

Exercícios de LFA.

Alunos: Andrey Baumhardt Ramos e Raphael Machado dos Santos

A) CONCEITOS

1 – **R:** A Hierarquia de Chomsky dividiu as gramáticas formais em 4 classes diferentes:

- 1- **Tipo 0:** Também conhecidas como irrestrita, este tipo de gramática não possui nenhuma regra de produção específica, são capazes de reconhecer linguagens recursivamente enumeráveis.
- 2- **Tipo 1:** Também conhecidas como gramáticas sensíveis ao contexto, são as gramáticas onde todas as regras de produção são da forma: $\alpha \rightarrow \beta$ tal que $|\beta| \geq |\alpha|$ onde tanto α quanto β podem ser um conjunto de terminais com não-terminais.
- 3- **Tipo 2:** Também conhecidas como gramáticas livre de contexto, são as gramáticas onde todas as regras de produção são da forma: $A \rightarrow \beta$ onde β é uma união entre terminais e não-terminais e A é apenas um símbolo não-terminal.
- 4- **Tipo 3:** Também conhecido como Gramática Regular ou

Linear, é a mais utilizada para construção de compiladores pois fornece uma maneira fácil de obter reconhecedores simples, possui somente um símbolo não-terminal do lado esquerdo. É subdividida em quatro categorias:

- a. **Gramática Linear a Direita (GLD):** regras de produção na forma $A \rightarrow wB$ ou $A \rightarrow w$, sendo A e B símbolos não-terminais e w um símbolo terminal, ou seja, cresce para a direita;
- b. **Gramática Linear a Esquerda (GLE):** regras de produção na forma $A \rightarrow Bw$ ou $A \rightarrow w$, sendo A e B símbolos não-terminais e w um símbolo terminal, ou seja, cresce para a esquerda;
- c. **Gramática Linear Unitária a Direita (GLUD):** igual a GLD porém não pode possuir mais de um símbolo terminal e não terminal do lado direito;
- d. **Gramática Linear Unitária a Direita (GLUE):** igual a GLE porém não pode possuir mais de um símbolo terminal e não terminal do lado direito;

2 – **R:** Gramáticas do Tipo 2 e Tipo 3, as gramáticas do Tipo 3 fornecem um meio eficiente e simples de reconhecer padrões, é a base para construção de um analisador léxico, já as gramáticas do tipo 2 tem extrema importância pois especificam adequadamente a estrutura sintática da linguagem de programação, sem elas não seria possível, por exemplo, reconhecer o fim de um while ou de um if. As linguagens de programação são reconhecidas por autômatos de pilha e geradas por gramáticas livre de contexto, a maioria das linguagens de programação são GLC.

3 - **R:** Gramática é um conjunto finito de regras as quais, quando aplicadas sucessivamente, geram palavras. Linguagem é o conjunto de todas as palavras geradas por uma gramática.

4 – **R**: Autômato Finito: Trata-se de um formalismo operacional ou reconhecedor, sendo um sistema de estados finitos;

Expressão Regular: Trata-se de um formalismo denotacional ou gerador, é definido a partir dos conjuntos de linguagens básicos e das operações de concatenação e união, infere na construção de todas as palavras da linguagem;

Gramática Regular: Trata-se do formalismo axiomático ou também considerado gerador, é uma gramática mas com restrições da forma das regras de produção.

5 – **R**: Trata-se de um formalismo denotacional, definem como as palavras da linguagem podem ser geradas, é definida a partir de conjuntos básico e operações, as principais operações são:

União → A expressão $(r + s)$ é a E.R que denota a linguagem $R \cup S$

Concatenação → A expressão (rs) é a E.R que denota a linguagem $RS = \{uv \mid u \in R \text{ e } v \in S\}$

Concatenação Sucessiva → A expressão (r^*) é a E.R que denota a linguagem gerada por R^* , ou seja, todas as palavras sobre R.

6 – **R**: $G=(V,T,P,S)$ tal que $V=\{A,B,C\}$, $T=\{a,b,c\}$, $S=\{S\}$

$P = \{ \quad S \rightarrow aS \mid aB \mid \varepsilon ;$

$\quad B \rightarrow bB \mid bC \mid \varepsilon ;$

$\quad C \rightarrow cC \mid \varepsilon ; \}$

Essa gramática é do tipo 2, tendo somente não-terminais do lado esquerdo e tanto terminais como não-terminais do lado direito.

B) GRAMÁTICAS E EXPRESSÕES REGULARES

$G1 = \{V, T, P, S\}$

$V = \{A\}$

$T = \{a, b, c\}$

$S = \{A\}$

$P = \{$

$\quad A \rightarrow aA$

$\quad A \rightarrow bA$

$\quad A \rightarrow cA$

$\quad A \rightarrow \varepsilon$

$\}$

$ER1 = (a|b|c)^*$

$G2 = \{V, T, P, S\}$

$V = \{A\}$

$T = \{a, b, c\}$

$S = \{A\}$

$P = \{$

A → aA

A → bA

A → cA

A → a

A → b

A → c

}

ER2 = (a|b|c)*(a|b|c)

G3 = {V, T, P, S}

V = {A, B}

T = {a, b, c}

S = {A}

P = {

A → BBB

B → a

B → b

B → c

}

ER3 = (a|b|c)(a|b|c)(a|b|c)

G4 = {V, T, P, S}

V = {A, B, C}

T = {a, b, c}

S = {A}

P = {

A → ε

A → B

A → BB

A → BBBBC

B → a

B → b

B → c

C → BC

C → ε

}

ER4 = ε|a|b|c| (a|b|c)(a|b|c)|(a|b|c)(a|b|c) (a|b|c)(a|b|c)(a|b|c)*

G5 = {V, T, P, S}

V = {A, B}

T = {a, b, c}

S = {A}

P = {

A → BBBA

B → a

B → b

B → c

A → ε

}

ER5 = ((a|b|c)(a|b|c)(a|b|c))*

G6 = {V, T, P, S}

V = {A, B}

T = {a, b, c}

S = {A}

P = {

A → BBA

B → a

B → b

B → c

A → ε

}

ER6 = ((a|b|c)(a|b|c))*

G7 = {V, T, P, S}

V = {A, B}

T = {a, b, c}

S = {A}

P = {

A → abbB

B → aB

B → bB

B → cB

A → ε

}

ER7 = abb(a|b|c)*

G8 = {V, T, P, S}

V = {A, B}

T = {a, b, c}

S = {A}

P = {

A → a

A → abB

A → bB

A → cB

B → aB

B → bB

B → cB

B → ε

$$\}$$

$$ER8 = a(ab|b|c)(a|b|c)^*$$

$$G9 = \{V, T, P, S\}$$

$$V = \{A, B\}$$

$$T = \{a, b, c\}$$

$$S = \{A\}$$

$$P = \{$$

$$A \rightarrow Bbbb$$

$$B \rightarrow aB$$

$$B \rightarrow bB$$

$$B \rightarrow cB$$

$$B \rightarrow \varepsilon$$

$$\}$$

$$ER9 = (a|b|c)^*bbb$$

$$G10 = \{V, T, P, S\}$$

$$V = \{A, B\}$$

$$T = \{a, b, c\}$$

$$S = \{A\}$$

$$P = \{$$

$$A \rightarrow aBc$$

$$B \rightarrow aB$$

$$B \rightarrow bB$$

$$B \rightarrow cB$$

$$B \rightarrow \varepsilon$$

$$\}$$

$$ER10 = a(a|b|c)^*c$$

$$G11 = \{V, T, P, S\}$$

$$V = \{A, B, C\}$$

$$T = \{a, b, c\}$$

$$S = \{A\}$$

$$P = \{$$

$$A \rightarrow BaBaBaBCBCB$$

$$B \rightarrow bB$$

$$B \rightarrow cB$$

$$B \rightarrow \varepsilon$$

$$B \rightarrow a$$

$$C \rightarrow \varepsilon$$

$$\}$$

$$ER11 = (b|c)^*a(b|c)^*a(b|c)^*a(b|c)^*(a|\varepsilon)(b|c)^*(a|\varepsilon)(b|c)^*$$

$$G12 = \{V, T, P, S\}$$

$V = \{A, B\}$
 $T = \{a, b, c\}$
 $S = \{A\}$
 $P = \{$
 $\quad A \rightarrow aBb$
 $\quad A \rightarrow aBc$
 $\quad A \rightarrow bBa$
 $\quad A \rightarrow bBc$
 $\quad A \rightarrow cBa$
 $\quad A \rightarrow cBb$
 $\quad B \rightarrow aB$
 $\quad B \rightarrow bB$
 $\quad B \rightarrow bB$
 $\quad B \rightarrow \varepsilon$
 $\}$

ER12 = $a(a|b|c)^*b| a(a|b|c)^*c| b(a|b|c)^*a| b(a|b|c)^*c| c(a|b|c)^*a| c(a|b|c)^*b$

G13 = $\{V, T, P, S\}$

$V = \{A, B\}$
 $T = \{a, b, c\}$
 $S = \{A\}$
 $P = \{$
 $\quad A \rightarrow BabcbB$
 $\quad B \rightarrow aB$
 $\quad B \rightarrow bB$
 $\quad B \rightarrow cB$
 $\quad B \rightarrow \varepsilon$
 $\}$

ER13 = $(a|b|c)^*abc(a|b|c)^*$

G15 = $\{V, T, P, S\}$

$V = \{A, B, C\}$
 $T = \{-, +, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
 $S = \{A\}$
 $P = \{$
 $\quad A \rightarrow -B$
 $\quad A \rightarrow +B$
 $\quad B \rightarrow 1C|2C|3C|4C|5C|6C|7C|8C|9C$
 $\quad C \rightarrow 0C|1C|2C|3C|4C|5C|6C|7C|8C|9C$
 $\quad C \rightarrow \varepsilon$
 $\}$

ER15 = $(-|+)^+ (0|1|2|3|4|5|6|7|8|9)^*$

C) IMPLEMENTAÇÃO

1- R:

```
import java.util.*;
import java.lang.*;
import java.io.*;

class Ideone
{
    public static void main (String[] args) throws java.lang.Exception
    {
        String frase = "Teste";
        String sufix = "| ";
        String prefix = "| ";
        int i,n,s;
        n = 4;
        s = frase.length();
        System.out.printf("A string é: "+ frase + "\nO tamanho da string é: " + s + "\nEla irá ser concatenada: " + n + " vezes\n");
        for (i=0;i<n;i++){
            frase = frase + "Teste";
        }
        System.out.printf(frase);
        for(i=0;i<s;i++){
            sufix = sufix + frase.substring(0,i+1) + " | ";
        }
        System.out.printf("\nO conjunto de sufixos é: " + sufix);
        for(i=s;i>0;i--){
            prefix = prefix + frase.substring(i-1,s) + " | ";
        }
        System.out.printf("\nO conjunto de prefixos é: " + prefix);
        System.out.printf("\nO conjunto de sub-palavras é: " + prefix+sufix);
    }
}
```