Métodos Formais em Engenharia de Software

1.º Ano de Mestrado de Informática da Universidade do Minho
Ano Lectivo de 2007/08

2ª Prova de avaliação — Teórica (manhã) — 10 de Julho 2008
09h30
Sala 1.08

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

PROVA COM CONSULTA (2 horas)

Questão 1 O caso de estudo das listas de chamadas de um telefone móvel abordado nesta unidade curricular envolve a função paramétrica que se segue, escrita em sintaxe VDM:

```
take[@A]: nat -> seq of @A -> seq of @A
take(n)(l) == if (l = [] or n <= 0) then [ ] else [hd l] ^ take[@A](n-1)(tl l);
```

Mesmo que não soubéssemos a definição desta função polimórfica, saberíamos sempre calcular, pelo teorema de Reynolds, uma propriedade (dita *"grátis"*) que a função satisfaz.

Complete o processo de cálculo dessa propriedade que a seguir se inicia, justificando **todos** os passos:

$$
\begin{array}{ll}
 & take : a^\star \leftarrow a^\star \leftarrow I\!N \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & take(R_{(a^\star \leftarrow a^\star) \leftarrow I\!N})take \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & take((R_a{}^\star \leftarrow R_a{}^\star) \leftarrow id)take \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & take \subseteq (R_a{}^\star \leftarrow R_a{}^\star) \cdot take \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & \vdots
\end{array}
$$

Questão 2 Diz-se que uma especificação S é implementada por R, escrevendo-se $S \vdash R$, sempre que R é mais definida e mais determinística que S (onde esta está definida), isto é:

$$S \vdash R \quad \equiv \quad (\delta\, S \subseteq \delta\, R) \ \wedge \ (R \cdot \delta\, S \subseteq S) \tag{1}$$

Seja S a relação especificada pelo seguinte par **pre**/**post**:

$$S : (y : I\!R) \leftarrow (x : I\!R)$$
pre TRUE
post $y - x > 2$

Encontre duas funções r e s tal que $S \vdash r$ e $S \not\vdash s$, justificando a sua resposta.

Questão 3 É conhecido o papel da lei

$$A \rightharpoonup (D \times (B \rightharpoonup C)) \quad \underset{\bowtie_n}{\overset{\triangle_n}{\leq}} \quad (A \rightharpoonup D) \times (A \times B \rightharpoonup C) \tag{2}$$

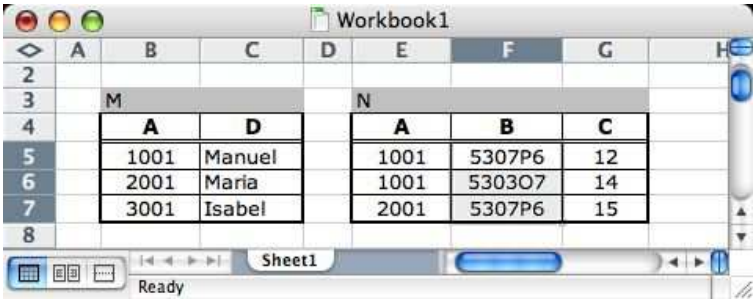
na decomposição de modelos de dados com vista à sua implementação em RDBMS.

1. Escrita em notação VDM, (2) fica

$$\texttt{map}\ A\ \texttt{to}\ (D * (\texttt{map}\ B\ \texttt{to}\ C)) \quad \underset{njoin}{\overset{unnjoin}{\leq}} \quad (\texttt{map}\ A\ \texttt{to}\ D) * (\texttt{map}\ A * B\ \texttt{to}\ C) \tag{3}$$

   Defina, em VDM (ou noutra notação formal que tenha aprendido na UCE), uma das funções `njoin` e `unnjoin`, à sua escolha.

2. A figura representa a situação de duas *tabelas* M e N tais que $(M, N) = unnjoin\ S$:



   Sabendo que $unnjoin$ é injectiva, identifique S. Sabendo que $njoin$ não é injectiva, identifique dois outros *mappings* (M', N') cuja abstracção seja também S.

3. Para que $njoin$ realize um *join* sem perdas — isto é, tal que $unnjoin(njoin(M, N)) = (M, N)$ se verifique — é preciso que esteja garantido o invariante descrito pelo rectângulo superior do diagrama que se segue:

$$\begin{array}{ccc} A & \xleftarrow{\pi_1} & A \times B \\ {\scriptstyle \delta\, M}\uparrow & \supseteq & \uparrow{\scriptstyle \delta\, N} \\ D \xleftarrow[M]{} A & \xleftarrow[\pi_1]{} & A \times B \xrightarrow[N]{} C \end{array}$$

   Indique, justificando, qual das duas formas

$$M \preceq N \cdot \pi_1^\circ \tag{4}$$

   ou

$$\delta\, M \xleftarrow{\pi_1} \delta\, N \tag{5}$$

   descreve esse invariante em notação-PF, e escreva-o em notação VDM *pointwise* (ou outra notação formal que tenha aprendido na UCE).

Questão 4 Atente na lei de refinamento

$$A^+ \times B \underset{F}{\overset{lstr}{\leq}} (A \times B)^+$$

onde A^+ designa o tipo das listas não vazias com elementos em A e defina a representação $lstr$ e a abstracção F numa notação à sua escolha.

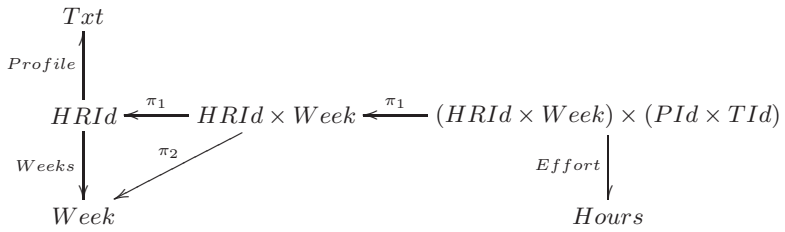
Questão 5 De acordo com o modelo de gestão de uma dada empresa de informática, os programadores e outros recursos humanos ($HResource$) apresentam, todas as semanas, um 'time card' que indica o número de horas que dedicaram a cada tarefa/projecto em curso. Esta informação é depois cruzada com a estimativa 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:

```
types

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

Com recurso às leis de refinamento de dados que foram estudadas nesta unidade curricular, alguém obteve o seguinte refinamento de `Db`, desenhado sob a forma de um diagrama relacional:



Justifique (ou conteste) 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*.

Questão 6 Considere o seguinte modelo VDM que especifica, abstractamente, a estrutura de um sistema de informação que, baseado no 'World Wide Web', dá a garantia de **integridade referencial**:

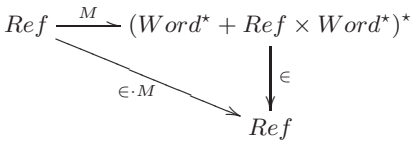
```
WWW  = map Ref to URL           -- URL=Universal Resource Location
       inv M == forall k in set dom M &
                   (exists i in set inds M(k) &
                       is_HyperLink(M(k)(i))) => M(k)(i).link in set dom M;
URL  = seq of Unit;
Unit = PlainText | HyperLink;
PlainText = seq of Word;
Word = seq of char;
HyperLink :: link: Ref           -- reference to another site
             txt: PlainText;     -- "underlined text"
Ref = token ;
```

Sabendo que

$$x \in l \quad \equiv \quad \langle \exists\, i \,:\, i \in \mathit{inds}\, l \,:\, x \in (l\, i) \rangle \tag{6}$$

é a relação de pertença associada a listas, complete o cálculo que se segue e que mostra que o invariante sobre o tipo `WWW` é a conversão para VDM *pointwise* de

$$\mathit{inv}\ M \stackrel{\text{def}}{=} (\in \cdot M)^\circ \preceq M$$



$$\equiv \quad \{ \ \ldots\ldots \}$$
$$\delta\,((\in \cdot M)^\circ) \subseteq \delta\, M$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$(\in \cdot M)^\circ \subseteq \top \cdot M$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$k'(\in \cdot M)k \ \Rightarrow\ k(\top \cdot M)k'$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$\langle \exists\, x \,::\, k' \in x \ \wedge\ x\, M\, k \rangle \ \Rightarrow\ k' \in \mathit{dom}\, M$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$(k \in \mathit{dom}\, M \ \wedge\ k' \in M\, k) \ \Rightarrow\ k' \in \mathit{dom}\, M$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$(k \in \mathit{dom}\, M \ \wedge\ \langle \exists\, i \,:\, i \in \mathit{inds}(Mk) \,:\, k' \in (M\, k)i \rangle) \ \Rightarrow\ k' \in \mathit{dom}\, M$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$(k \in \mathit{dom}\, M \ \wedge\ \langle \exists\, i \,:\, i \in \mathit{inds}(Mk) \,:\, k'[\bot\,, \in]\, (M\, k)i \rangle) \ \Rightarrow\ k' \in \mathit{dom}\, M$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$(k \in \mathit{dom}\, M \ \wedge\ \langle \exists\, i \,:\, i \in \mathit{inds}(Mk) \,:\, k'\, (\in \cdot i_2^\circ)\, (M\, k)i \rangle) \ \Rightarrow\ k' \in \mathit{dom}\, M$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$(k \in \mathit{dom}\, M \ \wedge\ \langle \exists\, i \,:\, i \in \mathit{inds}(Mk) \,:\, \langle \exists\, y \,::\, (M\, k)i = i_2\, y \ \wedge\ k' = \pi_1\, y \rangle \rangle) \ \Rightarrow\ k' \in \mathit{dom}\, M$$
$$\equiv \quad \{ \ \ldots\ldots \}$$
$$(k \in \mathit{dom}\, M \ \wedge\ \langle \exists\, i \,:\, i \in \mathit{inds}(Mk) \,:\, \langle \exists\, y \,::\, (M\, k)i = i_2\, y \rangle \rangle) \ \Rightarrow\ (\pi_1\, y) \in \mathit{dom}\, M$$