

Métodos de Programação III

LESI + LMCC (3º ano)

Época de Recurso

Data: 22 de Fevereiro de 2003
Hora: 09:30

Dispõe de 2:30 horas para realizar este exame

1 Questão: Definição de ERs

Considere as expressões regulares, p e q , abaixo e as linguagens por elas definidas, e responda às seguintes alíneas:

$$p = a^* (a + c b^*) (d + b + c) (\epsilon + b^*)$$

$$q = (a + c b^*) a^* ((d b^* + b^+ + c b^*) + d + b + c)$$

- Reduza a expressão q à forma mais simples.
- Mostre que $'aad' \in \mathcal{L}_p$, apresentando o processo de derivação sucessiva.
- Explique da melhor maneira que, apesar de $'aad' \in \mathcal{L}_q$, as duas linguagens \mathcal{L}_p e \mathcal{L}_q não são iguais.

2 Questão: Modelação com ERs

Um endereço de email identifica universalmente um utilizador de uma determinado domínio. Escreva uma expressão regular que define sintaticamente um endereço de email válido. Dê exemplos de frases válidas e inválidas, de acordo com a ER que definiu.

3 Questão: ERs, Sistemas de Produção e Lex

Analise o Sistema de Produção que especifica, na sintaxe do Lex, um determinado filtro de texto:

```
%%  
a { printf("hah"); }  
[a-zA-Z]+ ;  
[a-zA-Z][a-z]+/"mente" { mente++; }  
^[0-9]+ { s += atoi( yytext ); }  
[. : ; ? ! ]+/[ ]+[A-Z] { ECHO; sinal++; }
```

Responda, então, às questões seguintes:

- Explique, esclarecendo a sua resposta, qual o resultado de aplicar o filtro gerado pela especificação acima a um dado texto.
- Supondo que a terceira regra deve ser apenas aplicada, em exclusividade, após surgir no texto a indicação `#agora#` e enquanto não aparecer a indicação `#chega#`, adapte a especificação acima.

4 Questão: Modelação com Autómatos

Desenhe um autómato reactivo que modele a máquina de pagamento do parque do CPIII. No seu estado inicial, a máquina começa por aceitar um bilhete. Depois aceita o respectivo pagamento, que pode ser um cartão multibanco, seguido do código, e seguido de um sinal da SIBS a indicar se o pagamento é válido ou não. O pagamento (assuma que é sempre da importância fixa de 0,40 Euros) também pode ser efectuado em moedas de 10, 20 ou 50 cêntimos. No caso da importância exceder os 0,40 Euros, terá que ser dado troco ao utente. A qualquer momento, o utente pode carregar na tecla 'cancelar', e todo o processo é interrompido, sendo devolvido o dinheiro eventualmente já introduzido.

Se o pagamento for válido, então a máquina tem que perguntar se o utente quer ou não recibo. Se quiser recibo, imprime o recibo.

Em todos os casos, no fim, ejecta o cartão de estacionamento.

5 Questão: ERs, ANDs e ADs

Considere as expressões regulares abaixo,

$$p = (a^* + b^*) (a + b) c (c + b^+)$$

$$r = c (a b^* + \epsilon) c$$

e responda às seguintes alíneas:

- Desenhe (informalmente) o autómato determinista que define a linguagem \mathcal{L}_p .
- Converta a ER r num autómato não-determinista, mostrando cada passo formal dos processos de conversão.
- Escreva a expressão regular correspondente ao seguinte autómato determinista, \mathcal{A}_1 definido abaixo em Haskell; sistematize o processo de conversão (se possível, apresente regras que sistematizem a conversão em causa):
 $\mathcal{A}_1 = (\mathcal{V}, \mathcal{Q}, \mathcal{S}, \mathcal{Z}, \delta)$, com $\mathcal{V} = \{a, b, c\}$, $\mathcal{Q} = \{1, 2, 3, 4, 5\}$, $\mathcal{S} = 1$, $\mathcal{Z} = \{4\}$ e

$$\delta 1 a = 2$$

$$\delta 1 b = 3$$

$$\delta 1 c = 5$$

$$\delta 2 c = 4$$

$$\delta 3 a = 2$$

$$\delta 3 b = 4$$

$$\delta 4 c = 5$$

$$\delta 5 a = 2$$

$$\delta 5 b = 4$$

- Inverta o autómato determinista \mathcal{A}_1 e volte a converter para determinista o autómato não-determinista obtido após a inversão.

6 Questão: Gramáticas e Árvores

Considere a árvore de derivação abstracta, ADA, obtida ao reconhecer a frase $a, 9; b, 2; d, 4$ de acordo com uma gramática concreta G cujo axioma é o símbolo Desc, a qual está representada a seguir em Haskell, de acordo com os tipos e construtores que G induz (tal como sugerido nas aulas):

```
ADA = pares (par (letra 'a')(num 9))
        (lista (par (letra 'b')(num 2))
              (lista (par (letra 'd')(num 4))
                    (vazia)))
```

- Escreva uma possível gramática G .

- b) Justifique porque é que na alínea anterior se pediu *uma possível gramática* e não *a única gramática possível*.
- c) Supondo que o construtor `pares` aplicado aos seus dois argumentos devolve um valor do tipo `Desc`, mostre qual seria uma possível definição desse tipo em `Haskell`, sendo coerente com a resposta dada à alínea a).

7 Gramáticas

Analise atentamente a seguinte GIC, cujo Símbolo Inicial é `FCalculo`, em que todos os símbolos escritos em minúsculas são pseudo-terminais, as palavras-reservadas estão escritas em maiúsculas, os sinais estão entre apóstrofos e `&` representa a string nula.

```

FCalculo : &
          | Celula FCalculo
Celula: Id Conteudo '[' Formato ']'
Id: '(' Coord '-' Coord ')'
Conteudo: int
          | real
          | data
          | '"' string '"'
          | '=' Exp
Exp: Term OpAdd Exp
    | Term
Term: Factor OpMul Term
     | Factor
Factor: '(' Exp ')'
        | int
        | real
        | OpUn '(' Id ')'
        | OpLst '(' Id ':' Id ')'
OpUn: TRUNC | ABS | SQRT | LOG | SIN | COS | TAN | ATAN | EXP
OpLst: SUM | MEDIA | MAIOR | MENOR | COUNT
OpAdd: '+' | '-'
OpMul: '*' | '/'
Coord : int
Formato: &
         | Prop ';' Formato
Prop: ITALICO | BOLD | LIMITES | DIREITA | ESQUERDA

```

Print screen do Excel:

- a) Escreva as expressões regulares, na sintaxe do `Lex`, que definem os pseudo-terminais: `real`, `data` e `string`.
- b) Escreva uma frase da linguagem que use a produção
`Formato-> Prop ';' Formato`
- c) Desenhe a árvore de derivação, a começar no axioma, que permite representar o conteúdo da segunda linha do Excel (só a segunda linha!). Note que a célula A2 está formatada a **bold**.
- d) Escreva a função de um parser recursivo descendente para reconhecer o símbolo `Formato`.