow \pi_1, id \rightarrow \pi_2 \rangle} & \\
 A \rightarrow B \times C & \leq & (A \rightarrow B) \times (A \rightarrow C) \\
 & \xleftarrow{\langle \rightarrow, \rightarrow \rangle} &
 \end{array} \tag{1}$$

e a outra *pointwise*, em sintaxe VDM-SL:

$$\begin{array}{ccc}
 & \xrightarrow{unjoin} & \\
 \text{map } A \text{ to } B * C & \leq & (\text{map } A \text{ to } B) \times (\text{map } A \text{ to } C) \\
 & \xleftarrow{join} &
 \end{array} \tag{2}$$

onde

$$\text{join}(M, N) == \{ a \mid \rightarrow \text{mk}_-(M(a), N(a)) \mid a \text{ in set dom } M \text{ inter dom } N \}; \tag{3}$$

Estará a definição (3) de acordo com (1)? É o que se pretende que prove a seguir, com base na correspondência entre *mappings* VDM-SL e relações simples que conhece:

Dado $M: \text{map } @A \text{ to } @B$ em VDM-SL, a correspondente relação simples $B \xleftarrow{[[M]]} A$ é tal que

$$M = \{ a \mapsto b \mid b[[M]]a \} \tag{4}$$

onde

$$b[[M]]a \equiv a \in \text{dom } M \wedge b = M(a) \tag{5}$$

1. Complete então o cálculo que se segue:

$$\begin{aligned}
 & \text{joint}(M, N) = \{ a \mapsto x \mid x \langle [[M]], [[N]] \rangle a \} \\
 \equiv & \quad \{ \dots\dots\dots \} \\
 & \text{joint}(M, N) = \{ a \mapsto x \mid x(\pi_1^\circ \cdot [[M]] \cap \pi_2^\circ \cdot [[N]])a \} \\
 \equiv & \quad \{ \dots\dots\dots \}
 \end{aligned}$$

$$\begin{aligned}
& \text{joint}(M, N) = \{a \mapsto x \mid x(\pi_1^\circ \cdot \llbracket M \rrbracket) a \wedge x(\pi_2^\circ \cdot \llbracket N \rrbracket) a\} \\
\equiv & \quad \{ \dots\dots\dots \} \\
& \text{joint}(M, N) = \{a \mapsto x \mid (\pi_1 x) \llbracket M \rrbracket a \wedge (\pi_2 x) \llbracket N \rrbracket a\} \\
\equiv & \quad \{ \dots\dots\dots \} \\
& \text{joint}(M, N) = \{a \mapsto (b, c) \mid b \llbracket M \rrbracket a \wedge c \llbracket N \rrbracket a\} \\
\equiv & \quad \{ \dots\dots\dots \} \\
& \text{joint}(M, N) = \{a \mapsto (b, c) \mid a \in \text{dom } M \wedge b = M(a) \wedge a \in \text{dom } N \wedge c = N(a)\} \\
\equiv & \quad \{ \dots\dots\dots \} \\
& \text{joint}(M, N) = \{a \mapsto (b, c) \mid a \in (\text{dom } M \cap \text{dom } N) \wedge b = M(a) \wedge c = N(a)\} \\
\equiv & \quad \{ \dots\dots\dots \} \\
& \text{join}(M, N) == \{ a \mid \rightarrow \text{mk}_\bullet(M(a), N(a)) \mid a \text{ in set dom } M \text{ inter dom } N \};
\end{aligned}$$

2. Os dois quadros que se seguem representam pautas que podem ser vistas como relações simples entre alunos e classificações, isto é, *mappings* VDM-SL:

ALUNO	DISCIPLINA A
Manuela	12 (doze)
Paula	10 (dez)
Rui	16 (dezassexes)
Pedro	12 (doze)

ALUNO	DISCIPLINA B
Manuela	11 (onze)
Rodrigo	11 (onze)
Pedro	17 (dezasete)
Teresa	13 (doze)

(6)

Poderão eles ter sido o resultado de uma representação de acordo com a lei (1)? Justifique.

3. Especifique a função

$$(A \rightarrow B) \times (A \rightarrow C) \xrightarrow{\text{ojoin}} A \rightarrow ((B + 1) \times (C + 1)) \quad (7)$$

(*ojoin* = ‘outer join’) que, do mesmo par (6) deve poder calcular a informação do quadro seguinte, indicando quem fez só uma disciplina e qual a nota respectiva:

ALUNO	DISCIPLINA A	DISCIPLINA B
Teresa	-	13 (treze)
Rui	16 (dezassexes)	-
Rodrigo	-	11 (onze)
Paula	10 (dez)	-

NB: conforme desejar, pode apresentar a sua especificação em notação *pointfree* ou em VDM-SL, completando para isso o esboço

```

ojoin[@A,@B,@C] : (map @A to @B) * (map @A to @C) -> map @A to ([@B]*[@C])
ojoin(M,N) == is not yet specified;

```

e assumindo — como simplificação — que `nil` não é nunca habitante nem de `@B` nem de `@C`.

Questão 2 No exame de 2.^a chamada desta disciplina foi abordada a especificação formal do conceito de *cache* de memória. O recurso a *caches* pode ser visto como uma estratégia de refinamento em que a uma memória principal, modelada em notação VDM-SL por

```
Mem = map Index to Data ;
```

se adiciona uma cache

```

CachedMem :: cache: Mem
           main : Mem;

```

de acordo com a inequação de refinamento

$$\begin{array}{ccc}
 & \xrightarrow{Rep} & \\
 Mem & \leq & CachedMem \\
 & \xleftarrow{absf} &
 \end{array} \tag{8}$$

onde a abstracção é uma função,

```

absf : CachedMem -> Mem
absf(cm) == cm.main ++ cm.cache;

```

e a representação é uma relação:

```

Rep(a:Mem) r: CachedMem
post r.main = a and subM(r.cache, a);

```

para

```

subM: Mem * Mem -> bool
subM(M,N) == dom M subset dom N and forall i in set dom M & M(i) = N(i);

```

1. No referido exame foi feito o cálculo de uma condição essencial a (8):

$$absf \cdot Rep \subseteq id \tag{9}$$

Mostre, através de um raciocínio simples, que (9) é suficiente como garantia de que *Rep* é uma relação *injectiva* (mesmo se desconhecermos a respectiva pós-condição).

2. Na prática, as *caches* têm tamanho fixo, o que implica a aplicação de uma *política de substituição* quando a cache está cheia. No sentido de especificar uma dessas políticas — a *LRU* (remove “least recently used”) — reformulou-se o modelo acima da forma seguinte,

```

CachedMem' :: cache: Cache
            main : Mem;

Cache = seq of (Index*Data)
inv s == len s = card cidxs(s) and len s <= max;

```

onde *max* é o número máximo de ítems que a cache pode guardar e *cidxs* é a função auxiliar

```

cidxs : Cache -> set of Index
cidxs(s) == elems [ s(i).#1 | i in set inds s ];

```

Complete a seguinte definição da operação de *cash miss* segundo a política LRU: *o mais recente dado que se leva para a cache deverá ficar imediatamente acessível, isto é, “subir” para o topo (cabeça) da cache:*

```

CmissLRU : Index * CachedMem' -> CachedMem'
CmissLRU(i, mk_CachedMem'(l, M)) ==
    mk_CachedMem'(UpdateLRU(i, M(i), l), M)
pre i in set dom M \ cidxs(l);

UpdateLRU : Index * Data * Cache -> Cache
UpdateLRU(i, d, l) == .....
pre ..... ;

```

3. Defina em notação VDM-SL a função de abstracção *absf'* que adapta *absf* ao novo modelo *CachedMem'*.

Questão 3 Calcule o teorema grátis da função $1 \xleftarrow{!} A$ e mostre esse teorema é equivalente a

$$\langle \forall R :: R \subseteq \mathbb{T} \rangle \quad (10)$$

Questão 4 De acordo com o modelo de gestão de uma dada empresa de informática, os programadores e outros recursos humanos (*HResource*) apresentam, todas as semanas, um ‘time card’ que indica o número de horas que dedicaram a cada tarefa/projecto em curso. Esta informação é depois cruzada com a estimativa inicial do esforço para cada tarefa (medido em “homens*hora”), o que vem a permitir monitorar o andamento dos trabalhos e os desvios entre o real e o previsto.

Apresenta-se a seguir um fragmento da especificação formal do modelo de dados da aplicação que regista os ‘time card’s, escrita em sintaxe VDM-SL:

```
types

Db = map HRid to HResource;           -- HRid identifies human resources
HResource :: profile: Txt
                tcards: map Week to Effort;   -- total effort per week
Effort = map Task to Hours;           -- hours spent per task
Task :: project: PId
        subtask: TId;
Hours = nat1;
Txt    = seq of char;
PId    = seq of char;
TId    = seq of char;
Week   = nat1;                         -- there are 52 weeks in one year
```

Com recurso às leis de refinamento de dados que foram estudadas nesta disciplina, alguém obteve o seguinte refinamento de Db, constituído por 3 estruturas relacionais normalizadas:

```
DbR :: profiles: map HRid to Txt
        weeks: set of WeekInfo
        hours: map TcardsInfo to Hours;
WeekInfo :: hrid: HRid
            week: Week;
TcardsInfo :: hrid: HRid
            week: Week
            pid: PId
            tid: TId;
```

Justifique este refinamento apresentando o respectivo processo de cálculo, e indicando as leis que foram aplicadas em cada passo ou o respectivo par de funções de *abstracção/representação*.
