Métodos Formais de Programação I +
Opção I - Métodos Formais de Programação I

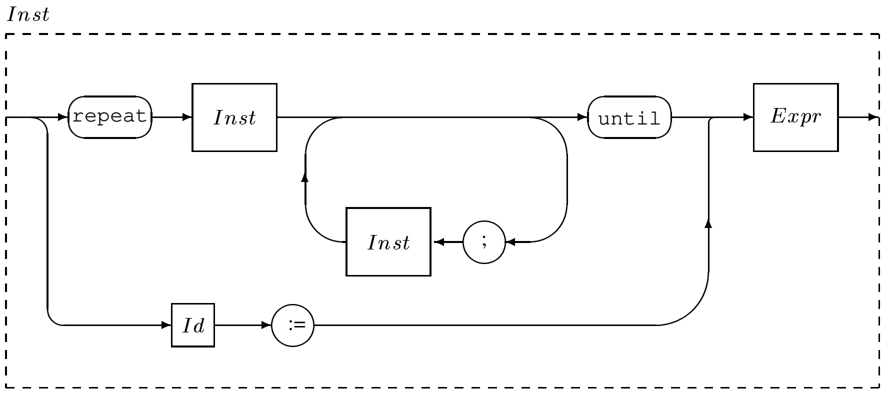
4.º Ano da LMCC (7007N2) + LESI (5307P2)
Ano Lectivo de 1999/2000

Exame (1.ª chamada) — 25 de Janeiro de 2000
09h30
Salas 2303 e 2304

NB: *Esta prova consta de **10** alíneas que valem, cada uma, 2 valores. Para sua consulta, encontra anexa a esta prova a listagem das principais leis de cálculo estudadas na disciplina.*

PROVA SEM CONSULTA (3 $horas$)

Questão 1 Construa a assinatura algébrica Σ correspondente ao seguinte fragmento da gramática da linguagem de programação PASCAL, expresso sobre a forma do grafo de sintaxe



onde $Inst$ é o não-terminal para *instruções*, Exp é o não-terminal para *expressões* e Id é o não-terminal para *identificadores*, isto é, correspondente ao BNF

$$\begin{array}{lcl} \langle \mathit{Inst} \rangle & ::= & \langle \mathit{Id} \rangle \ \texttt{:=} \ \langle \mathit{Expr} \rangle \mid \ \texttt{repeat} \ \langle \mathit{Inst} \rangle \langle \mathit{Insts} \rangle \ \texttt{until} \ \langle \mathit{Expr} \rangle \\ \langle \mathit{Insts} \rangle & ::= & \epsilon \mid \ \texttt{;} \ \langle \mathit{Inst} \rangle \langle \mathit{Insts} \rangle \end{array}$$

Questão 2 Indique qual das leis dadas em anexo é justificada pelo raciocínio que se segue,

$$\begin{array}{cl} & (i \times j) \cdot \langle g, h \rangle \\ = & \qquad \{ \ldots \text{(justifique)} \ldots \} \\ & \langle i \cdot \pi_1, j \cdot \pi_2 \rangle \cdot \langle g, h \rangle \\ = & \qquad \{ \ldots \text{(justifique)} \ldots \} \\ & \langle (i \cdot \pi_1) \cdot \langle g, h \rangle, (j \cdot \pi_2) \cdot \langle g, h \rangle \rangle \end{array}$$

$$
\begin{array}{cl}
= & \{\ \ldots \text{(justifique)} \ldots\ \} \\
 & \langle i \cdot (\pi_1 \cdot \langle g, h\rangle), j \cdot (\pi_2 \cdot \langle g, h\rangle)\rangle \\
= & \{\ \ldots \text{(justifique)} \ldots\ \} \\
 & \langle i \cdot g, j \cdot h\rangle
\end{array}
$$

Justifique cada passo.

Questão 3 Deduza a lei da fusão do coproduto (14) a partir da respectiva propriedade universal (11) e de outras que constam do anexo desta prova.

Questão 4 É dada a função seguinte, expressa em notação CAMILA:

```
f(x) = let (n = p1(x), m = p2(x) )
       in  if (n == m) then i1(NIL) else i2(<n, <succ(n),m> >);
```

1. Explique o propósito do anamorfismo cujo "gene" é a função dada. Construa o respectivo diagrama.
2. Mostre que $[\![f]\!]$ pode estar indefinida (*i.é*, nem sempre termina). Redefina f por forma a resolver esse problema.

Questão 5 Uma tabela relacional pode ser modelada por uma lista de tuplos, sendo cada tuplo encarado como uma lista de pares atributo/valor. Podem existir atributos omissos, por exemplo o atributo C no segundo tuplo desta tabela,

A	B	C
$a1$	$b1$	$c1$
$a1$	$b2$	
$a2$	$b1$	$c3$
$a2$	$b1$	$c1$
$a1$		$c1$
$a3$		$c1$
	$b1$	$c2$

(1)

que é modelada pela seguinte lista de listas, em notação CAMILA:

```
< < < "A","a1">,< "B","b1">,< "C","c1"> >,
  < < "A","a1">,< "B","b2"> >,
  < < "A","a2">,< "B","b1">,< "C","c3"> >,
  < < "A","a2">,< "B","b1">,< "C","c1"> >,
  < < "A","a1">,< "C","c1"> >,
  < < "A","a3">,< "C","c1"> >,
  < < "B","b1">,< "C","c2"> >,
  < >
>
```

1. Especifique em notação com variáveis as seguintes duas funções:

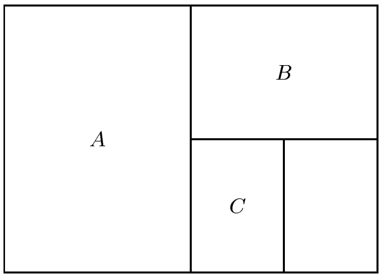
   $atributos(t)$ — deverá retornar o conjunto de todos os atributos que ocorrem em pelo menos um tuplo da tabela;

   $valores(a, t)$ — deverá retornar o conjunto de todos os valores que ocorrem na coluna a da tabela t, onde a é o identificador do respectivo atributo;

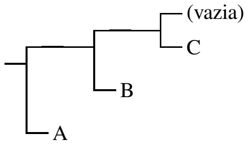
2. Será possível exprimir $atributos$ e $valores$ com catamorfismos de listas? Justifique cuidadosamente a sua resposta.
   Sugestão: transponha $valores$ (*cf.* 'currying') e concentre-se em $(\overline{valores})a$.

Questão 6 Pretende-se que o transporte de embalagens num supermercado se faça em paletes previamente arrumadas por um robot. Este deverá usar um algoritmo de arrumação por partição binária da base da palete ($100cm \times 141cm$). Para optimizar a arrumação, tanto as embalagens como as paletes são normalizadas, isto é, as suas bases são rectângulos cujo comprimento é $\sqrt{2}$ maior que a largura (Trata-se de um requisito que, além de útil para a partição binária do espaço, permite identificar as dimensões de uma embalagem apenas pela largura. Para simplificar, não nos procuparemos com a altura de embalagens e paletes.)

A figura que se segue mostra a arrumação de três caixas A, B e C numa palete, estando ainda disponível um espaço de arrumação que poderá receber uma embalagem com as dimensões de C ou ser dividido em dois sub-espaços para arrumar duas embalagens mais pequenas e assim sucessivamente:



ou seja, temos a seguinte "árvore de arrumação":



A partição binária divide sempre um espaço disponível ao meio, segundo o comprimento.

Suponha que, para especificar este problema, se definiram já as seguintes estruturas de informação:

$$Space \cong 1 + Box + Space^2$$ */* um espaço na palete ou está vazio, ou contém uma caixa ou está dividido em 2 sub-espaços */*

$$Box \cong Id \times Width$$ */* identificação da embalagem e sua largura de base */*

$$Width \cong I\!N_0$$

1. Mostre que a definição de $Space$ é polinomialmente recursiva (indique os coeficientes do polinómio) e construa o diagrama genérico dos seus catamorfismos.

2. Considere a seguinte especificação de $Space$ em notação CAMILA:

```
TYPE
      Space = [ S ];
      S     = Box | BoxS;
      Box   = I:STR W:INT;
      BoxS  = F:Space S:Space;
ENDTYPE
```

   Infira a álgebra $in_{\texttt{Space}}$ e a sua inversa $out_{\texttt{Space}}$ e use-as para especificar o operador

   $$whichBoxes \quad : \quad Space \rightarrow Ids$$ */* deverá dar como resultado todos os identificadores das embalagens já colocadas na palete */*

   sob a forma de um catamorfismo sobre `Space`.

3. Parametrize $Space$ em relação a Box, isto é defina $\mathsf{T}\ A$ tal que $Space = \mathsf{T}(Box)$ e defina o respectivo functor de tipo $\mathsf{T}\ f$. Use-o para definir uma função que duplique o tamanho das caixas de uma palete.

Anexo–Cálculo de Funções

COMPOSIÇÃO

Natural-id	$f \cdot id = id \cdot f = f$	(2)
Associatividade	$(f \cdot g) \cdot h \quad = \quad f \cdot (g \cdot h)$	(3)

PRODUTO

Universal-$\times$	$k = \langle f, g \rangle \quad \Leftrightarrow \quad \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases}$	(4)
Cancelamento-$\times$	$\pi_1 \cdot \langle f, g \rangle = f \quad , \quad \pi_2 \cdot \langle f, g \rangle = g$	(5)
Reflexão-$\times$	$\langle \pi_1, \pi_2 \rangle \quad = \quad id_{A \times B}$	(6)
Fusão-$\times$	$\langle g, h \rangle \cdot f = \langle g \cdot f, h \cdot f \rangle$	(7)
Absorção-$\times$	$(i \times j) \cdot \langle g, h \rangle = \langle i \cdot g, j \cdot h \rangle$	(8)
Functor-$\times$	$(g \cdot h) \times (i \cdot j) \quad = \quad (g \times i) \cdot (h \times j)$	(9)
Functor-id-$\times$	$id_A \times id_B \quad = \quad id_{A \times B}$	(10)

COPRODUTO

Universal-$+$	$k = [f, g] \quad \Leftrightarrow \quad \begin{cases} k \cdot i_1 = f \\ k \cdot i_2 = g \end{cases}$	(11)
Cancelamento-$+$	$[g, h] \cdot i_1 = g \quad , \quad [g, h] \cdot i_2 = h$	(12)
Reflexão-$+$	$[i_1, i_2] \quad = \quad id_{A+B}$	(13)
Fusão-$+$	$f \cdot [g, h] = [f \cdot g, f \cdot h]$	(14)
Absorção-$+$	$[g, h] \cdot (i + j) = [g \cdot i, h \cdot j]$	(15)
Functor-$+$	$(g \cdot h) + (i \cdot j) = (g + i) \cdot (h + j)$	(16)
Functor-id-$+$	$id_A + id_B = id_{A+B}$	(17)

EXPONENCIAÇÃO

Universal	$k = \overline{f} \quad \Leftrightarrow \quad f = ap \cdot (k \times id)$	(18)
Cancelamento	$f = ap \cdot (\overline{f} \times id)$	(19)
Reflexão	$\overline{ap} = id_{B^A}$	(20)
Fusão	$\overline{g \cdot (f \times id)} = \overline{g} \cdot f$	(21)
Absorção	$\overline{f \cdot g} = f^A \cdot \overline{g}$	(22)
Functor	$(g \cdot h)^A = g^A \cdot h^A$	(23)
Functor-id	$id^A = id$	(24)

INDUÇÃO

Universal-cata	$k = (\!\lvert \alpha \rvert\!) \quad \Leftrightarrow \quad k \cdot in = \alpha \cdot \mathsf{F}\, k$	(25)
Cancelamento-cata	$(\!\lvert \alpha \rvert\!) \cdot in = \alpha \cdot \mathsf{F}\, (\!\lvert \alpha \rvert\!)$	(26)
Reflexão-cata	$(\!\lvert in \rvert\!) = id_{\mu \mathsf{F}}$	(27)
Fusão-cata	$f \cdot (\!\lvert \alpha \rvert\!) = (\!\lvert \beta \rvert\!) \quad \text{if} \quad f \cdot \alpha = \beta \cdot \mathsf{F}\, f$	(28)

MISC.

Lei da troca	$[\langle f, g \rangle, \langle h, k \rangle] \quad = \quad \langle [f, h], [g, k] \rangle$	(29)