

Métodos de Programação III

LESI + LMCC (3º ano)

Exame de Época Especial

Data: 09 de Setembro de 2003
Hora: 17:00

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, \mathcal{L}_p e \mathcal{L}_q :

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

$$q = (a^+ + b^+)$$

$$r = a + (a^+ b^+) + b^+$$

e responda às seguintes alíneas:

- É um facto que as frases de \mathcal{L}_p e \mathcal{L}_q tem no mínimo um símbolo e começam sempre por um **a** ou um **b**. Explique, então, qual a diferença entre essas linguagens.
- Diga, justificando, se as ERs p e r são equivalentes, isto é, se geram exactamente a mesma linguagem.
- Mostre que *'aaa'* não pertence à linguagem \mathcal{L}_p , apresentando o processo de derivação sucessiva.

2 Questão: ERs, Sistemas de Produção e Lex

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

```
%{
int pal=0; s=0;
}%
%%
[a-zA-Z][a-zA-Z]+      { pal++; }
[a-zA-Z]               { printf("%c",toupper(yytext[0])); }
[0-9]+                 { s += atoi( yytext ); }
.                       { printf("."); }
%%
int main()
{ yylex();
  printf("%d \n %d \n", s, pal);
  return(0);
}
```

Responda, então, às questões seguintes:

- a) Mostre, esclarecendo a sua resposta, qual o resultado de aplicar o filtro gerado pela especificação acima ao texto seguinte:

Era uma vez um Rei, nascido em Agosto de 1310, que tinha sido baptizado em Janeiro de 1311 por 2 Fadas: 1 Boa e outra Má. O Rei ia a ALMEDINA,...

- b) Supondo que a terceira regra deve ser apenas aplicada quando o número for seguido de uma vírgula ou ponto, altere-a adequadamente.

3 Questão: Modelação com ERs

Numa versão simplista, um endereço de email é formado por 2 partes separadas pelo símbolo "@"". A primeira parte identifica o destinatário e é formada por uma só palavra (permite apenas o uso do "-"); a segunda parte corresponde ao domínio e é formada no mínimo por duas palavras separadas por um ponto. Assume-se que palavra é uma sequência de letras minúsculas.

Escolhendo os símbolos apropriados, defina a sintaxe de um endereço, através de uma Expressão Regular.

4 Questão: Modelação com Autómatos

Desenhe um Autómato Determinista que modele o comportamento do controlador de um sistema de iluminação de jardim.

O sistema tem dois modos de funcionamento: manual e automático. Se manual, espera que seja accionado o botão de **on**, permanecendo ligado até que seja accionado o botão de **off**, que o repõe no estado inicial (de selecção de modo). Se automático, pode ser controlado pelos sensores de luz, ou pelo timer. Quando na opção sensores, aguarda pelo sinal de baixa luminosidade e então acende as lâmpadas, só as desligando quando recebe o sinal de claridade; nesse caso volta ao estado anterior pois não muda sozinho o modo de funcionamento. Na opção de timer, espera que seja indicada a hora de ligação e o número de horas em que permanece ligado; após esses dados, acende as lâmpadas quando recebe o sinal de disparo do timer e volta ao estado anterior (lâmpadas apagadas) quando volta a receber o sinal do timer indicado que atingiu o tempo estabelecido. Como se disse a mudança para a outra opção automática ou para o modo manual terá de ser introduzida pelo operador no estado inicial ou no estado de espera de sinal para acender.

5 Questão: ERs, ANDs e ADs

Considere a expressão regular abaixo,

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

e responda às seguintes alíneas:

- a) Desenhe (informalmente) o autómato determinista que define a linguagem \mathcal{L}_p .
- b) Converta a ER p num autómato não-determinista, mostrando cada passo formal dos processos de conversão e faça a sua declaração em Haskell de acordo com a definição do tipo `Ndfa` apresentado nas aulas.
- c) Calcule o *ε*fecho do estado inicial do AND da alínea anterior.
- d) Escreva a expressão regular correspondente ao seguinte autómato determinista, \mathcal{A}_1 definido abaixo em Haskell:
 $\mathcal{A}_1 = (\mathcal{V}, \mathcal{Q}, \mathcal{S}, \mathcal{Z}, \delta)$, com $\mathcal{V} = \{a, b, c\}$, $\mathcal{Q} = \{1, 2, 3, 4, 5\}$, $\mathcal{S} = 1$, $\mathcal{Z} = \{4\}$ e

$\delta 1 a = 2$
 $\delta 2 b = 3$
 $\delta 2 a = 5$
 $\delta 2 c = 4$
 $\delta 3 a = 2$
 $\delta 3 c = 4$
 $\delta 4 a = 2$
 $\delta 5 a = 2$

6 Questão: escrita de uma gramática

Pretende-se desenvolver uma linguagem para descrever cursos de uma universidade. Para cada curso, a linguagem permitirá indicar o seu código e nome, grau (licenciatura, especialização, ou mestrado), o departamento que o dirige, o conjunto de áreas científicas (cada uma designada por uma sigla), os créditos e a lista das disciplinas (um ou mais). Opcionalmente, pode indicar-se para cada disciplina (além do nome que é obrigatório) o semestre (1 a 10) e/ou a escolaridade (número de horas teóricas e horas práticas).

Responda, então, às seguintes alíneas:

- Escreva uma gramática independente de contexto (GIC) que defina a linguagem pretendida.
- Dê um exemplo de uma frase válida da sua linguagem, apresentando a respectiva árvore de derivação.

7 Questão: Gramáticas, Linguagens e Parsing

Analise atentamente a seguinte GIC, G , cujo Símbolo Inicial é `Opers`, 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.

```

Opers  --> Op  '.'
Opers  --> Op  ';' Opers
Op      --> SOMA  Ns
Op      --> MULTIPLICA  Ns
Ns      --> num Resto
Resto   --> Ns
Resto   --> &

```

Responda, então, às alíneas seguintes:

- Mostre que

`SOMA 1 2; MULTIPLICA 3 4 .`

é uma frase válida da linguagem gerada por G , apresentando a respectiva árvore de derivação.

- Escreva as funções de um parser recursivo descendente para reconhecer os símbolos `Op` e `Resto`.
- Escreva em `Haskell`, usando a notação e os tipos apresentados nas aulas, a Gramática Abstracta equivalente a G .
- Admitindo que se pretende usar a linguagem `Haskell` para implementar as funções de reconhecimento e processamento das frases da linguagem gerada pela gramática acima, suponha que já existem funções para fazer o parsing dos símbolos `num` e `Resto` que retornam como resultado, respectivamente, um número inteiro e uma lista de inteiros. Escreva, então em `Haskell` e usando *combinadores de parsing*, uma função para reconhecer o símbolo `Ns` e retornar uma nova lista de inteiros.