



Universidade Estadual do Paraná – Campus Apucarana

Curso: Ciência da Computação

Disciplina: Linguagens Formais, Autômatos e Computabilidade

Professor: Guilherme Henrique de Souza Nakahata

Aluno: João Vitor de Souza Ribeiro **Turma:** 2º Ano



DOCUMENTAÇÃO II – LFA

1. INTRODUÇÃO

De acordo com o requisitado para a entrega do trabalho prático do terceiro bimestre da disciplina de Linguagens Formais, Autômatos e Computabilidade, faz-se necessário especificar o método de uso e os principais pontos relevantes para que tal seja, de fato, efetivo e possa se ter o maior aproveitamento do código fonte contido na entrega. Dessa forma, no presente documento serão estipuladas as orientações de uso e como proceder em cada caso de teste, cabendo ao docente entender como obrigatório ou não tal ação.

1. ESTRUTURA E ORIENTAÇÕES

1.1 RECEBIMENTO

Para que possamos especificar o funcionamento do código e as orientações a serem seguidas para compilá-lo da maneira mais eficiente, precisamos iniciar pelo recebimento dos dados iniciais, a descrição formal e as transições.

1.1.1 Descrição Formal

A descrição formal de uma Gramática Linear Unitária é composta pelos conjuntos de variáveis e terminais e um símbolo de partida. Cada um destes, acompanhado da escolha entre GLUD e GLUE, é recebido na primeira parte do código; quando visto no console, a Figura 1 exemplifica um caso de recebimento dos dados para a formação da descrição formal. Para receber a escolha do usuário, no entanto, deve-se digitar 1 para GLUD e 2 para GLUE, caso nenhum desses valores sejam identificados, o programa fica em loop até que 1 ou 2 seja de fato fornecido, como demonstra a Figura 2.

```

----- Gramática Regular LFA -----
Digite 1 para GLUD e 2 para GLUE: 2
Digite a quantidade de variáveis (V): 3
Digite a 1ª variável: S
Digite a 2ª variável: A
Digite a 3ª variável: B
Digite a quantidade de alfabeto da linguagem (T) - terminal: 2
Digite o 1º caractere do alfabeto: a
Digite o 2º caractere do alfabeto: b
Digite o símbolo de início: S

```

Figura 1

```

----- Gramática Regular LFA -----
Digite 1 para GLUD e 2 para GLUE: 3
Digite 1 para GLUD e 2 para GLUE: 6
Digite 1 para GLUD e 2 para GLUE: 9
Digite 1 para GLUD e 2 para GLUE: -2
Digite 1 para GLUD e 2 para GLUE: 0
Digite 1 para GLUD e 2 para GLUE: 1
Digite a quantidade de variáveis (V): 3
Digite a 1ª variável: s
Digite a 2ª variável: a
Digite a 3ª variável: b

```

Figura 2

1.1.2 Ordens de Produção

Além do recebimento da descrição formal e da escolha, precisamos receber do usuário as ordens de produção que compõem a Gramática Linear Unitária, seja ela à esquerda ou à direita. Para cada tipo de escolha imprimimos uma mensagem orientando como o usuário deve fornecer cada ordem de produção, evitando falhas no momento da execução. As Figuras 3 e 4 mostram como essas mensagens são impressas no caso de GLUD e GLUE, respectivamente.

```

-----
--> Digite no padrão A>bC
--> A e C pertencentes à V
--> b pertencente à T
--> para simbolizar brancos utilize *
-----

```

Figura 3

```

-----
--> Digite no padrão A>Bc
--> A e B pertencentes à V
--> c pertencente à T
--> para simbolizar brancos utilize *
-----

```

Figura 4

Para de fato recebermos as ordens de produção, deve-se inserir as ordens no formato informado, além de que, se existir a necessidade de

representar o vazio, deve-se utilizar o caractere “*”. Nas Figuras 5 e 6 o recebimento dessas informações estão exemplificados, seguido da representação visual da descrição formal e das ordens de produção recebidas, nas versões de GLUD e GLUE, respectivamente.

```
--> Digite no padrão A>bC
--> A e C pertencentes à V
--> b pertencente à T
--> para simbolizar brancos utilize *
-----
Digite a 1ª ordem de produção:
S>aA
Digite a 2ª ordem de produção:
A>aA
Digite a 3ª ordem de produção:
A>bB
Digite a 4ª ordem de produção:
A>**
Digite a 5ª ordem de produção:
B>bB
Digite a 6ª ordem de produção:
B>**
----- Descrição Formal -----
V = {S,A,B}
T = {a,b}
S = S
P = {S>aA
      A>aA
      A>bB
      A>**
      B>bB
      B>**}
```

Figura 5

```
--> Digite no padrão A>Bc
--> A e B pertencentes à V
--> c pertencente à T
--> para simbolizar brancos utilize *
-----
Digite a 1ª ordem de produção:
S>Bb
Digite a 2ª ordem de produção:
S>Aa
Digite a 3ª ordem de produção:
A>Aa
Digite a 4ª ordem de produção:
A>**
Digite a 5ª ordem de produção:
B>Bb
Digite a 6ª ordem de produção:
B>Aa
----- Descrição Formal -----
V = {S,A,B}
T = {a,b}
S = S
P = {S>Bb
      S>Aa
      A>Aa
      A>**
      B>Bb
      B>Aa}
```

Figura 6

1.1.3 Palavra teste

Após, é necessário receber uma palavra de teste, para verificar se esta pode ser aceita ou não pela Gramática. Uma String recebe a palavra que vai ser testada, de modo com que, após a execução, possa verificar se a palavra gerada pelas ordens de produção resultou na mesma palavra que foi fornecida pelo usuário. A Figura 7 representa o recebimento da palavra que é recebida.

```
Digite a palavra: aaab
```

Figura 7

Após o recebimento dessa palavra, o programa é executado, analisando as ordens de produção e verificando se a palavra pode ou não ser aceita, no caso de ser efetivamente aceita na Gramática Linear Unitária, uma mensagem é exibida informando que “A PALAVRA FOI ACEITA” e as ordens de produção que foram passadas pelo algoritmo, unidas a formação da palavra que está sendo gerado. No caso contrário, quando a palavra não foi aceita, a mensagem “A PALAVRA NÃO FOI ACEITA” é emitida ao usuário. As Figuras 8 e 9 exemplificam como as duas opções podem ser mostradas, respectivamente.

```
Digite a palavra: aaab
----- A PALAVRA FOI ACEITA -----
..... Ordens de produção passadas pelo algoritmo .....
<1> S>aA
aA
<2> A>aA
aaA
<2> A>aA
aaaA
<3> A>bB
aaabB
<6> B>**
aaab
.....
```

Figura 8

```
Digite a palavra: ba
----- A PALAVRA NÃO FOI ACEITA -----
```

Figura 9

Após apresentar uma destas mensagens, seguida ou não das ordens de produção, o programa pergunta ao usuário se deseja testar outra palavra na mesma linguagem, assim, duas opções são exibidas ao usuário, 1 para testar uma nova palavra ou 2 para não testar e encerrar o programa. No primeiro caso recebemos uma nova palavra do usuário e testamos ela passando pela

execução e verificação. Do contrário, apenas é exibida uma mensagem agradecendo e o programa se encerra. Nas Figuras 10 e 11 podemos ver quando uma nova palavra é testada, resultando em uma palavra aceita e em uma recusada, respectivamente. Já na Figura 12, quando o usuário deseja encerrar o programa.

```
.....
Deseja testar outra palavra? 1 - SIM | 2 - NÃO
1
Digite a palavra: a
----- A PALAVRA FOI ACEITA -----
..... Ordens de produção passadas pelo algoritmo .....
<2> S>Aa
Aa
<4> A>**
a
```

Figura 10

```
.....
Deseja testar outra palavra? 1 - SIM | 2 - NÃO
1
Digite a palavra: aaabc
----- A PALAVRA NÃO FOI ACEITA -----|
```

Figura 11

```
.....
Deseja testar outra palavra? 1 - SIM | 2 - NÃO
2
..... Obrigado .....
```

Figura 12

De todo modo, estes são os principais pontos para observação no código fonte, não carecendo de muitas orientações, a não ser aquelas expressas na própria execução, visualizadas através do console.