Opção 6 — Complementos de Especificação e Semântica
5º Ano da LMCC (7009O4)
Ano Lectivo de 1995/96

Exame (1ª chamada) — 25 de Janeiro de 1996
9h30
Sala 2013

NB: *sempre que os seus cálculos utilizarem resultados, leis, teoremas* etc. *disponíveis no material pedagógico desta disciplina não os copie para a sua folha de exame — limite-se a citar a sua referência* (e.g. *lei (A.68), exercício 8.16,* etc.*).*

PROVA COM CONSULTA (3 $horas$)

Questão 1 Demonstre ou refute os seguintes factos, em $Sets$:

$$2^{A \hookrightarrow B} \cong A \hookrightarrow (2^B + 1) \tag{1}$$

$$A^2 + 1 + 2 \times A \trianglelefteq 3 \hookrightarrow A \tag{2}$$

para A e B quaisquer.

RESOLUÇÃO: Comecemos por refutar (1) fazendo $A = 0$ e simplificando:

$$\begin{array}{rcl}
2^{0 \hookrightarrow B} & \cong & 0 \hookrightarrow (2^B + 1) \\
& \updownarrow & (8.39) \\
2^{((B+1)^0)} & \cong & ((2^B + 1) + 1)^0 \\
& \updownarrow & (8.33) \\
2^1 & \cong & 1 \\
& \updownarrow & (8.35) \\
2 & \not\cong & 1
\end{array}$$

Passando ao facto (2), mostra-se a seguir que ele é verdadeiro:

$$\begin{array}{rcl}
A^2 + 1 + 2 \times A & \trianglelefteq & 3 \hookrightarrow A \\
& \updownarrow & (8.90) \\
(A+1)^2 & \trianglelefteq & 3 \hookrightarrow A \\
& \updownarrow & (8.39) \\
(A+1)^2 & \trianglelefteq & (A+1)^3 \\
& \updownarrow & \\
(A+1)^2 & \trianglelefteq & (A+1)^2 \times (A+1) \\
& \updownarrow & (8.59) \\
& V &
\end{array}$$

□

Questão 2 Seja $i : A \longrightarrow C$ uma bijecção e defina-se a função $i \hookrightarrow B$ como se segue:

$$\begin{array}{rcl}
i \hookrightarrow B & : & (A \hookrightarrow B) \longrightarrow (C \hookrightarrow B) \\
(i \hookrightarrow B)(\sigma) & \stackrel{\text{def}}{=} & \left(\begin{array}{c} i(a) \\ \sigma(a) \end{array}\right)_{a \in dom(\sigma)}
\end{array}$$

(**NB:** Poderá ver em $i \hookrightarrow B$ a aplicação do functor $\lambda X.X \hookrightarrow B$ a i.)

1. Mostre que, para $(i \hookrightarrow B)(\sigma)$ ser uma construção bem definida, não é necessário que i seja bijectiva, bastando que se verifique o facto seguinte, entre i e σ:

$$\forall a, a' \in dom(\sigma) : i(a) = i(a') \Rightarrow \sigma(a) = \sigma(a')$$

2. Verifique se a função de abstracção

$$\bowtie_n (\sigma, \tau) \stackrel{\text{def}}{=} \left(\begin{array}{c} a \\ \langle \sigma(a), \left(\begin{array}{c} \pi_2(p) \\ \tau(p) \end{array} \right)_{p \in sel(a,\tau)} \rangle \end{array} \right)_{a \in dom(\sigma)}$$

referente à lei (8.72) se pode redefinir como se segue:

$$\bowtie_n (\sigma, \tau) \stackrel{\text{def}}{=} \left(\begin{array}{c} a \\ \langle \sigma(a), (\pi_2 \hookrightarrow C)(\tau \mid (\{a\} \times B)) \rangle \end{array} \right)_{a \in dom(\sigma)}$$

isto é, recorrendo à construção estudada na alínea anterior.

RESOLUÇÃO:

1. Para $(i \hookrightarrow B)(\sigma)$ ser uma função (finita) em $C \hookrightarrow B$ é necessário que, vista como uma relação r,

$$r = \{\langle i(a), \sigma(a) \rangle \mid \ a \in dom(\sigma)\}$$

esta fatisfaça fdp (C.31), *i.é* que

$$\begin{array}{c} \forall p, p' \in r : \pi_2(p) \neq \pi_2(p') \Rightarrow \pi_1(p) \neq \pi_1(p') \\ \updownarrow \\ \forall p, p' \in r : \pi_1(p) = \pi_1(p') \Rightarrow \pi_2(p) = \pi_1(p') \\ \updownarrow \\ \forall a, a' \in dom(\sigma) : i(a) = i(a') \Rightarrow \sigma(a) = \sigma(a') \end{array}$$

como o exercício sugere. Em suma, a situação seguinte não pode acontecer:

$$\begin{tikzcd} B & A \arrow[l, "\sigma"'] \arrow[r, "i"] & C \\ \sigma(a) & a \arrow[l] \arrow[dr] & \\ & & i(a) = i(a') \\ \sigma(a') & a' \arrow[l] \arrow[ur] & \end{tikzcd}$$

i.é "i não pode unificar mais do que σ".

2. Basta trabalhar $(\pi_2 \hookrightarrow C)(\underbrace{\tau \mid (\{a\} \times B)}_{\tau'})$. Antes de mais, tem-se:

$$\begin{array}{rcl} \tau' & = & \tau \mid (\{a\} \times B) \\ & = & \left(\begin{array}{c} p \\ \tau(p) \end{array} \right)_{p \in (dom(\tau) \cap (\{a\} \times B))} \\ & = & \left(\begin{array}{c} p \\ \tau(p) \end{array} \right)_{p \in \{x \in dom(\tau) \mid \ \pi_1(x) = a \wedge \pi_2(x) \in B\}} \\ & = & \left(\begin{array}{c} p \\ \tau(p) \end{array} \right)_{p \in \{\langle a', b \rangle \in dom(\tau) \mid \ a' = a\}} \\ & = & \left(\begin{array}{c} p \\ \tau(p) \end{array} \right)_{p \in sel(a,\tau)} \end{array}$$

Vamos agora verificar se $(\pi_2 \hookrightarrow C)(\tau')$ está bem definida:

$$\begin{array}{c}
\forall x, x' \in dom(\tau') : \pi_2(x) = \pi_2(x') \Rightarrow \tau'(x) = \tau'(x') \\
\updownarrow \\
\forall x, x' \in sel(a, \tau) : \pi_2(x) = \pi_2(x') \Rightarrow \tau'(x) = \tau'(x') \\
\updownarrow \\
\forall x, x' \in sel(a, \tau) : x = x \Rightarrow \tau'(x) = \tau'(x') \\
\updownarrow \\
V
\end{array}$$

Calculando, então, $(\pi_2 \hookrightarrow C)(\tau')$:

$$\pi_2 \hookrightarrow C)(\tau') \quad = \quad \begin{pmatrix} \pi_2(p) \\ \tau(p) \end{pmatrix}_{p \in sel(a,\tau)}$$

que é exactamente o que encontramos na definição de $\bowtie_n$ (8.74). Logo, o facto em causa é verdadeiro. $\square$

Questão 3 Demonstre ou refute os factos seguintes, em que ocorre a construção introduzida na questão 2, e que lhe poderão ser úteis em raciocínios envolvendo funções de abstracção obtidas por composição estrutural de saltos de refinamento:

$$dom((i \hookrightarrow B)(\sigma)) = 2^i(dom(\sigma)) \tag{3}$$

$$2^f \circ elems \circ g^* = 2^{(f \circ g)} \circ elems \tag{4}$$

RESOLUÇÃO: A prova de (3) é imediata, uma vez que se tem:

$$\begin{array}{rcl}
dom((i \hookrightarrow B)(\sigma)) & = & dom(\begin{pmatrix} i(a) \\ \sigma(a) \end{pmatrix}_{a \in dom(\sigma)}) \\
& = & \{i(a) \mid \ a \in dom(\sigma)\} \\
& = & i[\{a \mid \ a \in dom(\sigma)\}] \\
& = & 2^i(dom(\sigma))
\end{array}$$

Prova de (4):

$$\begin{array}{rcl}
(2^f \circ elems \circ g^*)(l) & = & 2^f(elems(g^*(l))) \\
& = & 2^f(elems(<g(a) \mid a \leftarrow l>)) \\
& \underset{(A.70)}{=} & 2^f(g[elems(l)]) \\
& = & 2^f(2^g(elems(l))) \\
& = & (2^f \circ 2^g)(elems(l)) \\
& = & 2^{f \circ g}(elems(l)) \\
& = & (2^{f \circ g} \circ elems)(l)
\end{array}$$

NB: A justificação de $2^f \circ 2^g = 2^{(f \circ g)}$ não carece de prova por $\lambda X.2^X$ ser um functor. $\square$

Questão 4 Relembre o exercício 8.16, onde a estrutura de informação

$$\begin{array}{rcl}
Equip & \cong & \#Equip \hookrightarrow (\#Unid \hookrightarrow \mathbb{N})_+ \\
\#Unid & \cong & \#Comp + \#Equip
\end{array}$$

(que é parte do modelo de DPP) se implementa sob a forma de duas tabelas EC e EE:

$$
\begin{array}{rrll}
DPP1 & \cong & \quad\cdots & \times \\
 & EC: & 2^{Row3} & \times \quad \textit{/*tabela de decomposição de */} \\
 & & & \qquad \textit{/*equipamentos em seus componentes */} \\
 & EE: & 2^{Row4} & \qquad \textit{/*tabela de decomposição de */} \\
 & & & \qquad \textit{/*equipamentos em sub-equipamentos */}
\end{array}
$$

onde

$$
\begin{array}{rcrll}
Row3 & \cong & Eq: & \#Equip & \times \\
 & & Co: & \#Comp & \times \\
 & & Nr: & \mathbb{N} & \\
Row4 & \cong & Eq: & \#Equip & \times \\
 & & Se: & \#Equip & \times \\
 & & Nr: & \mathbb{N} &
\end{array}
$$

É a estas tabelas que se refere o seguinte SQL embebido em Pro-C, extraído de um anexo do relatório de um grupo do ano passado:

```
EXEC SQL CREATE TABLE EC
      ( Eq VARCHAR(15), Co VARCHAR(15), Nr NUMBER, PRIMARY KEY (Eq,Co), ...);
```

e

```
EXEC SQL CREATE TABLE EE
      ( Eq VARCHAR(15), Se VARCHAR(15), Nr NUMBER, PRIMARY KEY (Eq,Se), ...);
```

Transcreve-se agora da mesma fonte o diagrama de refinamento de uma operação x sobre $Equip$, que se deixa propositadamente incompleto,

$$
\begin{tikzcd}
Equip \times \ldots \arrow[r, "x"] & \ldots \\
(2^{Row3} \times 2^{Row4}) \times \ldots \arrow[u, "\ldots \times id"] \arrow[r, "y"'] & \ldots \arrow[u, "f"']
\end{tikzcd}
$$

e onde apenas se sabe que $f(\langle r, s\rangle) = j_1(mkf(r)) \cup j_2(mkf(s))$ (onde $j_{k=1,2} = i_k \hookrightarrow \mathbb{N}$, sendo $i_{k=1,2}$ os operadores de injecção no co-produto), e que y é da forma

$$
\begin{array}{rll}
y(\langle ec, ee\rangle, a) \stackrel{\mathrm{def}}{=} & let & r = \ldots ec \ldots a \ldots \\
 & & s = \ldots ee \ldots a \ldots \\
 & in & \langle r, s\rangle
\end{array}
$$

Pesquisando o mesmo anexo, vem a descobrir-se que o código SQL embebido em Pro-C que codifica y inclui:

```
EXEC SQL SELECT Co,Nr FROM EC WHERE Eq=a INTO r;
EXEC SQL SELECT Se,Nr FROM EE WHERE Eq=a INTO s;
```

1. Preencha as reticências do diagrama, indicando que leis dos $Sets$ estão em jogo.
2. Qual lhe parece ser a especificação formal de x? Justifique convenientemente

---

RESOLUÇÃO:

1. Sejam α, β, *etc.* identificadores das reticências a preencher em:

$$
\begin{tikzcd}
Equip \times \alpha \arrow[r, "x"] & \beta \\
(2^{Row3} \times 2^{Row4}) \times \omega \arrow[u, "\gamma \times id"] \arrow[r, "y"'] & \delta \arrow[u, "f"']
\end{tikzcd}
$$

Obviamente, $\alpha = \omega$. Há então que inferir $\alpha, \beta, \gamma, \delta$. Antes de mais, vamos completar y:

$$
\begin{array}{rll}
y(\langle ec, ee\rangle, a) \stackrel{\mathrm{def}}{=} & let & r = \{\langle Co(t), Nr(t)\rangle \mid t \in ec \wedge Eq(t) = a\} \\
 & & s = \{\langle Se(t), Nr(t)\rangle \mid t \in ee \wedge Eq(t) = a\} \\
 & in & \langle r, s\rangle
\end{array}
$$

Daqui, infere-se:

$$
\begin{array}{rll}
a \in \#Equip & \textit{cf.} & Eq : \#Equip \\
r \in 2^{\#Comp \times \mathbb{N}} & \textit{cf.} & Co : \#Comp \times Nr : \mathbb{N} \ldots \\
s \in 2^{\#Equip \times \mathbb{N}} & \textit{cf.} & Se : \#Equip \times Nr : \mathbb{N} \ldots
\end{array}
$$

Logo $\alpha = \#Equip$ e $\delta = 2^{\#Comp \times \mathbb{N}} \times 2^{\#Equip \times \mathbb{N}}$.
Vejamos agora que se pode inferir de f o seguinte:

$$mkf(r) \in \#Comp \hookrightarrow \mathbb{N} \quad e \quad mkf(s) \in \#Equip \hookrightarrow \mathbb{N}$$

cf. (8.63):

$$(i_1 \hookrightarrow \mathbb{N})(mkf(r)) \cup (i_2 \hookrightarrow \mathbb{N})(mkf(s)) \quad \in \quad (\#Comp + \#Equip) \hookrightarrow \mathbb{N}$$

Assim, o co-domínio de f é $\beta = \#Unid \hookrightarrow \mathbb{N}$.
Falta apenas γ, que é uma parte da função de abstracção entre $DPP1$ e DPP, *cf.* resolução do exercício 8.16:

$$
\begin{array}{rcl}
\gamma & = & h \circ (i \hookrightarrow \mathbb{N}) \circ f_{(8.63)} \circ \lambda\langle \rho_4, \rho_5\rangle.\langle \rho_4, \rho_5\rangle \circ (2^j \times 2^j) \circ (mkf \times mkf) \circ (id \times id) \\
 & = & h \circ (i \hookrightarrow \mathbb{N}) \circ f_{(8.63)} \circ \lambda\langle \rho_4, \rho_5\rangle.\langle \rho_4, \rho_5\rangle \circ ((mkf \circ 2^j) \times (mkf \circ 2^j)) \\
 & = & h \circ (i \hookrightarrow \mathbb{N}) \circ f_{(8.63)} \circ ((mkf \circ 2^j) \times (mkf \circ 2^j))
\end{array}
$$

2. Conjectura-se que seja:

$$
x(\sigma, a) \quad \stackrel{\mathrm{def}}{=} \quad \left\{ \begin{array}{ccc} a \in dom(\sigma) & \Rightarrow & \sigma(a) \\ \neg(a \in dom(\sigma)) & \Rightarrow & (\quad) \end{array} \right.
$$

Vejamos que, de facto, x é implementada por y. A equação de refinamento é:

$$x(\gamma(r_1, r_2)) \quad = \quad f(y(r_1, r_2))$$

Justificação informal: é fácil de ver que as duas tabelas r e s contêm exactamente as quantidades de componentes ou sub-equipamentos de a. O que f faz é transformar essas tabelas em multiconjuntos (mkf), afixar as estiquetas 1 e 2 nos elementos dos domínios dessas funções e assim garantir uma boa união, que vem a ser o multicomjunto em $\#Unid \hookrightarrow \mathbb{N}$ que descreve a expansão de a. No caso de a não ser um equipamento conhecido, essas tabelas serão vazias, daí resultando, por y, a função finita vazia, tal como x devolve.
NB: A justificação formal não é difícil se fizermos

$$f_{(8.63)}(\sigma_1, \sigma_2) = (i_1 \hookrightarrow A)(\sigma_1) \cup (i_2 \hookrightarrow A)(\sigma_2)$$

no desenvolvimento da equação de refinamento.

□

Questão 5 Pretende-se simular a função

$$
\begin{array}{rcl}
sum & : & 2^{\mathbb{N}_0} \longrightarrow \mathbb{N}_0 \\
sum(S) & \stackrel{\mathrm{def}}{=} & \sum_{n \in S} n
\end{array}
$$

no contexto do refinamento "clássico" de conjuntos para sequências (listas) que é estabelecido pela função $elems$:

$$
\begin{tikzcd}
2^{\mathbb{N}_0} \arrow[r, "sum"] & \mathbb{N}_0 \\
{\mathbb{N}_0}^{\star} \arrow[u, "elems"] \arrow[r, "sums"'] & \mathbb{N}_0 \arrow[u, "id"']
\end{tikzcd}
$$

Suponha que alguém já fez o cálculo de $sums$ e chegou a:

$$
\begin{array}{rcl}
sums & : & \mathbb{N}_0{}^\star \longrightarrow \mathbb{N}_0 \\
sums(l) & \stackrel{\rm def}{=} & \left\{ \begin{array}{rcl} l = <> & \Rightarrow & 0 \\ \neg(l = <>) & \Rightarrow & head(l) + sums(tail(l)) \end{array} \right.
\end{array}
$$

Terá o cálculo de $sums$ sido bem feito? Justifique convenientemente a resposta.

RESOLUÇÃO: À primeira vista parece mal feito, pois $+$ não é idempotente, *i.é* $a + a \neq a$. Logo, eventuais repetições na lista que é argmento de $sums$ vão distorcer o resultado, não se verificando a equação de refinamento:

$$sum(elems(l)) \quad = \quad sums(l)$$

Contra-exemplo ($l = <a, a>$):

$$sum(elems(<a, a>)) \quad = \quad a$$

enquanto que:

$$
\begin{array}{rcl}
sums(<a, a>) & = & a + sums(<a>) \\
 & = & a + (a + sums(<>)) \\
 & = & a + (a + 0) \\
 & = & 2a
\end{array}
$$

Qual foi o vício de raciocínio, então? Vamos tentar inspeccionar o cálculo feito:

$$
\begin{array}{rcl}
sums(l) & = & sum(elems(l)) \\
 & = & \displaystyle\sum_{n \in elems(l)} n \\
 & = & \left\{ \begin{array}{rcl} l = <> & \Rightarrow & \sum_{n \in elems(<>)} n \\ \neg(l = <>) & \Rightarrow & \sum_{n \in \{head(l)\} \cup elems(tail(l))} n \end{array} \right. \\
 & \neq & \left\{ \begin{array}{rcl} l = <> & \Rightarrow & 0 \\ \neg(l = <>) & \Rightarrow & \sum_{n \in \{head(l)\}} + \sum_{elems(tail(l))} n \end{array} \right. \\
 & = & \left\{ \begin{array}{rcl} l = <> & \Rightarrow & 0 \\ \neg(l = <>) & \Rightarrow & head(l) + sums(tail(l)) \end{array} \right.
\end{array}
$$

O vício ocorreu em $\neq$, pois:

$$\Theta_{a \in A \cup B} x(a) \quad \neq \quad \Theta_{a \in A} x(a) \theta \Theta_{a \in B} x(a)$$

em geral. $\square$

Questão 6 O seguinte diagrama de refinamento,

$$
\begin{tikzcd}
(A \hookrightarrow B) \times A \arrow[r, "ap"] & B \cup \{NIL\} \\
2^{A \times B} \times A \arrow[u, "mkf \times id"] \arrow[r, "x"'] & 2^B \arrow[u, "y"']
\end{tikzcd}
$$

representa o exercício de implementação relacional da função de aplicação de uma função finita a um argumento,

$$
ap(\sigma, a) \stackrel{\rm def}{=} \left\{ \begin{array}{rcl} a \in dom(\sigma) & \Rightarrow & \sigma(a) \\ a \notin dom(\sigma) & \Rightarrow & NIL \end{array} \right.
$$

em que, para simplificar, se supõe sempre $NIL \notin B$ e assim se pode escrever $B \cup \{NIL\}$ em lugar de $B + 1$.

Suponha que quem calculou o refinamento acabou por codificar $x(r, a)$, em SQL, da forma que se segue

$$\texttt{SELECT } \pi_2 \texttt{ FROM } r \texttt{ WHERE } a = \pi_1\texttt{;}$$

mas perdeu o documento que continha todo o processo de cálculo. Deduza a função de abstracção y que se ajusta a tal raciocínio.

RESOLUÇÃO: Do SQL proposto infere-se

$$x(r, a) \stackrel{\text{def}}{=} \{\pi_2(t) \mid t \in r \wedge a = \pi_1(t)\}$$

Ter-se-á:

$$\begin{array}{rcl}
ap(mkf(r), a) & = & y(\{\pi_2(t) \mid t \in r \wedge a = \pi_1(t)\}) \\
& \updownarrow & \\
\left\{ \begin{array}{rcl} a \in dom(mkf(r)) & \Rightarrow & mkf(r)(a) \\ \neg(a \in dom(mkf(r))) & \Rightarrow & NIL \end{array} \right. & = & y(\{\pi_2(t) \mid t \in r \wedge a = \pi_1(t)\}) \\
& \updownarrow & \\
\left\{ \begin{array}{rcl} a \in \pi_1[r] & \Rightarrow & the(\{b \in B \mid a\ r\ b\}) \\ \neg(a \in \pi_1[r]) & \Rightarrow & NIL \end{array} \right. & = & y(\{\pi_2(t) \mid t \in r \wedge a = \pi_1(t)\}) \\
& \updownarrow & \\
\left\{ \begin{array}{rcl} a \in \pi_1[r] & \Rightarrow & the(\{\pi_2(t) \mid t \in r \wedge a = \pi_1(r)\}) \\ \neg(a \in \pi_1[r]) & \Rightarrow & NIL \end{array} \right. & = & y(\left\{ \begin{array}{rcl} a \in \pi_1[r] & \Rightarrow & \{\pi_2(t) \mid t \in r \wedge a = \pi_1(r)\} \\ \neg(a \in \pi_1[r]) & \Rightarrow & \emptyset \end{array} \right.)
\end{array}$$

Assim, podemos inferir:

$$y(S) \stackrel{\text{def}}{=} \left\{ \begin{array}{rcl} S = \emptyset & \Rightarrow & NIL \\ \neg(S = \emptyset) & \Rightarrow & the(S) \end{array} \right.$$

□
