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

4.º Ano da LMCC (7008N2) + LESI (5308P3)
Ano Lectivo de 2006/07

Exame (2.ª chamada da época normal) — 4 de Julho 2007
09H30
Salas 1314, 2201

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

PROVA SEM CONSULTA (2 horas)

Questão 1 Demonstrar a seguinte propriedade da relação $\vdash$:

$$(\mathsf{ker}\, g) \vdash f \quad \equiv \quad g \cdot f = g \tag{1}$$

Questão 2 Em *Métodos de Programação I* foi estudado o algoritmo que calcula o quadrado de n somando os n primeiros ímpares, isto é, o hilomorfismo

$$\begin{array}{ccc} \mathbb{N}_0 & \xrightarrow{(id+\langle\omega,id\rangle)\cdot in^{\circ}} & 1 + \mathbb{N}_0 \times \mathbb{N}_0 \\ {\scriptstyle sq}\downarrow & & \downarrow{\scriptstyle id+id\times sq} \\ \mathbb{N}_0 & \xleftarrow[{[\underline{0}\,,(+)]}]{} & 1 + \mathbb{N}_0 \times \mathbb{N}_0 \end{array} \qquad \text{(onde } \omega\ n = 2n+1\text{)} \tag{2}$$

que, em VDM-SL, se escreverá:

```
sq: nat -> nat
sq(n) == cases n:
            0 -> n,
            others -> 2*n-1+sq(n-1)
         end;
```

Contudo, onde está a prova de que o algoritmo calcula *de facto* o qudrado do seu argumento?

1. Partindo da especificação da operação *quadrado de um número natural*

   $$Sq(i : \mathbb{N}_0)\ r : \mathbb{N}_0$$
   $$\textbf{post}\ \ r = i^2$$

   complete o raciocínio seguinte que mostra que, de facto, sq satisfaz a especificação Sq:

   $$\begin{array}{ll} & Sq \vdash sq \\ \equiv & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & (\_)^2 \vdash [\![\, [\underline{0}\,,(+)], (id + \langle\omega, id\rangle)\cdot in^{\circ} \,]\!] \\ \equiv & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \end{array}$$

$$\llbracket [\underline{0}\,,(+)],(id+\langle\omega,id\rangle)\cdot in^\circ\,\rrbracket\subseteq(\_)^2$$

$\Leftarrow$ { .... }

$$[\underline{0}\,,(+)]\cdot(id+id\times(\_)^2)\cdot(id+\langle\omega,id\rangle)\cdot in^\circ\subseteq(\_)^2$$

$\equiv$ { .... }

$$[\underline{0}\,,(+)\cdot\langle\omega,(\_)^2\rangle]\subseteq(\_)^2\cdot in$$

$\equiv$ { .... }

$$[\underline{0}\,,(+)\cdot\langle\omega,(\_)^2\rangle]=(\_)^2\cdot[\underline{0}\,,(1+)]$$

$\equiv$ { .... }

$$(+)\cdot\langle\omega,(\_)^2\rangle=(\_)^2\cdot(1+)$$

$\equiv$ { .... }

$$2x+1+x^2=(x+1)^2$$

$\equiv$ { .... }

TRUE

2. Mostre que Sq pode ainda ser satisfeita por um algoritmo não recursivo, isto é, converta sq num ciclo-*while* e escreva essa nova versão em notação VDM.

Questão 3 Sempre que numa inequação de refinamento o lado abstracto é uma soma $A+B$ de tipos de dados é natural escrever-se a relação de representação como um either $[R\,,S]$. Torna-se então conveniente investigar em que condições é que $[R\,,S]$ é injectiva e inteira. Complete o raciocínio que se segue e que aborda a primeira dessas questões:

$[R\,,S]$ é injectiva

$\equiv$ { .... }

$$[R\,,S]^\circ\cdot[R\,,S]\ \subseteq\ id$$

$\equiv$ { .... }

$$[R\,,S]^\circ\cdot R\cdot i_1^\circ\cup[R\,,S]^\circ\cdot S\cdot i_2^\circ\ \subseteq\ id$$

$\equiv$ { .... }

$$(i_1\cdot R^\circ\cup i_2\cdot S^\circ)\cdot R\cdot i_1^\circ\cup(i_1\cdot R^\circ\cup i_2\cdot S^\circ)\cdot S\cdot i_2^\circ\ \subseteq\ id$$

$\equiv$ { .... }

$$(i_1\cdot R^\circ\cdot R\cdot i_1^\circ\subseteq id)\ \wedge\ (i_2\cdot S^\circ\cdot R\cdot i_1^\circ\subseteq id)\ \wedge\ (i_1\cdot R^\circ\cdot S\cdot i_2^\circ\subseteq id)\ \wedge\ (i_2\cdot S^\circ\cdot S\cdot i_2^\circ\subseteq id)$$

$\equiv$ { .... }

$$(R^\circ\cdot R\subseteq id)\ \wedge\ (i_2\cdot S^\circ\cdot R\cdot i_1^\circ\subseteq id)\ \wedge\ (i_1\cdot R^\circ\cdot S\cdot i_2^\circ\subseteq id)\ \wedge\ (S^\circ\cdot S\subseteq id)$$

$\equiv$ { .... }

R é injectiva $\wedge$ S é injectiva $\wedge$ $(i_2\cdot S^\circ\cdot R\cdot i_1^\circ\subseteq id)$

$\equiv$ { .... }

R é injectiva $\wedge$ S é injectiva $\wedge$ $(S^\circ\cdot R\subseteq\bot)$ (3)

Questão 4 Considere os tipos de estruturas de dados *"apontador para `struct`"* $(A \times B + 1)$ and *"apontador dentro de `struct`"* $((A+1) \times B)$. A questão é: qual destes padrões representa o outro? Verifique se $R \stackrel{\mathrm{def}}{=} [i_1 \times id\ , \langle i_2, !^\circ \rangle]$ em

$$A \times B + 1 \;\underset{f}{\overset{R}{\rightleftarrows}}\; (A+1) \times B \qquad (\leq) \tag{4}$$

é injectiva.
Sugestão: aproveite o resultado (3) da questão anterior.

Questão 5 Derive a lei de cancelamento-cata

$$(\!| R |\!) \cdot in \quad = \quad R \cdot \mathsf{F}\,(\!| R |\!) \tag{5}$$

a partir da lei de cancelamento-hilo (15). **Seguidamente**, complete a demonstração de que catamorfismos preservam injectivividade dos seus genes, isto é

$$R \text{ é injectiva } \Rightarrow (\!| R |\!) \text{ é injectiva}$$

O cálculo a completar é:

$$
\begin{array}{ll}
 & (\!| R |\!)^\circ \cdot (\!| R |\!) \subseteq id \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & (\!| R |\!)^\circ \cdot [\![\, R, in^\circ \,]\!] \subseteq [\![\, in, in^\circ \,]\!] \\
\Leftarrow & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & (\!| R |\!)^\circ \cdot R \subseteq in \cdot \mathsf{F}(\!| R |\!)^\circ \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & R^\circ \cdot (\!| R |\!) \subseteq \mathsf{F}(\!| R |\!) \cdot in^\circ \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & R^\circ \cdot (\!| R |\!) \cdot in \subseteq \mathsf{F}(\!| R |\!) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & R^\circ \cdot R \cdot \mathsf{F}(\!| R |\!) \subseteq \mathsf{F}(\!| R |\!) \\
\Leftarrow & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & R^\circ \cdot R \subseteq id
\end{array}
$$

Questão 6 Considere a aplicação da lei

$$\mu\mathsf{G} \;\underset{Unf}{\overset{R}{\rightleftarrows}}\; \underbrace{(K \rightharpoonup \mathsf{G}\,K)}_{\text{“heap”}} \times K \qquad (\leq) \tag{6}$$

ao caso $\mu\mathsf{G} = LTree$, onde $LTree$ se especifica em VDM como se segue:

```
LTree =  Leaf | Split ;
Leaf   :: value: int ;
Split  :: left:  LTree
          right: LTree ;
```

isto é, $\mathsf{G}\,X = \mathbb{Z} + X \times X$. É dada também a seguinte especificação do tipo de dados correspondente ao lado direito de (6):

```
Heap :: mem: map nat to (int | (nat * nat))
        pointer: nat;
```

1. Completar especificação em VDM da operação que re-aloca as células de um par (memória, apontador) de acordo com uma função f injectiva:

```
heapf: (nat -> nat) -> Heap -> Heap
heapf(f)(mk_Heap(m,p)) ==
      let m' = { ..... |-> cases m(k) :
                             mk_(k1,k2) -> ..... ,
                             others     -> .....
                           end
                 | k in set dom m}
      in mk_Heap( .... , .... );
```

2. Analise a seguinte representação funcional para a lei (6) que usa a função definida na alínea anterior:

```
rep: LTree -> Heap
rep(t) ==
   cases t :
      mk_Leaf(i) -> mk_Heap({1 |-> i} , 1) ,
      mk_Split(t1,t2) ->
              let h1 = rep(t1),
                  h2 = rep(t2),
                  h1' = heapf(lambda x : nat & 2*x)(h1),
                  h2' = heapf(lambda x : nat & 2*x+1)(h2)
              in mk_Heap({1 |-> mk_(h1'.pointer,h2'.pointer)}
                         munion h1'.mem munion h2'.mem, 1)
         end;
```

Porque é que o uso de `munion` é seguro? (Justifique informalmente.) Suponha ainda que pede a um interpretador de VDM para calcular a expressão `rep(t)` onde o valor de teste `t` é

```
t: LTree = mk_Split(
               mk_Split(mk_Leaf(-1),mk_Leaf(-2)),
               mk_Leaf(-3));
```

Reproduza o valor do tipo `Heap` que o interpretador deverá devolver nesse caso de teste.

Anexo–Algumas leis de cálculo que podem ser úteis

Refinamento algorítmico:

$$S \vdash R \quad \equiv \quad (\delta\, S \subseteq \delta\, R) \;\wedge\; (R \cdot \delta\, S \subseteq S) \tag{7}$$

Condicional de McCarthy (versão relacional)

$$R \rightarrow S\;,\; T \quad \stackrel{\text{def}}{=} \quad (S \cdot \delta\, R) \cup (T \cdot \neg \delta\, R) \tag{8}$$

Refinamento de dados:

$$\begin{array}{rcll} A \rightharpoonup 1 & \cong & \mathcal{P} A & (9) \\ A \rightarrow B \times C & \leq & (A \rightarrow B) \times (A \rightarrow C) & (10) \\ A \rightarrow B & \leq & A \rightharpoonup \mathcal{P} B & (11) \\ (B \times C) \rightharpoonup A & \leq & B \rightharpoonup (C \rightharpoonup A) & (12) \\ A \rightharpoonup (D \times (B \rightharpoonup C)) & \leq & (A \rightharpoonup D) \times ((A \times B) \rightharpoonup C) & (13) \\ \mu \mathsf{F} & \leq & (K \rightharpoonup \mathsf{F}\, K) \times K & (14) \end{array}$$

Hilomorfismos:

$$[\!\![\, R, S \,]\!\!] \quad = \quad R \cdot \mathsf{F}\, [\!\![\, R, S \,]\!\!] \cdot S \tag{15}$$

$$[\!\![\, R, S \,]\!\!] \subseteq T \quad \Leftarrow \quad R \cdot \mathsf{F}\, T \cdot S \subseteq T \tag{16}$$

$$V \cdot [\!\![\, S, H \,]\!\!] \subseteq [\!\![\, T, H \,]\!\!] \quad \Leftarrow \quad V \cdot S \subseteq T \cdot (\mathsf{G}\, V) \tag{17}$$

$$[\!\![\, R, S \,]\!\!]^\circ = [\!\![\, S^\circ, R^\circ \,]\!\!] \tag{18}$$

$$V \cdot [\!\![\, S, R \,]\!\!] = [\!\![\, T, R \,]\!\!] \quad \Leftarrow \quad V \cdot S = T \cdot (\mathsf{F}\, V) \tag{19}$$

$$[\!\![\, S, R \,]\!\!] \cdot V = [\!\![\, S, U \,]\!\!] \quad \Leftarrow \quad R \cdot V = \mathsf{F}\, V \cdot U \tag{20}$$

Factorização iterativa: para θ associativa, tem-se

$$\begin{array}{rcll} \langle \mu\, f \;::\; p \rightarrow b\;,\; \theta \cdot \langle d, f \cdot e \rangle \rangle & = & p \rightarrow b\;,\; \theta \cdot (id \times b) \cdot w \cdot \langle d, e \rangle & (21) \\ & & \textit{onde} & \\ & & w = \underline{\textit{while}}\; (\neg \cdot p \cdot \pi_2)\; \underline{\textit{do}}\; \langle \theta \cdot (id \times d), e \cdot \pi_2 \rangle & \end{array}$$

Sendo (θ, u) um monoide, tem-se

$$\langle \mu\, f \;::\; p \rightarrow \underline{u}\;,\; \theta \cdot \langle d, f \cdot e \rangle \rangle \quad = \quad \pi_1 \cdot w \cdot \langle \underline{u}, id \rangle \tag{22}$$

onde w é o mesmo que em (21).