

LISTA DE EXERCÍCIOS

A. Expressões regulares e autômatos finitos

1) Apresente o conjunto das cadeias de caracteres (com pelo menos 5 elementos, quando houver) que casam com as expressões regulares abaixo:

- a) a
- b) a*
- c) a+
- d) (a|b|c)+
- e) (ab)*
- f) (0|1)+
- g) ab*(c|ε)
- h) (ab)*(cd)*

2) A partir das cadeias de caracteres abaixo, apresente expressões regulares que as definem:

- a) {a, b, ab, abba, aaab, baaa, ...}
- b) {abc, aabc, aaabc, aaaabc, ...}
- c) {1, 2, 20, 1234, 56789, 803459, ...}
- d) {if, else, do, while}
- e) {casa, zap, media, peso, IRPF, ...}
- f) {nota01, p1, xyz, xyz123, cod2f1, ...}
- g) números reais

3) Considerando o alfabeto $\Sigma = \{a, b, c, d, \dots, y, z\}$, apresente uma expressão regular que aceite todas as cadeias de caracteres que:

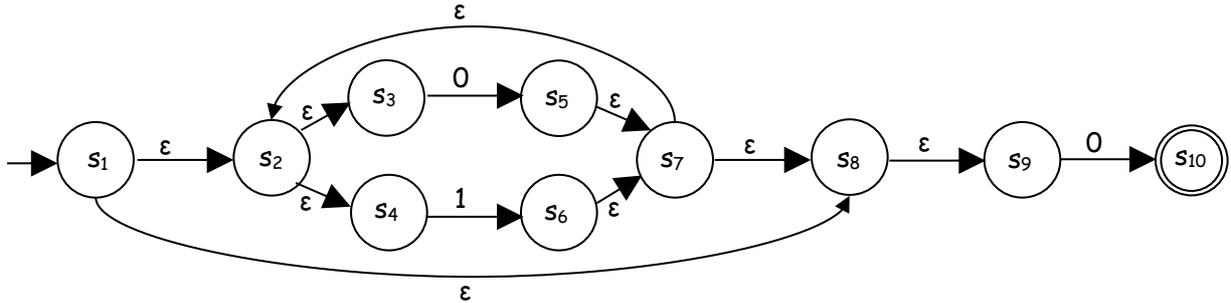
- a) contenham exatamente um caractere z.
- b) contenham no máximo um caractere z.

4) Considerando o alfabeto $\Sigma = \{a, b, c\}$, apresente uma expressão regular que aceite todas as cadeias que não contenham dois caracteres b consecutivos.

5) Escreva expressões regulares para os conjuntos de cadeias abaixo:

- a) Cadeias de letras minúsculas que começam e terminam com a.
- b) Cadeias de letras minúsculas que começam ou terminam com a.
- c) Cadeias de dígitos sem zeros à esquerda.
- d) Cadeias de dígitos que representam números pares.
- e) Cadeias de dígitos onde todos os 2 ocorrem imediatamente antes de um 9.
- f) Cadeias com caracteres a e b com pelo menos três b consecutivos.
- g) Cadeias com caracteres a e b que contêm exatamente dois b.
- h) Cadeias com caracteres a e b que terminam com aa ou bb.
- i) Cadeias com caracteres a e b que não possuem dois a consecutivos.

6) Considere o AFND abaixo:



- Escreva a expressão regular das cadeias reconhecidas por este AFND.
- Converta este AFND em um AFD equivalente.
- Minimize o AFD obtido.

7) Use o algoritmo de Thompson para converter a expressão regular $(a|b)^*a(a|b)|\epsilon$ em um AFND. Em seguida, encontre um AFD equivalente, e depois minimize este AFD.

B. Gramáticas livres de contexto

8) Considere a gramática:

$$A \rightarrow AA \mid (A) \mid \epsilon$$

- Descreva a linguagem gerada.
- Mostre que esta gramática é ambígua.

9) Considere a gramática:

$$\begin{aligned} \text{exp} &\rightarrow \text{exp soma termo} \mid \text{termo} \\ \text{soma} &\rightarrow + \mid - \\ \text{termo} &\rightarrow \text{termo mult fator} \mid \text{fator} \\ \text{mult} &\rightarrow * \\ \text{fator} &\rightarrow (\text{exp}) \mid \text{número} \end{aligned}$$

Escreva derivações à esquerda e árvores sintáticas para as sentenças abaixo:

- $3+4*5-6$
- $3*(4-5+6)$
- $3-(4+5*6)$

10) Escreva uma gramática para expressões *booleanas* contendo as constantes T e F, e os operadores lógicos \wedge , \vee , \neg e $()$. O operador \vee deve ter precedência menor que \wedge , que por sua vez deve ter precedência menor que \neg . A repetição de operadores \neg é permitida (exemplo: $\neg\neg T$), e a gramática não pode ser ambígua.

C. Análise sintática descendente

11) Considerando $T = \{\text{OUTRA, IDENT, "(", ")", ":="}\}$, escreva um *parser* recursivo para a gramática abaixo:

decl \rightarrow atrib-decl | ativ-decl | OUTRA
atrib-decl \rightarrow IDENT := exp
ativ-decl \rightarrow IDENT (exp lista)

12) Considere a gramática abaixo, onde $T = \{\text{NUM, IDENT, "(", ")", ";"}\}$:

lexp \rightarrow atomo | lista
atomo \rightarrow NUM | IDENT
lista \rightarrow (lexp-seq)
lexp-seq \rightarrow lexp-seq ; lexp | lexp

- Encontre uma gramática LL(1) equivalente.
- Construa os conjuntos *First* e *Follow* para os não-terminais da gramática equivalente.
- Construa a tabela preditiva para a gramática equivalente.
- Dada a entrada "(a ; (b ; (2)) ; (c))", simule as ações do *parser* tabular LL(1).

13) Considere a gramática a seguir, onde $T = \{\text{"int", "float", IDENT, ",", "}"\}$, para declarações simplificadas em C:

decl \rightarrow tipo var-lista
tipo \rightarrow int | float
var-lista \rightarrow IDENT , var-lista | IDENT

- Encontre uma gramática LL(1) equivalente.
- Construa os conjuntos *First* e *Follow* para os não-terminais da gramática equivalente.
- Construa a tabela preditiva para a gramática equivalente.
- Dada a entrada "int x, y, z", simule as ações do *parser* tabular LL(1).

D. Análise sintática ascendente

14) Considere a gramática a seguir, onde $T = \{a, "(", ")", ";\}$:

$$E \rightarrow (L) \mid a$$

$$L \rightarrow L ; E \mid E$$

- Construa o AFD de itens LR(0) para esta gramática.
- Construa a tabela de análise sintática SLR(1).
- Mostre a pilha de análise sintática e as ações de um analisador SLR(1) para a cadeia de entrada “(a); a; (a; a)”.

15) Considere a gramática a seguir, onde $T = \{“(", “)”\}$:

$$S \rightarrow S (S) \mid \varepsilon$$

- Construa o AFD de itens LR(0) para esta gramática.
- Construa a tabela de análise sintática SLR(1).
- Mostre a pilha de análise sintática e as ações de um analisador SLR(1) para a cadeia de entrada “(())”.