

SIMULADOR DE AUTÔMATO FINITO DETERMINÍSTICO

Sumário

1. INTRODUÇÃO.....	1
2. DEFINIÇÃO DE AUTÔMATO FINITO DETERMINÍSTICO	3
2.1. FUNÇÃO PROGRAMA ESTENDIDA	4
3. SIMULADOR DE AUTÔMATO FINITO DETERMINÍSTICO	6
3.1. INSERINDO/REMOVENDO SÍMBOLOS.....	7
3.2. INSERINDO/REMOVENDO ESTADOS.....	8
3.3. DEFININDO O CONJUNTO DOS ESTADOS TERMINAIS	10
3.4. DEFININDO O ESTADO INICIAL	11
3.5. DEFININDO A FUNÇÃO DE TRANSIÇÃO	11
3.6. PROCESSANDO UMA PALAVRA DE ENTRADA.....	12
4. BIBLIOGRAFIA CONSULTADA	14

1. INTRODUÇÃO

Um Autômato Finito Determinístico (AFD) ou simplesmente Autômato Finito é um modelo extremamente restrito de computador. Ele pode ser visto como uma máquina composta, basicamente, de três partes:

- Fita.* Dispositivo de entrada que contém a informação a ser processada;
- Unidade de Controle.* Reflete o estado corrente da máquina. Possui uma unidade de leitura (cabeça da fita) a qual acessa uma célula da fita de cada vez e movimenta-se exclusivamente para a direita;
- Programa ou Função de Transição.* Função que comanda as leituras e define o estado da máquina.

Uma representação de Autômato Finito pode ser vista na figura abaixo:

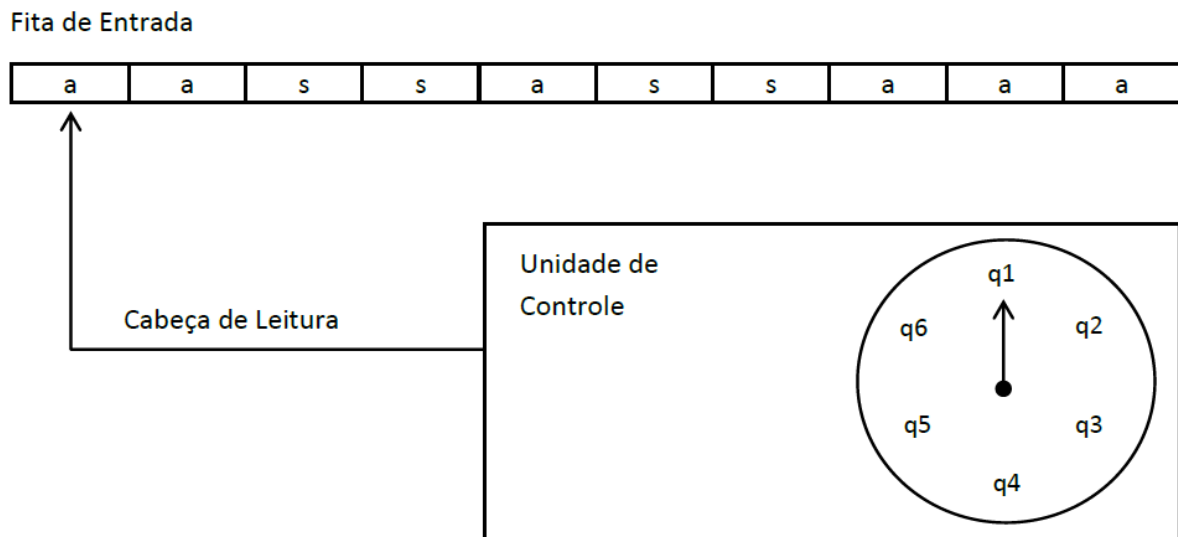


Figura 1.1. Modelo de um Autômato Finito Determinístico.

A fita é finita (à esquerda e à direita), sendo dividida em células, onde cada uma armazena um símbolo. Os símbolos pertencem a um alfabeto de entrada. Não é possível gravar sobre a fita.

Inicialmente, a cabeça de leitura está posicionada na célula mais à esquerda da fita. A intervalos regulares, o autômato lê um símbolo e entra num novo estado, que dependerá do estado corrente e do símbolo lido. O autômato para quando a cabeça de leitura chega ao fim da fita, e se o estado a que ele foi levado ao terminar é um estado final ou terminal, a entrada é considerada aceita, senão ela é rejeitada.

Um Autômato Finito é, em outras palavras, um dispositivo de reconhecimento de linguagem. O que o torna um modelo tão restrito dos computadores reais é a completa ausência de memória externa à sua Unidade de Controle (na verdade, ele não é realmente isento de memória, ele simplesmente tem uma capacidade de memória que é fixada “em sua fabricação” a qual não pode mais ser expandida).

Vários tipos de algoritmos e programas de computador são projetados com base em Autômatos Finitos. A fase de análise léxica de um compilador (durante a qual são identificados os elementos básicos de um programa, tais como “begin” e “+”) é um desses exemplos. Outro exemplo é o de localizar uma ocorrência de uma cadeia dentro

de outra – por exemplo, se quiser saber se as cadeias *autômato*, *estado* aparecem ou não neste texto.

Em complemento a esta introdução informal, no **capítulo 2**, teremos a definição matemática de Autômato Finito. Veremos também como uma Função Programa Estendida descreve o passo a passo do processamento de um Autômato para uma palavra de entrada w .

No **capítulo 3**, teremos a descrição do funcionamento do Simulador de Autômato Finito Determinístico. Com este simulador, pode-se criar qualquer Autômato Finito e fazer o processamento de qualquer palavra de entrada para este Autômato criado. O resultado do processamento do Autômato para a palavra de entrada w é exibido em forma de Função Programa Estendida, conforme será estudado no capítulo 2.1.

2. DEFINIÇÃO DE AUTÔMATO FINITO DETERMINÍSTICO

Um Autômato Finito Determinístico **M** é uma 5-upla:

$$M = \{\Sigma, Q, \delta, q_0, F\}$$

onde:

- Σ alfabeto de símbolos de entrada;
- Q** conjunto de estados possíveis do autômato o qual é finito;
- δ função programa ou função de transição:
 $\delta: Q \times \Sigma \rightarrow Q$
 a qual é uma função parcial;
- q₀** estado inicial tal que q_0 é elemento de **Q**;
- F** conjunto de estados finais ou terminais, tal que **F** está contido em **Q**.

A função de transição pode ser representada como um grafo finito direto, como ilustrado na figura abaixo:

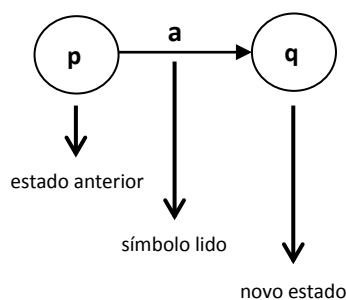


Figura 2.1. Representação da função de transição como um grafo.

Neste caso, os estados iniciais e finais são representados como ilustrado na figura abaixo:

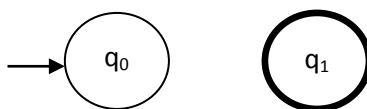


Figura 2.2. Representação de um estado inicial (esq.) e final (dir.) como nodos de grafos.

O processamento de um Autômato Finito **M**, para uma palavra de entrada w , consiste na sucessiva aplicação da função de transição para cada símbolo de w (da esquerda para a direita) até ocorrer uma condição de parada.

A parada de um processamento pode ser de duas maneiras: aceitando ou rejeitando uma entrada w . As condições de parada são as seguintes:

- a) Após processar o último símbolo da fita, o Autômato Finito assume um estado final: o autômato para e a entrada w é aceita;
- b) Após processar o último símbolo da fita, o Autômato Finito assume um estado não-terminal: o autômato para e a entrada w é rejeitada;

- c) A função programa é indefinida para o argumento (estado corrente e símbolo lido): a máquina para e a palavra de entrada w é rejeitada.

2.1. FUNÇÃO PROGRAMA ESTENDIDA

Seja $M = \{\Sigma, Q, \delta, q_0, F\}$ um Autômato Finito Determinístico. A Função Programa Estendida denotada por:

$$\underline{\delta}: Q \times \Sigma^* \rightarrow Q$$

é a função programa $\delta: Q \times \Sigma \rightarrow Q$ estendida para palavras e é indutivamente definida como segue:

$$\begin{aligned}\underline{\delta}(q, \varepsilon) &= q \\ \underline{\delta}(q, aw) &= \underline{\delta}(\delta(q, a), w)\end{aligned}$$

A linguagem aceita por um Autômato Finito $M = \{\Sigma, Q, \delta, q_0, F\}$, denotada por $ACEITA(M)$ ou $L(M)$, é o conjunto de todas as palavras pertencentes a Σ^* aceitas por M , ou seja:

$$ACEITA(M) = \{w \mid \underline{\delta}(q_0, w) \in F\}$$

Analogamente, $REJEITA(M)$ é o conjunto de todas as palavras pertencentes a Σ^* rejeitadas por M .

As seguintes afirmações são verdadeiras:

- $ACEITA(M) \cap REJEITA(M) = \emptyset$
- $ACEITA(M) \cup REJEITA(M) = \Sigma^*$
- o complemento de $ACEITA(M)$ é $REJEITA(M)$
- o complemento de $REJEITA(M)$ é $ACEITA(M)$

Uma linguagem aceita por um Autômato Finito Determinístico é uma *Linguagem Regular* ou *Tipo 3*.

Exemplo 1:

Seja a linguagem:

$$L_1 = \{w \mid w \text{ possui } aa \text{ ou } bb \text{ como subpalavra}\}$$

o autômato Finito:

$$M_1 = \{\{a, b\}, \{q_0, q_1, q_2, q_3\}, \delta_1, q_0, \{q_3\}\}$$

onde δ_1 é como abaixo, representada na forma de uma tabela, reconhece a linguagem L_1 .

δ_1	a	b
q_0	q_1	q_2
q_1	q_3	q_2
q_2	q_1	q_3
q_3	q_3	q_3

Na figura abaixo vemos o grafo que representa o Autômato M_1 :

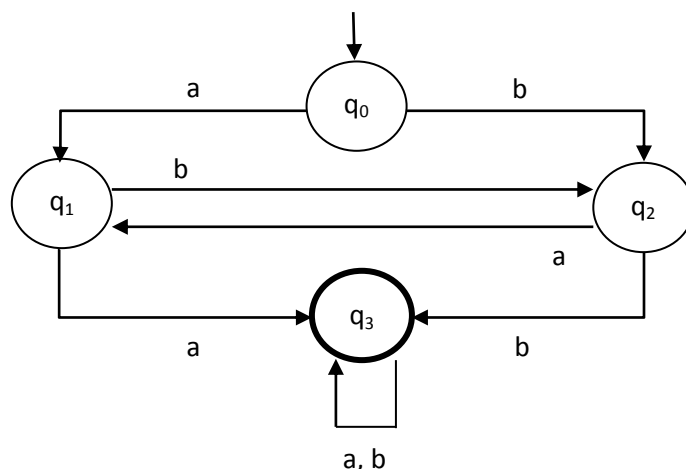


Figura 2.3. Grafo do Autômato Finito Determinístico M_1 .

A função programa estendida aplicada à palavra **abaa** a partir do estado inicial q_0 é como segue:

$\delta(q_0, abaa) =$	função estendida sobre abaa
$\delta(\delta(q_0, a), baa) =$	processa <u>a</u> baa
$\delta(q_1, baa) =$	função estendida sobre baa
$\delta(\delta(q_1, b), aa) =$	processa <u>b</u> aa
$\delta(q_2, aa) =$	função estendida sobre aa
$\delta(\delta(q_2, a), a) =$	processa <u>a</u> a
$\delta(q_1, a) =$	função estendida sobre a
$\delta(\delta(q_1, a), \epsilon) =$	processa <u>a</u>
$\delta(q_3, \epsilon) = q_3$	função estendida sobre ϵ : fim da indução

a palavra **abaa** é aceita pelo Autômato M_1 , pois q_3 , pertence ao conjunto dos estados terminais.

3. SIMULADOR DE AUTÔMATO FINITO DETERMINÍSTICO

O Simulador de Autômato Finito é um programa feito em linguagem de programação Java que visa a criar a gramática de um Autômato Finito Determinístico qualquer e fazer o processamento de uma palavra de entrada, exibindo o resultado do processamento em forma de Função Programa Estendida (ver *capítulo 2.1*).

A tela principal do programa é mostrada na figura abaixo:

Simulador de Autômato Finito Determinístico

Alfabeto (Σ)
 Símbolos. (Ex.: a, b, c)
 1
 a
 b
 2

Estados (Q)
 Estados. (Ex.: q1, q2, q3)
 3
 Estados Não-Terminais
 q0
 q1
 q2
 4

Estado Inicial (q0)
 q0
 5

Estados Terminais (F)
 q3
 6

Função de Transição (δ)
 7

	a	b
q0	q1	q2
q1	q3	q2
q2	q1	q3
q3	q3	q3

Palavra
 abaa
 8

Figura 3.1. Tela Principal do simulador de Autômato Finito Determinístico.

Os círculos em amarelo indicam as posições de campos de edição da gramática do Autômato Finito, conforme a descrição abaixo:

1. *Campo de edição do Alfabeto.* Este campo será usado para digitar os símbolos a serem inseridos no alfabeto;
2. *Lista de Símbolos do Alfabeto.* Lista contendo todos os símbolos inseridos no alfabeto;
3. *Campo de edição dos Estados.* Este campo será usado para digitar os estados do Autômato Finito;
4. *Lista de Estados Não-Terminais.* Lista contendo todos os estados do Autômato que não são estados finais;
5. *Campo de definição do Estado Inicial.* Este campo será utilizado para se definir qual o estado inicial do Autômato;

6. *Lista de Estados Terminais*. Lista contendo todos os estados definidos como terminais;
7. *Tabela Função de Transição*. Tabela utilizada para construir a Função de Transição do Autômato Finito.
8. *Campo de edição da Palavra de Entrada*. Este campo será utilizado para digitar a palavra de entrada a ser processada pelo Autômato Finito definido.

3.1. INSERINDO/REMOVENDO SÍMBOLOS

Para inserir um novo símbolo no alfabeto, digite o símbolo no *Campo de edição do Alfabeto* [1] e tecla ENTER. Se for mais do que um símbolo, pode digitar cada um deles separado por vírgula sem deixar espaço e teclar ENTER. Ao inserir, cada símbolo válido será listado na *Lista de Símbolos* [2].

Pode ocorrer de o usuário digitar uma construção inválida, como uma que não seja caractere único (ex.: aa, ab, can) ou que gere símbolos que já tenham sido inseridos. Se isso ocorrer, vai ser exibida a seguinte caixa de mensagens, listando as construções que não foram aceitas como símbolos válidos:

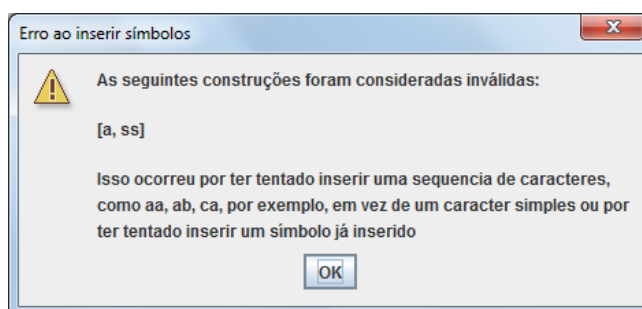


Figura 3.2. Caixa de mensagem "Erro ao inserir símbolos".

Para remover símbolos, escolha os que deseja remover na *Lista de Símbolos* [2], segurando a tecla CTRL e clicando em cada um que vai ser removido, depois tecla DELETE, ou se preferir, use o botão de contexto do mouse (botão direito). Neste último caso, vai aparecer o seguinte menu suspenso:



Figura 3.3. Menu "Manipulação dos símbolos do Alfabeto".

escolha a opção *Remover*.

Qualquer que seja a forma escolhida para remover os símbolos do alfabeto, a seguinte caixa de diálogo será exibida:

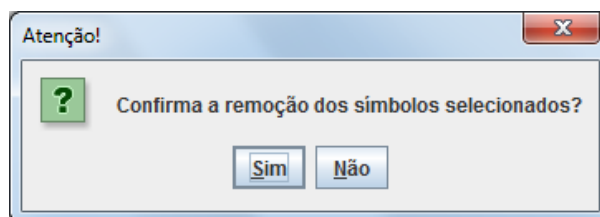


Figura 3.4. Caixa de diálogo “Remover símbolos selecionados”.

clique no botão **Sim** e os símbolos selecionados serão removidos.

Se for para remover todos os símbolos do Alfabeto, use a opção de menu *Limpar Alfabeto*. A seguinte caixa de diálogo será exibida:

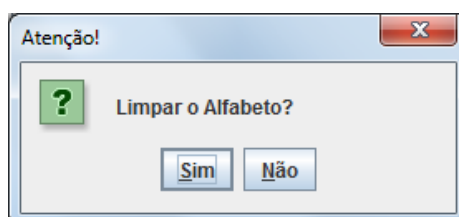


Figura 3.5. Caixa de diálogo “Limpar o Alfabeto”.

clique no botão **Sim** e todos os símbolos serão removidos.

3.2. INSERINDO/REMOVENDO ESTADOS

Para inserir um novo estado, digite o nome do estado no *Campo de edição dos Estados [3]* e tecele ENTER. Se for mais do que um estado, pode digitar o nome de cada um deles separado por vírgula e sem espaços e teclar ENTER.

A sintaxe correta para o nome de um Estado deverá ser a seguinte:

$$q_n$$

onde:

- q** letra ‘q’ do alfabeto;
- n** número positivo.

Todos os estados inseridos serão incluídos, inicialmente, na *Lista de Estados Não-Terminais [4]*.

Pode ocorrer de o usuário digitar algum nome de estado que não obedeça a esta sintaxe (ex.: qa, qm, q-1, a2), ou que já tenha sido inserido. Neste caso, a seguinte caixa de diálogo será exibida, listando as construções que não foram aceitas como estados válidos:

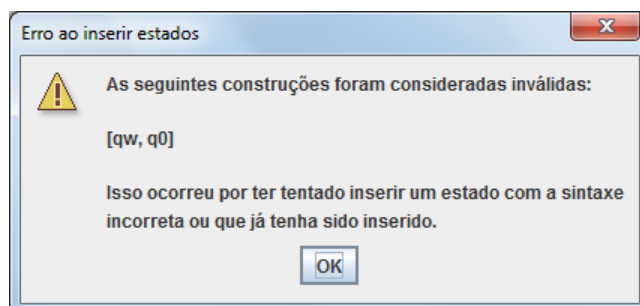


Figura 3.6. Caixa de mensagem "Erro ao inserir estados".

Para remover estados, selecione os que você deseja que sejam removidos na *Lista de Estados Não-Terminais* [4] ou na *Lista de Estados Terminais* [6], segurando a tecla CTRL e clicando em cada um que deve ser removido e tecle DELETE ou use o botão de contexto do mouse (botão direito). Neste último caso, um menu suspenso será exibido. No caso da *Lista dos Estados Não-Terminais*, o menu terá o seguinte aspecto:

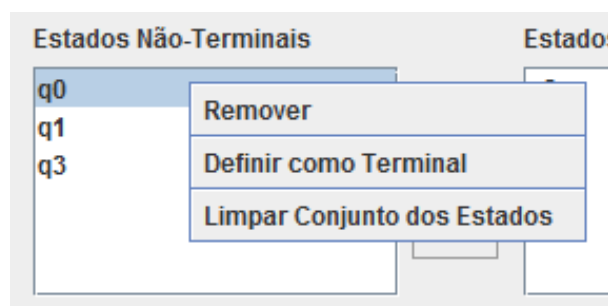


Figura 3.7. Menu "Manipulação de estados não-terminais".

clique em *Remover*.

Qualquer que seja a forma escolhida para remover os estados, a seguinte caixa de diálogo será exibida:

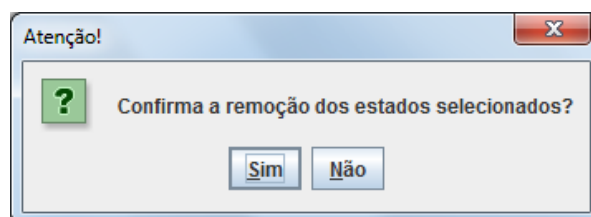


Figura 3.8. Caixa de diálogo "Remover estados selecionados".

clique no botão **Sim**, e os estados selecionados serão removidos.

O menu suspenso da Lista de Estados Terminais também têm uma opção *Remover*. O processo para exclusão é o mesmo.

Se quiser remover todos os estados do conjunto dos estados, selecione a opção de menu *Limpar Conjunto dos Estados*, quer seja no menu da *Lista de Estados Terminais*, quer seja no da *Lista de Estados Não-Terminais*. A seguinte caixa de diálogo será exibida:

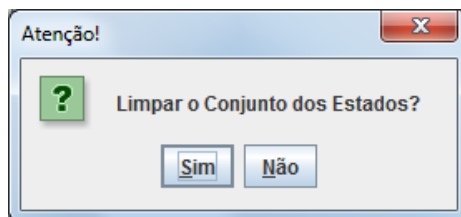


Figura 3.9. Caixa de diálogo “Limpar Conjunto dos Estados”.

clique no botão **Sim**, e todos os estados (terminais e não-terminais) serão removidos.

3.3. DEFININDO O CONJUNTO DOS ESTADOS TERMINAIS

O conjunto dos estados Terminais (ou finais) é obtido a partir da *Lista de Estados Não-Terminais* [4].

Selecione os estados que deseja que sejam definidos como terminais na *Lista de Estados Não-Terminais*, segurando a tecla CTRL e clicando em cada um que deve ser movido, e clique com o botão de contexto do mouse (botão direito). O seguinte menu será exibido:

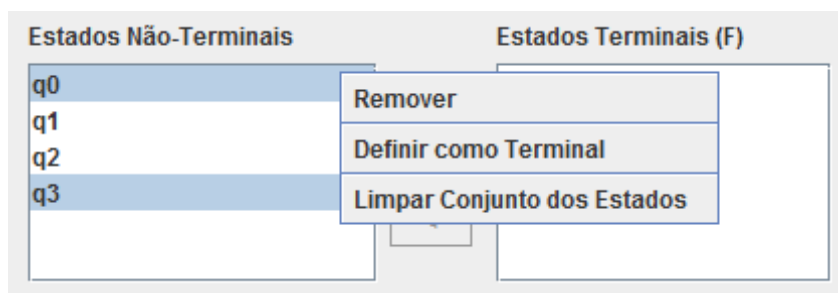


Figura 3.10. Menu “Manipulação de estados não-terminais”.

clique em *Definir como Terminal*. Os estados selecionados passarão à *Lista dos Estados Terminais* [6] à direita.

De forma análoga, para definir os estados terminais como não terminais novamente, selecione os estados na *Lista dos Estados Terminais*, seguindo o mesmo procedimento e clique com o botão de contexto do mouse (botão direito). O seguinte menu será exibido:

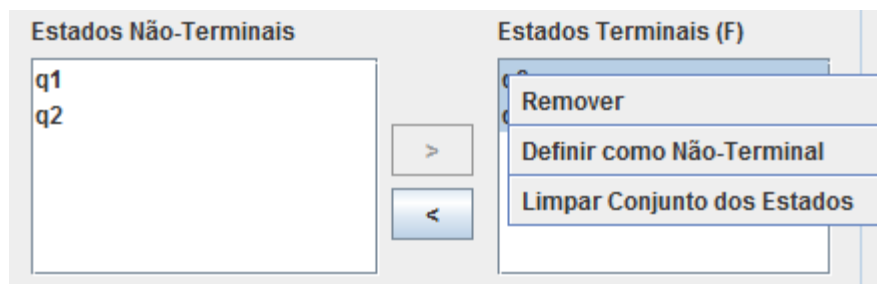


Figura 3.11. Menu “Manipulação de estados terminais”.

clique em *Definir como Não-Terminal*. Os estados selecionados voltarão à Lista dos Estados Não-Terminais à esquerda.

Pode se definir os estados como terminais selecionando os estados que deseja que sejam movidos da *Lista de Estados Não-Terminais* para a *Lista de Estados Terminais*, e clicando no botão *Mover para a Lista de Estados Terminais*, conforme indicado na figura abaixo:

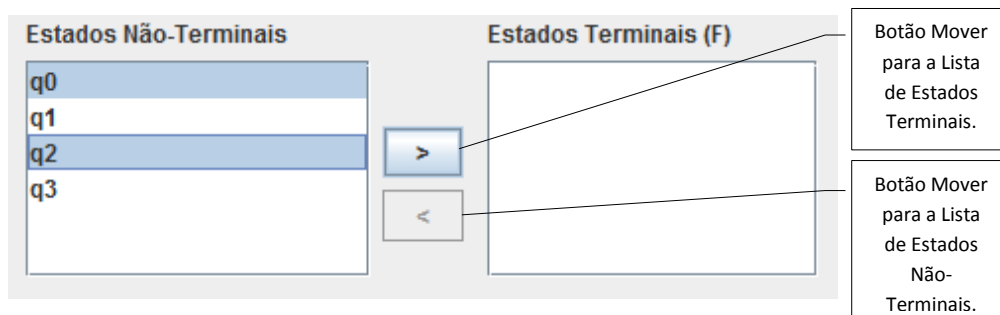


Figura 3.12. Botões “mover os estados”.

De forma análoga, use o botão *Mover para a Lista de Estados Não-Terminais* para mover os estados da *Lista de Estados Terminais* para a *Lista de Estados Não-Terminais*.

Há ainda uma terceira forma de trocar os estados de lista. Basta dar um duplo clique no estado que deseja mover para a outra lista.

3.4. DEFININDO O ESTADO INICIAL

Expandir o *Campo de definição do Estado Inicial* [5], e escolha o estado que será o estado inicial do autômato, conforme mostra a figura abaixo:

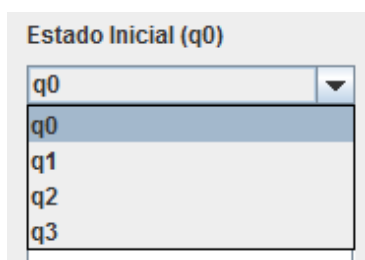


Figura 3.13. Campo para a definição do Estado Inicial.

o estado que foi selecionado será exibido neste campo.

3.5. DEFININDO A FUNÇÃO DE TRANSIÇÃO

A Função de Transição será definida por meio da *Tabela Função de Transição* [7]. Esta tabela tem no topo todos os símbolos do alfabeto e na primeira coluna todos os estados. Tais células não podem ser editadas e aparecem com

uma cor diferenciada (cinza) e com bordas mais espessas, conforme mostra a figura abaixo:

Função de Transição (δ)		
	a	b
q0	q1	q2
q1	q3	q2
q2	q1	q3
q3	q3	q3

Figura 3.14. Tabela Função de Transição.

Cada célula editável têm uma lista suspensa contendo todos os estados do Autômato, conforme mostra a figura abaixo:

Função de Transição (δ)	
	a
q0	<div>▼</div>
q1	q0
q2	q1
q3	q2
	q3

Figura 3.15. Componente de edição da Tabela Função de Transição.

Cada par **Estado atual/Símbolo lido** deve levar a um novo estado que será definido nestas células, com base no estado na linha e no símbolo no topo da coluna.

3.6. PROCESSANDO UMA PALAVRA DE ENTRADA

A palavra que vai ser processada pelo Autômato deve ser digitada no *Campo de edição da Palavra de Entrada* [8]. Após digitar a palavra de entrada, tecle ENTER ou clique no botão *Processar Palavra* ao lado do campo de edição, conforme mostra a figura abaixo:

Palavra
abaa
...

Figura 3.16. Campo de edição da Palavra de Entrada.

Pode haver erros ao processar a palavra de entrada, como um alfabeto não definido, ou alguma transição não definida. Nestes casos, uma mensagem será exibida, informando o erro ocorrido. Digite todos os símbolos e estados e defina cada transição e volte a clicar no botão processar palavra.

O resultado será exibido na *Tela Resultado do Processamento*, conforme mostra a figura abaixo:

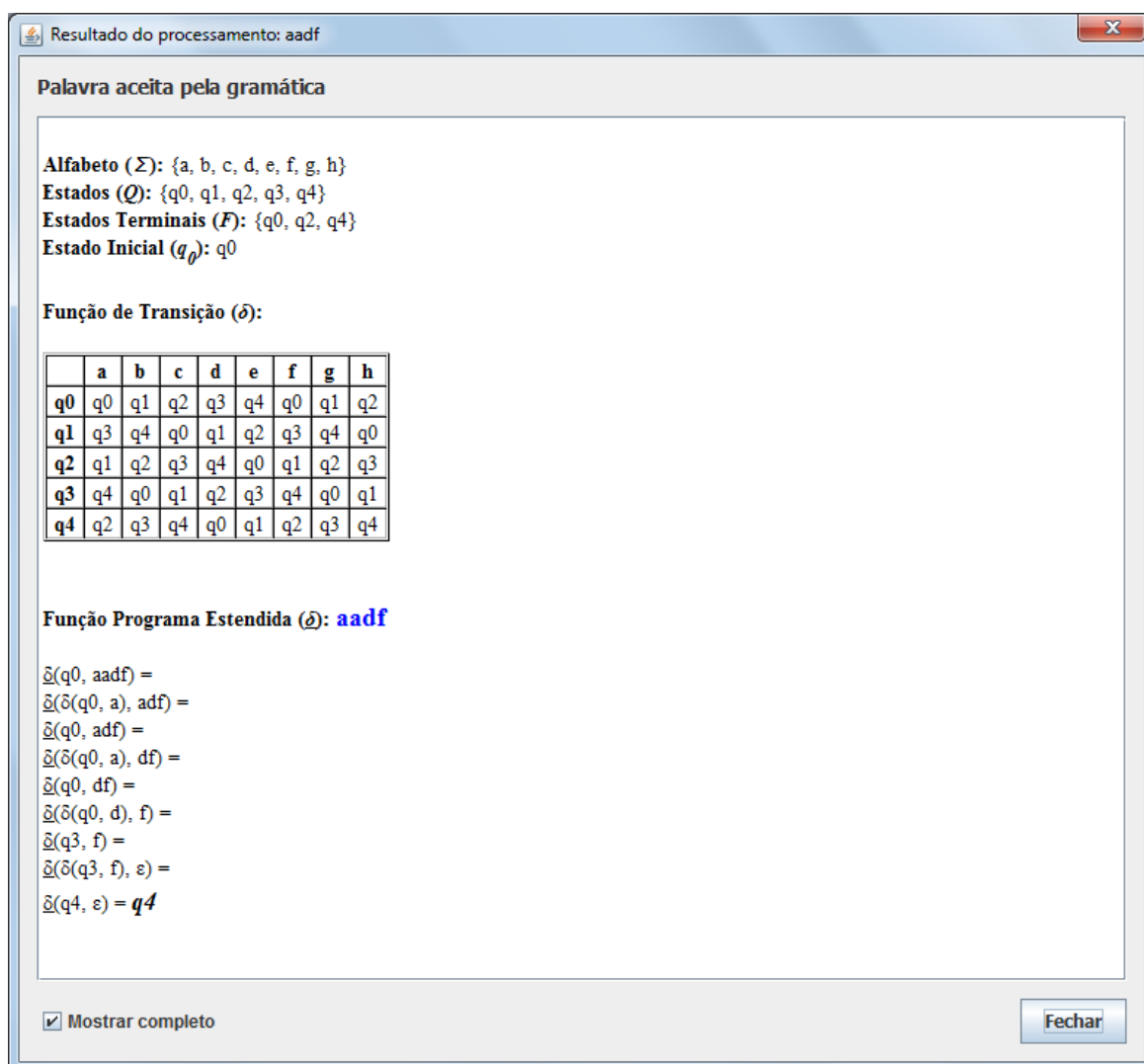


Figura 3.17. Tela Resultado do Processamento (Modo completo).

Note que todos os termos da gramática do autômato são exibidos nesta tela. Isso ocorre quando o campo *Mostrar completo* está marcado. Se desejar, pode exibir apenas o resultado da *Função Programa Estendida* (Capítulo 2.1), desmarcando o campo *Mostrar completo*. Ao marcar ou desmarcar, a opção escolhida será salva para as próximas vezes que executar o programa.

4. BIBLIOGRAFIA CONSULTADA

Menezes, Paulo Fernando Blauth. *Linguagens Formais e Autômatos* / Paulo Fernando Blauth Menezes - Porto Alegre: Instituto de Informática da UFRGS: Editora Sagra Luzzato, 2000. (Série livros didáticos, número 3).

Lewis, Harry R. *Elementos de teoria da computação* / Harry R. Lewis e Christos H. Papadimitriou; trad. Edson Furmankiewicz. 2ª. Ed. – Porto Alegre: Bookman, 2000.

APÊNDICE 1. GERANDO O AUTÔMATO DO EXEMPLO DO CAPÍTULO 2

No **capítulo 2** temos um exemplo de linguagem a ser reconhecida por um Autômato Finito (*Exemplo 1*).

Vamos rever:

a linguagem:

$$L_1 = \{w \mid w \text{ possui } aa \text{ ou } bb \text{ como subpalavra}\}$$

o Autômato Finito que reconhece a linguagem:

$$M_1 = \{\{a, b\}, \{q_0, q_1, q_2, q_3\}, \delta_1, q_0, \{q_3\}\}$$

δ_1	a	b
q_0	q_1	q_2
q_1	q_3	q_2
q_2	q_1	q_3
q_3	q_3	q_3

a palavra ser processada pelo autômato:

abaa

Vamos fornecer estes dados para o simulador e processar a palavra *abaa*.

Passo 1: Digite os símbolos do alfabeto no *Campo de edição do Alfabeto*.

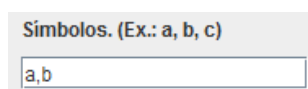


Figura 1. Símbolos do alfabeto.

Ao teclar ENTER, os símbolos serão listados na *Lista de Símbolos do Alfabeto*.

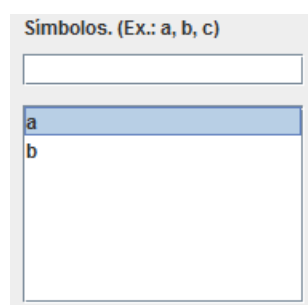
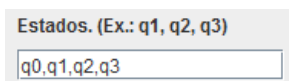


Figura 2. Lista de símbolos do alfabeto.

Passo 2: Digite os estados do Autômato no *Campo de edição dos Estados*.

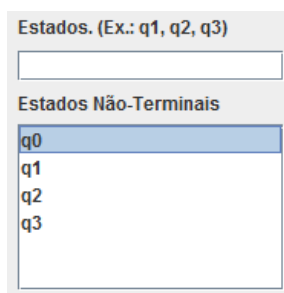


Estados. (Ex.: q1, q2, q3)

q0,q1,q2,q3

Figura 3. Estados do Autômato.

Ao teclar ENTER, os estados serão listados na Lista dos Estados Não-Terminais.



Estados. (Ex.: q1, q2, q3)

Estados Não-Terminais

- q0
- q1
- q2
- q3

Figura 4. Lista de Estados Não-Terminais.

Passo 3: Expanda a Lista *Campo de definição do Estado Inicial* e escolha *q0*.

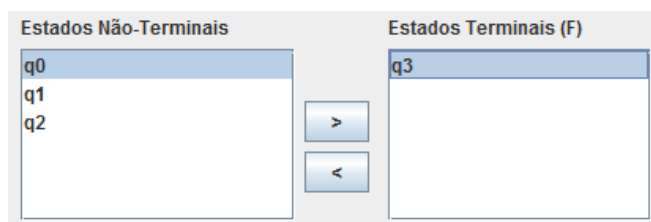


Estado Inicial (q0)

- q0
- q1
- q2
- q3

Figura 5. Campo de Definição do Estado Inicial.

Passo 4: Dê um duplo clique em *q3* na *Lista de Estados Não-Terminais*. Ele vai passar para a *Lista de Estados Terminais*.



Estados Não-Terminais

- q0
- q1
- q2

Estados Terminais (F)

- q3

>

<

Figura 6. Lista de Estados Terminais.

Passo 5: Edite cada célula da Tabela Função de Transição de forma que fique assim:

Função de Transição (δ)		
	a	b
q0	q1	q2
q1	q3	q2
q2	q1	q3
q3	q3	q3

Figura 7. Tabela Função de Transição.

Passo 5: Digite a palavra de entrada *abaa* no Campo de Edição da Palavra de Entrada e tecle ENTER.

Palavra

abaa

Figura 8. Palavra de Entrada.

A seguinte tela será exibida:

Resultado do processamento: abaa

Palavra aceita pela gramática

Alfabeto (Σ): {a, b}
 Estados (Q): {q0, q1, q2, q3}
 Estados Terminais (F): {q3}
 Estado Inicial (q_0): q0

Função de Transição (δ):

	a	b
q0	q1	q2
q1	q3	q2
q2	q1	q3
q3	q3	q3

Função Programa Estendida ($\hat{\delta}$): **abaa**

$\hat{\delta}(q0, abaa) =$
 $\hat{\delta}(\delta(q0, a), baa) =$
 $\hat{\delta}(q1, baa) =$
 $\hat{\delta}(\delta(q1, b), aa) =$
 $\hat{\delta}(q2, aa) =$
 $\hat{\delta}(\delta(q2, a), a) =$
 $\hat{\delta}(q1, a) =$
 $\hat{\delta}(\delta(q1, a), \epsilon) =$
 $\hat{\delta}(q3, \epsilon) = \mathbf{q3}$

☒ Mostrar completo

Fechar

Figura 8. Tela Resultado do Processamento.

Ela exibe todos os detalhes da gramática do Autômato Finito Determinístico criado e o resultado do processamento da Palavra de Entrada em forma de Função Programa estendida.