



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

1. INTRODUÇÃO

De acordo com o requisitado para a entrega do trabalho prático do segundo 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 é formada pelo conjunto de estados, conjunto de letras do alfabeto principal, conjunto de símbolos do alfabeto auxiliar, marcador de início, símbolo de branco, estado inicial e conjunto de estados finais. Cada um destes 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. Na Figura 2, por outro lado, é representada a descrição formal impressa em um padrão semelhante ao que usamos em sala.

```

===== MÁQUINA DE TURING =====
Digite a quantidade de letras do alfabeto: 2
Informe a letra 1 do alfabeto: 1
Informe a letra 2 do alfabeto: 0
Informe uma letra para representar os estados: S
Digite a quantidade total de estados: 4
Digite o estado inicial - somente o nº -: 0
Digite a quantidade de estados finais: 1
Digite o 1º estado final - somente o nº -: 3
Digite o marcador de início: <
Digite o marcador de branco: *

```

Figura 1

```

***** Descrição Formal *****
Σ = {1,0}
E = {S0,S1,S2,S3}
i = S0
F = {S3}
γ = {<,*}
< = <
β = *

```

Figura 2

1.1.2 Transições

Além do recebimento da descrição, precisamos receber do usuário as transições que farão parte da máquina de Turing, assim, imprimimos uma tabela padronizada, facilitando ao usuário localizar onde cada inserção deve ir, a Figura 3 explicita isso.

```

***** Tabela de Transição *****

```

δ	1	0	<	*
S0	1,1	1,2	1,3	1,4
S1	2,1	2,2	2,3	2,4
S2	3,1	3,2	3,3	3,4
S3	4,1	4,2	4,3	4,4

Figura 3

Para de fato inserir as transições, deve-se inserir, primeiro, o estado futuro – em um padrão completo, S0, por exemplo –, após, o alfabeto futuro e, por fim, a direção que deve seguir – somente são aceitas duas direções D e E. Para os casos em que o campo deve ser anulado, o usuário pode colocar x ou X em qualquer um dos campos de recebimento, os três campos daquela posição serão anulados. Nas Figuras 4 e 5 o recebimento dessas informações estão exemplificados, seguido de uma nova tabela de transições, agora preenchida com as informações recém coletadas.

```

***** Transições *****
Obs: Se não existir uma transição, insira x
Obs: Uma transição inválida fará com que o campo seja anulado
----- Digite as transições -----
-----
Digite o estado futuro da transição {1,1}: S0
Digite o alfabeto futuro da transição {1,1}: 1
Digite a direção da transição {1,1}: D
-----
Digite o estado futuro da transição {1,2}: S0
Digite o alfabeto futuro da transição {1,2}: 0
Digite a direção da transição {1,2}: D
-----
Digite o estado futuro da transição {1,3}: x
O campo foi anulado!
-----
Digite o estado futuro da transição {1,4}: S1
Digite o alfabeto futuro da transição {1,4}: 0
Digite a direção da transição {1,4}: D
-----
Digite o estado futuro da transição {2,1}: x
O campo foi anulado!
-----
Digite o estado futuro da transição {2,2}: x
O campo foi anulado!
-----
Digite o estado futuro da transição {2,3}: x
O campo foi anulado!
-----
Digite o estado futuro da transição {2,4}: S2
Digite o alfabeto futuro da transição {2,4}: 0
Digite a direção da transição {2,4}: E
-----
Digite o estado futuro da transição {3,1}: S2
Digite o alfabeto futuro da transição {3,1}: 1
Digite a direção da transição {3,1}: E

```

Figura 4

```

-----
Digite o estado futuro da transição {3,2}: S2
Digite o alfabeto futuro da transição {3,2}: 0
Digite a direção da transição {3,2}: E
-----
Digite o estado futuro da transição {3,3}: S3
Digite o alfabeto futuro da transição {3,3}: <
Digite a direção da transição {3,3}: D
-----
Digite o estado futuro da transição {3,4}: x
O campo foi anulado!
-----
Digite o estado futuro da transição {4,1}: x
O campo foi anulado!
-----
Digite o estado futuro da transição {4,2}: x
O campo foi anulado!
-----
Digite o estado futuro da transição {4,3}: x
O campo foi anulado!
-----
Digite o estado futuro da transição {4,4}: x
O campo foi anulado!
***** Tabela de Transição *****
δ    1      0      <      *
S0   S0,1,D  S0,0,D  Sx,x,x  S1,0,D
S1   Sx,x,x  Sx,x,x  Sx,x,x  S2,0,E
S2   S2,1,E  S2,0,E  S3,<,D  Sx,x,x
S3   Sx,x,x  Sx,x,x  Sx,x,x  Sx,x,x
*****

```

Figura 5

1.1.3 Palavra teste

Após, é necessário receber uma palavra de teste, para verificar se esta pertence ou não a linguagem da MT. Um vetor *vet[]* receberá a String da palavra digitada. Ao receber a palavra, é impresso a fita inicial, sendo o vetor preenchido de (marcador de início + palavra + brancos), como demonstra a Figura 6.

```
*****
Digite uma palavra para ser testada: 100
***** FITA INICIAL *****
< 1 0 0 * * * * *
*****
```

Figura 6

Logo após a impressão desta fita inicial, o programa retorna a fita modificada (fita final) e a decisão a respeito da aceitação desta palavra mediante a linguagem analisada. Se a palavra for aceita será impresso “A palavra foi aceita!”, caso contrário, “A palavra não foi aceita!” será exibido no console. A Figura 7 mostra algumas interações de conferência para palavras numa Máquina de Turing transdutora que deve multiplicar o número binário de entrada por 4 (quádruplo do número binário).

```
*****
Digite uma palavra para ser testada: 100
***** FITA INICIAL *****
< 1 0 0 * * * * *
*****
***** FITA FINAL *****
< 1 0 0 0 0 * * * * *
*****
A palavra foi aceita!
Deseja testar outra palavra? 1 - SIM | 2 - NÃO
Opção Escolhida: 1
Digite uma palavra para ser testada: 001
***** FITA INICIAL *****
< 0 0 1 * * * * *
*****
***** FITA FINAL *****
< 0 0 1 0 0 * * * * *
*****
A palavra foi aceita!
Deseja testar outra palavra? 1 - SIM | 2 - NÃO
Opção Escolhida: 1
Digite uma palavra para ser testada: 111
***** FITA INICIAL *****
< 1 1 1 * * * * *
*****
***** FITA FINAL *****
< 1 1 1 0 0 * * * * *
*****
A palavra foi aceita!
Deseja testar outra palavra? 1 - SIM | 2 - NÃO
Opção Escolhida: 2
=====
===== OBRIGADO POR UTILIZAR A MÁQUINA DE TURING =====
=====
```

Figura 7

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.