Métodos Formais de Programação II +
Especificação e Semântica +
Opção I - Especificação e Semântica

4.º Ano da LMCC (7008N2 + 7007A4) + LESI (5307C3)
Ano Lectivo de 1995/96

Exame (1.ª chamada) — 25 de Junho 1996
14h30
Salas 2211 e 2212

Importante:
Os alunos que pretendem ser avaliados à disciplina Métodos Formais de Programação II *(7008N2) devem resolver as questões 4, 5, 6, 7, 8 e 9.*
Os alunos que pretendem ser avaliados à disciplina Especificação e Semântica *(7007A4) ou* Opção I - Especificação e Semântica *(5307C3) devem resolver as questões 1, 2, 3, 4, 5 e 9.*

PROVA COM CONSULTA (3 $horas$)

Questão 1 Considere a seguinte definição de uma função sobre sequências:

$$
\begin{array}{l}
subl : A^\star \times \mathbb{N} \times \mathbb{N}_0 \to A^\star \\
subl(l, i, n) \stackrel{\text{def}}{=} \\
\quad \left\{ \begin{array}{lcll}
n = 0 \lor l = <> & \Rightarrow & <> & \\
n > 0 \land l \neq <> & \Rightarrow & let & h = head(l) \\
 & & & t = tail(l) \\
 & & in & \left\{ \begin{array}{lcl} i = 1 & \Rightarrow & <h> \frown subl(t, i, n-1) \\ i > 1 & \Rightarrow & subl(t, i-1, n) \end{array} \right.
\end{array} \right.
\end{array}
$$

1. Comece por preencher o quadro

i	n	$subl(l, i, n)$
2	2	
3	3	
4	3	

para $l = <2, 3, 1, 1, 2>$. Descreva, então, por palavras suas, o propósito e a utilidade da função $subl$.

2. Mostre (*e.g.* por indução em $A^\star$) que o seguinte facto se verifica:

$$subl(x, 1, length(x)) \quad = \quad x \tag{1}$$

RESOLUÇÃO:

1. Tem-se: para $i = n = 2$:

$$
\begin{array}{rcl}
subl(<2, 3, 1, 1, 2>, 2, 2) & = & subl(<3, 1, 1, 2>, 1, 2) \\
 & = & <3> \frown subl(<1, 1, 2>, 1, 1) \\
 & = & <3> \frown <1> \frown subl(<1, 2>, 1, 0) \\
 & = & <3> \frown <1> \frown <> \\
 & = & <3, 1>
\end{array}
$$

para $i = n = 3$:

$$\begin{array}{rcl} subl(<2,3,1,1,2>,3,3) & & subl(<3,1,1,2>,2,3) \\ & = & subl(<1,1,2>,1,3) \\ & = & <1> \frown subl(<1,2>,1,2) \\ & = & <1,1> \frown subl(<2>,1,1) \\ & = & <1,1,2> \frown subl(<>,1,0) \\ & = & <1,1,2> \frown <> \\ & = & <1,1,2> \end{array}$$

para $i = 4, n = 3$:

$$\begin{array}{rcl} subl(<2,3,1,1,2>,4,3) & = & subl(<3,1,1,2>,3,3) \\ & = & subl(<1,1,2>,2,3) \\ & = & subl(<1,2>,1,3) \\ & = & <1> \frown subl(<2>,1,2) \\ & = & <1,2> \frown subl(<>,1,1) \\ & = & <1,2> \frown <> \\ & = & <1,2> \end{array}$$

A intuição diz-nos que $subl$ extrai de uma sequência l a sua subsequência de n elementos (ou os que fôr possível extrair) que começa na i-esima posição.

2. O facto a provar é

$$\forall x \in A^\star : subl(x, 1, length(x)) = x$$

(= x é idêntica à sua própria sub-sequência de igual comprimento a começar na 1.ª posição).

Prova por indução sobre $A^\star$:

(a) *Caso de Base*: ($x = <>$). Tem-se, neste caso, trivialmente:

$$\begin{array}{rcl} subl(<>, 1, length(<>)) = <> & \Leftrightarrow & <> = <> \\ & \Leftrightarrow & V \end{array}$$

(b) *Salto indutivo*: ($x \neq <>$)

i. *Hipótese de indução*:

$$\forall x \in A^\star : subl(tail(x), 1, length(tail(x))) = tail(x)$$

ii. *Passo*: Teremos, sucessivamente:

$$\begin{array}{rcl} subl(x, 1, length(x)) & \underset{\text{pois } length(x) > 0}{=} & \begin{array}{ll} let & h = head(x) \\ & t = tail(l) \\ in & <h> \frown subl(t, 1, length(x) - 1) \end{array} \\ & \underset{\text{por (A.76)}}{=} & <head(x)> \frown subl(tail(x), 1, length(tail(x))) \\ & \underset{\text{pela H.indução}}{=} & <head(x)> \frown tail(x) \\ & = & x \end{array}$$

como se pretendia demonstrar.

□

Questão 2 Suponha que alguém especifica um *plano de estudos* de uma licenciatura como sendo um conjunto de disciplinas,

$$PlanoEstudos \quad \cong \quad 2^{Disciplina}$$

onde uma disciplina ou é obrigatória ou opcional,

$$Disciplina \quad \cong \quad Obrigat + Opcional$$

sendo uma disciplina obrigatória caracterizada por uma série de atributos,

$$
\begin{array}{rcllr}
Obrigat & \cong & A : 5 & \times & \textit{/*ano do curso */} \\
 & & AC : ACien & \times & \textit{/*área científica */} \\
 & & N : Nome & \times & \textit{/*nome */} \\
 & & R : 3 & \times & \textit{/*regime (1.º sem., 2.º sem., anual) */} \\
 & & E : Esc & & \textit{/*escolaridade,} \text{ etc. */}
\end{array}
$$

e uma disciplina opcional caracterizada como se segue,

$$
\begin{array}{rcllr}
Opcional & \cong & A : 5 & \times & \textit{/*ano do curso */} \\
 & & Nr : \mathbb{N} & \times & \textit{/*número da opção */} \\
 & & R : 3 & \times & \textit{/*regime (1.º sem., 2.º sem., anual) */} \\
 & & E : Esc & \times & \textit{/*escolaridade,} \text{ etc. */} \\
 & & O : 2^{Opcao} & & \textit{/*opções oferecidas */}
\end{array}
$$

onde cada opção se caracteriza por:

$$
\begin{array}{rcllr}
Opcao & \cong & AC : ACien & \times & \textit{/*área científica */} \\
 & & N : Nome & & \textit{/*nome */}
\end{array}
$$

Especifique funções sobre $PlanoEstudos$ capazes de calcular:

1. Os nomes de todas as disciplinas efectivamente leccionadas.
2. O conjunto das áreas científicas exclusivamente opcionais.

RESOLUÇÃO:

1.

$$
\begin{array}{rcll}
nomes & : & PlanoEstudos \rightarrow 2^{Nome} & \\
nomes(pe) & \stackrel{\text{def}}{=} & let & ob = \{d \mid d \in pe \wedge \text{is-}Obrigat(d)\} \\
 & & & op = pe - ob \\
 & & in & 2^{N}(ob) \cup \bigcup_{d \in op} 2^{N}(D(d))
\end{array}
$$

2.

$$
\begin{array}{rcll}
areas & : & PlanoEstudos \rightarrow 2^{ACien} & \\
areas(pe) & \stackrel{\text{def}}{=} & let & ob = \{d \mid d \in pe \wedge \text{is-}Obrigat(d)\} \\
 & & & op = pe - ob \\
 & & in & (\bigcup_{d \in op} 2^{AC}(D(d))) - 2^{AC}(ob)
\end{array}
$$

□

Questão 3 Tal como se referiu no exercício 2.41 (pág. 108 da sebenta), muitos predicados sobre funções parciais finitas $\sigma \in A \hookrightarrow B$ são da forma

$$\forall i \in dom(\sigma) : \phi(\sigma(i)) \tag{2}$$

onde $\phi : B \rightarrow 2$ é um predicado sobre B, sendo usada a notação

$$A \hookrightarrow \phi$$

como abreviatura da quantificação (2), para A finito.

Demonstre a seguinte propriedade dessa construção:

$$(A \hookrightarrow \phi(\sigma_1)) \wedge (A \hookrightarrow \phi(\sigma_2)) \quad \Rightarrow \quad A \hookrightarrow \phi(\sigma_1 \dagger \sigma_2) \tag{3}$$

Sugestão: relembre os factos provados no mesmo exercício, nomeadamente:

$$\begin{array}{rcl} A \hookrightarrow \phi(\sigma_1 \cup \sigma_2) & \Leftrightarrow & (A \hookrightarrow \phi(\sigma_1)) \wedge (A \hookrightarrow \phi(\sigma_2)) \\ A \hookrightarrow \phi(\sigma) & \Rightarrow & A \hookrightarrow \phi(\sigma \setminus S) \end{array}$$

RESOLUÇÃO: Sabemos que

$$\sigma_1 \dagger \sigma_2 = \sigma_1 \setminus dom(\sigma_2) \cup \sigma_2$$

cf. (A.49). Logo, pelos factos invocados do exercício 2.41, ter-se-á, sucessivamente:

$$\begin{array}{rcl} A \hookrightarrow \phi(\sigma_1 \dagger \sigma_2) & = & A \hookrightarrow \phi(\sigma_1 \setminus dom(\sigma_2) \cup \sigma_2) \\ & \Leftrightarrow & A \hookrightarrow \phi(\sigma_1 \setminus dom(\sigma_2)) \wedge A \hookrightarrow \phi(\sigma_2) \\ & \Leftarrow & A \hookrightarrow \phi(\sigma_1) \wedge A \hookrightarrow \phi(\sigma_2) \end{array}$$

QED. □

Questão 4 Verifique quais das funções em $\mathbb{N}$ $\lambda(n,m).n \times m$ (multiplicação de naturais) e $\lambda(n,m).n^m$ (exponenciação em $\mathbb{N}$) são pontos fixos da funcional seguinte:

$$F(f) \stackrel{\text{def}}{=} \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & y \\ x > 1 & \Rightarrow & y + f(x-1,y) \end{array} \right.$$

RESOLUÇÃO: No primeiro caso temos (o raciocínio decorre de propriedades elementares das operações $-$ e $+$ sobre $\mathbb{N}$):

$$\begin{array}{rcl} F(\lambda(n,m).n \times m) & = & \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & y \\ x > 1 & \Rightarrow & y + (x-1) \times y \end{array} \right. \\ & = & \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & y \\ x > 1 & \Rightarrow & y + x \times y - y \end{array} \right. \\ & = & \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & y \\ x > 1 & \Rightarrow & y - y + x \times y \end{array} \right. \\ & = & \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & y \\ x > 1 & \Rightarrow & 0 + x \times y \end{array} \right. \\ & = & \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & y \\ x > 1 & \Rightarrow & x \times y \end{array} \right. \\ & = & \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & 1 \times y \\ x > 1 & \Rightarrow & x \times y \end{array} \right. \\ & = & \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & x \times y \\ x > 1 & \Rightarrow & x \times y \end{array} \right. \\ & \underset{\text{por (B.2)}}{=} & \lambda(x,y).x \times y \end{array}$$

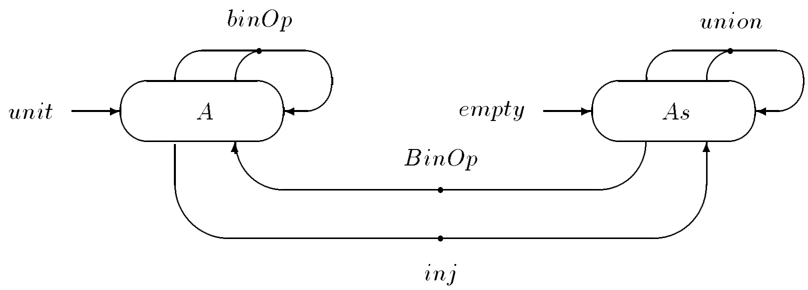
Logo a função dada é ponto fixo da funcional F.

No segundo caso temos:

$$\begin{array}{rcl} F(\lambda(n,m).n^m) & = & \lambda(x,y).\left\{ \begin{array}{lcl} x = 1 & \Rightarrow & y \\ x > 1 & \Rightarrow & y + (x-1)^y \end{array} \right. \\ & = & \lambda(x,y).y + (x-1)^y \end{array}$$

sendo imediatamente óbvio que $\lambda(n,m).n^m$ não é ponto fixo: veja-se o caso $x = 1$, em que $x^y = 1$ e a função dá y como resultado. □

Questão 5 Considere uma interface axiomática J cuja assinatura subjacente é descrita pelo seguinte diagrama-ADJ,



que vem equipada com o seguinte conjunto de axiomas:

$$
\begin{aligned}
BinOp(empty) &\equiv unit & (4)\\
BinOp(inj(i)) &\equiv i & (5)\\
BinOp(union(x,y)) &\equiv binOp(BinOp(x), BinOp(y)) & (6)
\end{aligned}
$$

Suponha que alguém definiu o seguinte modelo,

$$
\begin{cases}
MOD : I \longrightarrow J\\
MOD(A) \cong I.Item\\
MOD(As) \cong 2^A\\
MOD(empty) \stackrel{\text{def}}{=} \emptyset\\
MOD(union) \stackrel{\text{def}}{=} \lambda(x,y).x \cup y\\
MOD(unit) \stackrel{\text{def}}{=} I.\epsilon\\
MOD(inj) \stackrel{\text{def}}{=} \lambda a.\{a\}\\
MOD(binOp) \stackrel{\text{def}}{=} I.\theta\\
MOD(BinOp) \stackrel{\text{def}}{=} \lambda x. \begin{cases} x = \emptyset & \Rightarrow & I.\epsilon \\ \neg(x = \emptyset) & \Rightarrow & let \quad a \in x \\ & & in \quad binOp(a, BinOp(x - \{a\})) \end{cases}
\end{cases}
$$

parametrizado por uma interface argumento I, e pretende agora demonstrar que se trata de facto de um J-modelo.

1. Mostre que, qualquer que seja I, MOD satisfaz pelo menos um dos três J-axiomas.
2. Descubra qual é a "menor" axiomatização de I que garante que MOD passa a satisfazer todos os J-axiomas.

RESOLUÇÃO:

1. Trata-se do axioma (4), como se prova facilmente:

$$
\begin{array}{rcl}
MOD(BinOp(empty)) & = & MOD(unit)\\
& \updownarrow & \\
MOD(BinOp)(MOD(empty)) & = & I.\epsilon\\
& \updownarrow & \\
MOD(BinOp)(\emptyset) & = & I.\epsilon\\
& \updownarrow & \\
\begin{cases} \emptyset = \emptyset & \Rightarrow & I.\epsilon \\ \neg(\emptyset = \emptyset) & \Rightarrow & let \quad \ldots \\ & & in \quad \ldots \end{cases} & = & I.\epsilon\\
& \updownarrow & \\
I.\epsilon & = & I.\epsilon\\
& \updownarrow & \\
& V &
\end{array}
$$

2. Comecemos por ver o que é necessário para provar (5):

$$\begin{array}{rcl} MOD(BinOp(inj(i))) & = & MOD(i) \\ & \updownarrow & \\ MOD(BinOp)(\{i\}) & = & i \\ & \updownarrow & \\ \begin{array}{ll} let & a \in \{i\} \\ in & MOD(binOp)(a, MOD(BinOp)(\{i\} - \{a\})) \end{array} & = & i \\ & \updownarrow & \\ I.\theta(i, MOD(BinOp)(\emptyset)) & = & i \\ & \updownarrow & \\ I.\theta(i, I.\epsilon) & = & i \end{array}$$

Logo, o I-axioma que é preciso para completar a prova acima é

$$\theta(i, \epsilon) \equiv i$$

(isto é, ϵ deve ser elemento neutro de θ).
Passemos agora a (6). Repare-se antes de mais que, sendo ϵ elemento neutro de θ, a definição de $BinOp$ tem a estrutura de f_φ dada em (2.80). Assim, tem-se:

$$BinOp(x) = I.\Theta_{a \in x} a$$

desde que θ seja associativa e comutativa:

$$\begin{array}{rcl} \theta(i, \theta(j,k)) & \equiv & \theta(\theta(i,j),k) \\ \theta(i,j) & \equiv & \theta(j,i) \end{array}$$

Passemos agora ao facto a provar,

$$MOD(BinOp(union(x,y))) = MOD(binOp(BinOp(x), BinOp(y)))$$

que se reduz a:

$$I.\Theta_{a \in x \cup y} = I.\theta(I.\Theta_{a \in x}, I.\Theta_{a \in y})$$

Ora é conhecido que este facto só se verifica se $I.\theta$ fôr idempotente, isto é, se

$$I.\theta(i,i) \equiv i$$

Em suma, há 4 axiomas a impôr a I:

$$\begin{array}{rcl} \theta(i, \epsilon) & \equiv & i \\ \theta(i, \theta(j,k)) & \equiv & \theta(\theta(i,j),k) \\ \theta(i,j) & \equiv & \theta(j,i) \\ \theta(i,i) & \equiv & i \end{array}$$

□

Questão 6 Na sequência da questão 5, mostre que o seguinte isomorfismo de J-álgerbras não se verifica,

$$MOD(\mathcal{T}_I) \cong \mathcal{T}_J \qquad (7)$$

onde $\mathcal{T}_I$ e $\mathcal{T}_J$ designam respectivamente os modelos triviais das interfaces I e J.

RESOLUÇÃO: Vejamos o que acontece em relação à espécie As:

$$
\begin{array}{rcl}
(MOD(\mathcal{T}_I))(As) & \cong & \mathcal{T}_J(As) \\
& \updownarrow & \\
2^{\mathcal{T}_I(Item)} & \cong & 1 \\
& \updownarrow & \\
2^1 & \cong & 1 \\
& \updownarrow & \\
2 & \cong & 1 \\
& \updownarrow & \\
& F &
\end{array}
$$

Logo o isomorfismo não se verifica nesta espécie, o que é suficiente para mostrar que $MOD(\mathcal{T}_I)$ e $\mathcal{T}_J$ não são J-álgebras isomorfas. $\square$

Questão 7 Considere uma interface axiomática $I = \langle \Sigma, X, E \rangle$ em que Σ admite os operadores

$$
\begin{array}{rcl}
+ & : & nat \times nat \to nat \\
1 & : & \to nat
\end{array}
$$

e E inclui o axioma

$$
x + y \quad \equiv \quad y + x
$$

Sejam $t = 1 + x$ e $t' = x + 1$ dois $\Sigma(X)$-termos. Indique, justificando formalmente, quais das seguintes afirmações estão correctas:

– Os alcances $[t]$ e $[t']$ são conjuntos idênticos. (8)

– A intersecção de $[t]$ com $[t']$ é o conjunto vazio de Σ-termos. (9)

– $[t] \cap [t']$ é um conjunto singular. (10)

RESOLUÇÃO: Partindo da definição de *alcance de t*,

$$
[t] = \{\mathcal{W}^\rho(t) \mid \ \rho \in V_\mathcal{W}\}
$$

(Definição 3.5), teremos, no nosso caso:

$$
\begin{array}{rcl}
[1 + x] & = & \{\mathcal{W}^\rho(1 + x) \mid \ \rho = \begin{pmatrix} x \\ t \end{pmatrix} \wedge t \in W_{\Sigma, nat}\} \\
& = & \{1 + 1, 1 + (1 + 1), 1 + (1 + (1 + 1)), \ldots\}
\end{array}
$$

e

$$
\begin{array}{rcl}
[x + 1] & = & \{\mathcal{W}^{\rho'}(x + 1) \mid \ \rho' = \begin{pmatrix} x \\ t' \end{pmatrix} \wedge t' \in W_{\Sigma, nat}\} \\
& = & \{1 + 1, (1 + 1) + 1, ((1 + 1) + 1) + 1, \ldots\}
\end{array}
$$

Para validarmos as afirmações basta calcularmos $[1 + x] \cap [x + 1]$:

$$
[1 + x] \cap [x + 1] \quad = \quad \{1 + t \mid \ t \in W_{\Sigma, nat} \wedge \exists t' \in W_{\Sigma, nat} : 1 + t = t' + 1\}
$$

O sinal "=" na expressão anterior é o de igualdade sintática, a congruência da álgebra $\mathcal{W}$; a este nível o axioma dado é irrelevante (só o seria se a congruência fosse $\cong_E$). Assim, por exemplo, $(1 + 1) + 1 \neq 1 + (1 + 1)$ e só para $t = t' = 1$ é que a igualdade se verifica, isto é:

$$
[1 + x] \cap [x + 1] \quad = \quad \{1 + 1\}
$$

correspondendo a $\rho(x) = \rho'(x) = 1$.

Em suma, só a última afirmação é que é verdadeira. $\square$

Questão 8 Verifique se o modelo semântico definido no exercício 3.13 (pág. 145 da sebenta) como semântica de uma pequena linguagem para Lógica Temporal satisfaz os seguintes axiomas:

$$\odot(f\ \mathtt{and}\ f') \equiv (\odot f)\ \mathtt{and}\ (\odot f') \tag{11}$$

$$\mathtt{not}\ (\bigcirc f) \equiv \bigcirc(\ \mathtt{not}\ f) \tag{12}$$

RESOLUÇÃO: No primeiro caso, basta mostrar que $[\![\odot(f\ \mathtt{and}\ f')]\!]$ é o mesmo significado que $[\![(\odot f)\ \mathtt{and}\ (\odot f')]\!]$. Teremos

$$\begin{array}{rcl} [\![\odot(f\ \mathtt{and}\ f')]\!] & = & \lambda i.\left\{\begin{array}{lcl} i=0 & \Rightarrow & F \\ i>0 & \Rightarrow & [\![f\ \mathtt{and}\ f']\!](i-1) \end{array}\right. \\ & = & \lambda i.\left\{\begin{array}{lcl} i=0 & \Rightarrow & F \\ i>0 & \Rightarrow & [\![f]\!](i-1) \wedge [\![f']\!](i-1) \end{array}\right. \end{array}$$

quanto ao lado esquerdo, e

$$\begin{array}{rcl} [\![(\odot f)\ \mathtt{and}\ (\odot f')]\!] & = & \lambda i.[\![\odot f]\!](i) \wedge [\![\odot f']\!](i) \\ & = & \lambda i.(\left\{\begin{array}{lcl} i=0 & \Rightarrow & F \\ i>0 & \Rightarrow & [\![f]\!](i-1) \end{array}\right. \wedge \left\{\begin{array}{lcl} i=0 & \Rightarrow & F \\ i>0 & \Rightarrow & [\![f']\!](i-1) \end{array}\right. \\ & = & \lambda i.\left\{\begin{array}{lcl} i=0 & \Rightarrow & F \wedge F \\ i>0 & \Rightarrow & [\![f]\!](i-1) \wedge [\![f']\!](i-1) \end{array}\right. \\ & = & \lambda i.\left\{\begin{array}{lcl} i=0 & \Rightarrow & F \\ i>0 & \Rightarrow & [\![f]\!](i-1) \wedge [\![f']\!](i-1) \end{array}\right. \end{array}$$

quanto a lado direito. A igualdade é evidente.

Quanto a segundo axioma o processo resume-se a

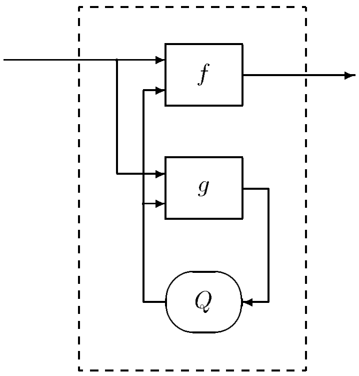
$$\begin{array}{rcl} [\![\ \mathtt{not}\ (\bigcirc f)]\!] & = & \lambda i.\neg([\![\bigcirc f]\!](i)) \\ & = & \lambda i.\neg([\![f]\!](i+1)) \end{array}$$

e

$$\begin{array}{rcl} [\![\bigcirc(\ \mathtt{not}\ f)]\!] & = & \lambda i.[\![\ \mathtt{not}\ f]\!](i+1) \\ & = & \lambda i.(\lambda j.\neg([\![f]\!](j)))(i+1) \\ & = & \lambda i.\neg([\![f]\!](i+1)) \end{array}$$

chegando-se também a uma igualdade. □

Questão 9 O diagrama seguinte representa a estrutura bi-funcional de uma dada acção (evento) determinística E, com apenas um argumento e uma saída, num modelo com estado interno Q:



Dadas as seguintes definições de f e g,

$$\begin{array}{rcll} f & = & \lambda(\sigma, a).\ let & n = H(a) \\ & & \quad in & \left\{ \begin{array}{lcl} n \in dom(\sigma) & \Rightarrow & a \in \sigma(n) \\ n \notin dom(\sigma) & \Rightarrow & F \end{array} \right. \\ g & = & \lambda(\sigma, a).\ let & n = H(a) \\ & & \quad in & \sigma \dagger \left(\begin{array}{c} n \\ \{a\} \cup \left\{ \begin{array}{lcl} n \in dom(\sigma) & \Rightarrow & \sigma(n) \\ n \notin dom(\sigma) & \Rightarrow & \emptyset \end{array} \right. \end{array} \right) \end{array}$$

onde $H : A \to \mathbb{N}$ é uma função pré-definida,

1. complete a seguinte definição da assinatura do evento,

$$E : \ \ldots\ldots\ldots\ldots \ \to \ \ldots\ldots\ldots\ldots$$

   e do modelo formal de Q:

$$Q \cong \ \ldots\ldots\ldots\ldots$$

2. descreva por palavras suas o significado deste evento, relacionando-o com um modelo de gestão da informação bem conhecido na literatura algorítmica;
3. mostre que E satisfaz o seguinte invariante:

$$\text{inv-}Q(\sigma) \stackrel{\text{def}}{=} \forall n \in dom(\sigma) : (\emptyset \neq \sigma(n) \wedge \forall i \in \sigma(n) : H(i) = n)$$

   i.é, prove o facto seguinte, para a e σ universalmente quantificadas:

$$\text{inv-}Q(\sigma) \wedge \sigma' = g(\sigma, a) \quad \Rightarrow \quad \text{inv-}Q(\sigma')$$

   Sugestão: desdobre o facto a provar em duas quantificações separadas e prove-as independentemente.

RESOLUÇÃO:

1. Designaremos por α e β os espaços a completar em E, isto é, temos:

$$\begin{array}{rcl} E & : & \alpha \to \beta \\ E(a, r') & \stackrel{\text{def}}{=} & \sigma' = g(\sigma, a) \wedge r' = f(\sigma, a) \end{array}$$

   tendo-se, obrigatoriamente,

$$\begin{array}{rcl} f & : & Q \times \alpha \to \beta \\ g & : & Q \times \alpha \to Q \end{array}$$

   Da definição de f infere-se, de imediato, $\beta = 2$ (Booleanos). Infere-se também que Q terá de ter a estrutura de uma função finita cujo domínio é do mesmo tipo que o contra-domínio de H, isto é, $\mathbb{N}$; ao aplicar H a a ficamos a saber que α é A; infere-se ainda que $\sigma(n)$ é um conjunto a que a pode pertencer; sendo a do tipo A, ter-se-á, em suma:

$$E : A \to 2$$

   e

$$Q \cong \mathbb{N} \hookrightarrow 2^A$$

   A definição de g confirma esta tipificação.

2. Abstratamente, trata-se de uma operação de arquivo de $a \in A$ numa família de conjuntos de A indexada por $\mathbb{N}$, sendo o índice do conjunto a albergar a identificado pelo cálculo de $H(a)$.

   Podemos pensar numa infinidade de instâncias deste objecto computacional, por exemplo, A um tipo de dados $Aluno$, $H(a)$ o cálculo da média final de curso de a, e Q a estrutura de uma base de dados que arquiva alunos em classes de equivalência indexadas por nota final. Mas todas essas instâncias caem na classe de uma conhecida estrutura computacional, a *tabela de 'hashing'*, onde H é a função de 'hashing' (relembrar figura 1.8 da sebenta e respectivo exercício).

3. O facto a provar simplifica em

$$\text{inv-}Q(\sigma) \quad \Rightarrow \quad \text{inv-}Q(g(\sigma, a))$$

Por (A.5), inv-$Q(\sigma)$ desdobra-se em duas quantificações,

$$\text{inv-}Q_1(\sigma) \wedge \text{inv-}Q_2(\sigma)$$

onde:

$$\begin{array}{lcl} \text{inv-}Q_1(\sigma) & \stackrel{\text{def}}{=} & \forall n \in dom(\sigma) : \emptyset \neq \sigma(n) \\ \text{inv-}Q_2(\sigma) & \stackrel{\text{def}}{=} & \forall n \in dom(\sigma) : (\forall i \in \sigma(n) : H(i) = n) \end{array}$$

O facto a provar pode ser demonstrado através de duas provas mais simples,

$$\begin{array}{rcll} \text{inv-}Q_1(\sigma) & \Rightarrow & \text{inv-}Q_1(g(\sigma, a)) & (a) \\ \text{inv-}Q_2(\sigma) & \Rightarrow & \text{inv-}Q_2(g(\sigma, a)) & (b) \\ \hline \underbrace{\text{inv-}Q_1(\sigma) \wedge \text{inv-}Q_2(\sigma)}_{\text{inv-}Q(\sigma)} & \Rightarrow & \underbrace{\text{inv-}Q_1(g(\sigma, a)) \wedge \text{inv-}Q_2(g(\sigma, a))}_{\text{inv-}Q(g(\sigma,a))} & \end{array}$$

Prova de (a):

$$\begin{array}{lcl} \text{inv-}Q_1(\sigma) & \Rightarrow & \text{inv-}Q_1(\sigma \dagger \begin{pmatrix} H(a) \\ \{a\} \cup \ldots \end{pmatrix}) \\ & \updownarrow & \\ \text{inv-}Q_1(\sigma) & \Rightarrow & \text{inv-}Q_1(\sigma \setminus \{H(a)\}) \wedge (\emptyset \neq \{a\} \cup \ldots) \\ & \updownarrow & \\ \text{inv-}Q_1(\sigma) & \Rightarrow & \emptyset \neq \{a\} \cup \ldots \\ & \updownarrow & \\ \text{inv-}Q_1(\sigma) & \Rightarrow & V \\ & \updownarrow & \\ & V & \end{array}$$

(**NB**: deixa-se ao leitor a tarefa de justificar os passos dados).
Prova de (b) — segue o mesmo esquema da anterior:

$$\begin{array}{lcl} \text{inv-}Q_2(\sigma) & \Rightarrow & \text{inv-}Q_2(\sigma \dagger \begin{pmatrix} H(a) \\ \{a\} \cup \ldots \end{pmatrix}) \\ & \updownarrow & \\ \text{inv-}Q_2(\sigma) & \Rightarrow & \text{inv-}Q_2(\sigma \setminus \{H(a)\}) \wedge (\forall i \in \{a\} \cup \ldots : H(i) = H(a)) \\ & \updownarrow & \\ \text{inv-}Q_2(\sigma) & \Rightarrow & \forall i \in \{a\} \cup \ldots : H(i) = H(a) \\ & \updownarrow & \\ \text{inv-}Q_2(\sigma) & \Rightarrow & H(a) = H(a) \wedge \forall i \in \left\{ \begin{array}{lcl} H(a) \in dom(\sigma) & \Rightarrow & \sigma(H(a)) \\ H(a) \notin dom(\sigma) & \Rightarrow & \emptyset \end{array} \right. : H(i) = H(a) \\ & \updownarrow & \\ \text{inv-}Q_2(\sigma) & \Rightarrow & \left\{ \begin{array}{lcl} H(a) \in dom(\sigma) & \Rightarrow & \forall i \in \sigma(H(a)) : H(i) = H(a) \\ H(a) \notin dom(\sigma) & \Rightarrow & V \end{array} \right. \\ & \updownarrow & \\ & V & \end{array}$$

(**NB**: deixa-se ao leitor a tarefa de justificar os passos dados).

□