

Simulador de Autômato Finito Determinístico em Java

Simulador de Autômato Finito Determinístico - leandro_almeida.email@yahoo.com.br

Alfabeto (Σ)
Símbolos. (Ex.: a, b, c)

a
d
f

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

Estados Não-Terminais (Q - F)

q0
q2

Estado Inicial (q0)

q0

Estados Terminais (F)

q1

>
<

Função de Transição (δ)

	a	d	f
q0	q0	q1	q1
q1	q0	q1	q1
q2	q0	q1	q1

Palavra

afd

...

Leandro Aparecido de Almeida

Sumário

1. Introdução.....	1
2. Definição de Autômato Finito Determinístico.....	2
3. Funcionamento do Simulador.....	3
Bibliografia consultada.....	8

1. Introdução

Um autômato finito determinístico é um modelo extremamente restrito de computador. Ele compartilha com um computador real a característica de que ele tem uma "unidade central de processamento" com capacidade fixa, finita. Ele recebe como entrada uma **cadeia** que lhe é fornecida em uma **Fita de Entrada**, com um símbolo gravado em cada célula da fita. Ele não produz nenhuma saída, exceto uma indicação, informando se essa entrada foi ou não aceita.

A parte principal da máquina propriamente dita é um bloco com "mecanismos" internos que podem se apresentar em um momento específico em algum dos seus diferentes estados internos. Esse bloco – denominado **unidade de controle** – pode ler o símbolo gravado em qualquer posição da fita de entrada com a ajuda de um **cabeçote móvel de leitura (unidade de leitura)**. Inicialmente, o cabeçote de leitura está posicionado na célula mais à esquerda da fita, e a unidade de controle, configurada em um **estado inicial** preestabelecido. A intervalos regulares, o autômato lê um símbolo da fita de entrada e entra em um novo estado, que depende somente do estado atual e do símbolo que acabou de ser lido. Após a leitura de um símbolo de entrada, o cabeçote de leitura se movimenta uma célula para a direita da fita de entrada, posicionando-se de modo que, no próximo movimento ele leia o símbolo contido na próxima célula da fita. Esse processo é repetido continuamente até que o autômato alcance o final da Fita de Entrada. Caso o estado a que ele foi conduzido com a leitura do último símbolo da Fita de Entrada pertença ao conjunto dos **estados finais** do autômato, então a cadeia é considerada aceita, caso contrário, não. A linguagem aceita pela máquina é o conjunto das cadeias que ela reconhece.

Uma representação do autômato Finito Determinístico é mostrado na figura abaixo:

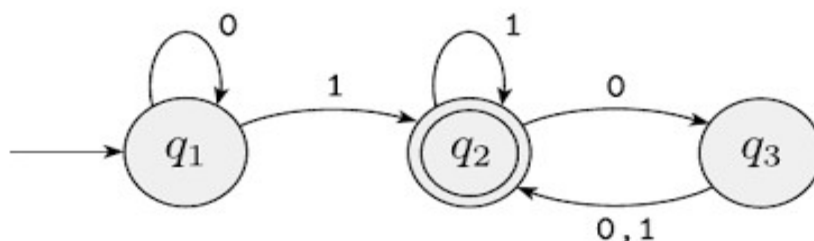


Figura 1. Modelo de Autômato Finito Determinístico.

No próximo capítulo, expressaremos formalmente o autômato em termos matemáticos. No restante do texto estudaremos o funcionamento do simulador.

2. Definição de Autômato Finito Determinístico

Um autômato finito determinístico é uma 5-tupla:

$$M=(\Sigma, Q, \delta, q_0, F)$$

Onde:

Σ : Alfabeto dos símbolos de entrada;

Q : Conjunto dos estados;

δ : Função Programa, ou Função de Transição, tal que: $\delta: Q \times \Sigma \rightarrow Q$, que é uma função parcial;

q_0 : Estado inicial, que pertence a Q ;

F : Conjunto dos Estados Finais ou Terminais, contido em Q .

O processamento de um Autômato Finito M , para uma palavra de entrada w , consiste na sucessiva aplicação da Função Programa para cada símbolo de w , da esquerda para a direita, até ocorrer uma situação de parada.

As condições de parada são as seguintes:

- *Após processar o último símbolo da fita, o Autômato assume um estado terminal*: o autômato para e a entrada w é aceita;
- *Após processar o último símbolo da fita, o Autômato assume um estado não-terminal*: o autômato para e a entrada w é rejeitada;
- *A função programa é indefinida para o argumento (estado corrente e símbolo lido)*: A máquina para e a palavra de entrada w é rejeitada.

3. Funcionamento do Simulador

A tela do simulador é mostrada na figura abaixo:

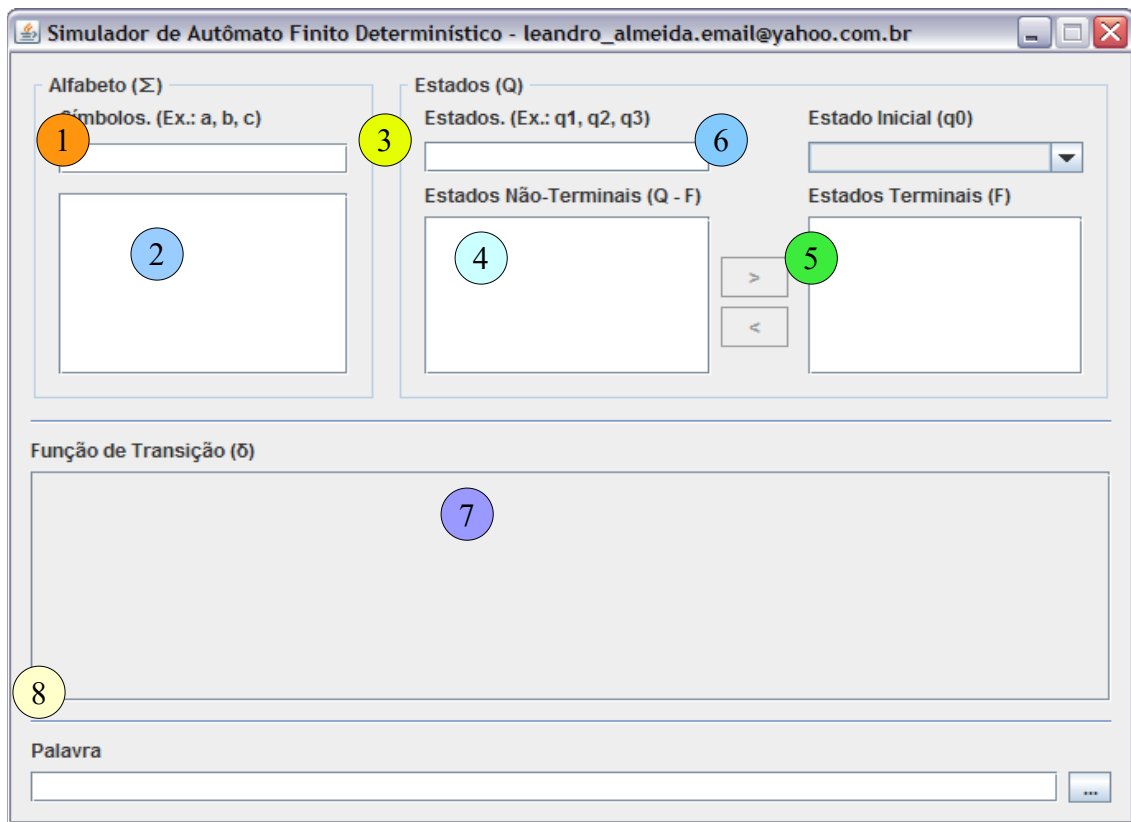


Figura 2. Tela principal do Simulador.

Abaixo o significado de cada campo marcado por um círculo na figura:

1. **Entrada do Alfabeto:** Campo para entrada dos símbolos do alfabeto (Σ). Os símbolos são caracteres ASCII e podem ser inseridos tanto individualmente, quanto separados por vírgula. Após digitar os símbolos, basta teclar ENTER ou TAB, que eles serão listados na **Lista de Símbolos**. Caso o usuário entre com uma construção onde há a presença de mais de um símbolo, como *aa*, *bb*, *vc*, por exemplo, então tal construção não será considerada válida, e será descartada, sendo que as demais, se estiverem corretas, serão inseridas normalmente.
2. **Lista de Símbolos:** Todos os símbolos do alfabeto estarão listados nela. Para excluir um, ou vários símbolos, ou mesmo todos, existe um menu de contexto, conforme mostra a figura abaixo:



Figura 3. Menu de contexto da Lista de Símbolos.

3.

Para selecionar todos os símbolos em um intervalo, segure a tecla SHIFT, selecione o primeiro símbolo do intervalo, depois o último. Todos os símbolos entre estes serão selecionados. Clique em **Remover** e confirme na caixa de diálogo.

Ao remover os símbolos, a **Tabela de Transição** será redesenhada.

Para remover determinados símbolos que não estão num intervalo contínuo, segure a tecla CTRL, e clique nos símbolos que deseja selecionar. Feito isso, clique em **Remover**, e confirme na caixa de diálogo.

Caso queira excluir todos os símbolos, clique em **Limpar Alfabeto**, e confirme na caixa de diálogo.

*Obs.: A opção **Remover** só ficará habilitada quando algum símbolo estiver selecionado na lista.*

4. **Entrada do Conjunto dos estados:** Campo para entrada do conjunto dos estados (Q).

Há uma restrição quanto à nomeação dos estados, que não é uma regra para nome de estados, mas que aqui irá facilitar o procedimento de ordenação no código-fonte. Um estado deverá ter o nome no formato:

qn

Onde:

q: letra *q* do alfabeto;

n: um índice numérico positivo qualquer, não necessariamente sequencial;

Tal nomenclatura é comumente encontrada.

Após digitar os nomes dos estados, tecle ENTER ou TAB, que eles serão listados na **Lista de Estados Não-Terminais**.

Os estados podem ser inseridos tanto individualmente, como separados por vírgula. Se algum estado não tiver a sintaxe correta, então ele será descartado.

5. **Lista de Estados Não-Terminais:** Todos os estados que são inseridos são definidos por padrão como não-terminais e serão listados nela.

Para excluir um ou mais estados não terminais, ou para defini-los como terminais, ou para remover todos os estados, há um menu de contexto, conforme mostra a figura abaixo:

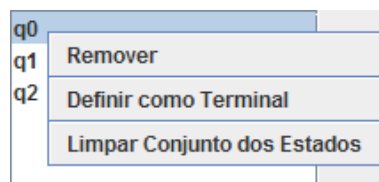


Figura 4. Menu de contexto da Lista de Estados Não Terminais.

O processo para selecionar vários estados para remoção e aqui também para defini-los como terminais é o mesmo da **Lista de Símbolos**.

Para definir um estado como terminal, clique em **Definir como Terminal**. Ele será removido da lista, e passará para a **Lista de Estados Terminais** à direita.

Opcionalmente, você pode fazer a mesma coisa, selecionando o(s) estado(s) e teclando ENTER, ou clicando no botão [>].

6. **Lista de Estados Terminais:** Nela estão listados todos os estados definidos como terminais. Seu funcionamento é o mesmo da Lista de Estados Não-Terminais, mudando apenas a direção para onde os estados serão movidos, caso queira mudá-los para não-terminais;

7. **Definição do Estado Inicial:** Campo para definição do estado inicial. Basta desdobrar a lista e selecionar qual estado será o estado inicial;

8. **Tabela de Transição:** É o componente para a definição da **Função de Transição**. Perceba que no topo são listados os símbolos, e à esquerda os estados. Com base na leitura do símbolo, e estando o

autômato no estado referido, a função deve convergir para um novo estado (eventualmente o mesmo). Logo, a célula formada com o encontro do símbolo com o estado deve receber este estado para o qual a função converge. Veja na figura abaixo a tabela de transição:

	a	d	s
q0	q0	q0	q0
q1	q1	q1	q1
q2	q2	q1	
			<div>q0</div> <div>q1</div> <div>q2</div>

Figura 5. Tabela de Transição.

Para definir o estado, basta clicar na célula referida, e selecionar um estado na lista suspensa, sem precisar digitar nada.

9. **Palavra a ser Analisada:** Campo para entrada da palavra a ser analisada. Para que uma palavra seja verificada, é necessário que todas as transições da Função de Transição estejam definidas. Se tudo correr normalmente, ao teclar ENTER, ou clicar no botão à direita, ou escolher a opção **Verificar** no menu de contexto do campo, a palavra será analisada, e o resultado do processamento será exibido em forma de **Função Programa Estendida**, na tela de resultado do processamento.

A tela é mostrada na figura abaixo:

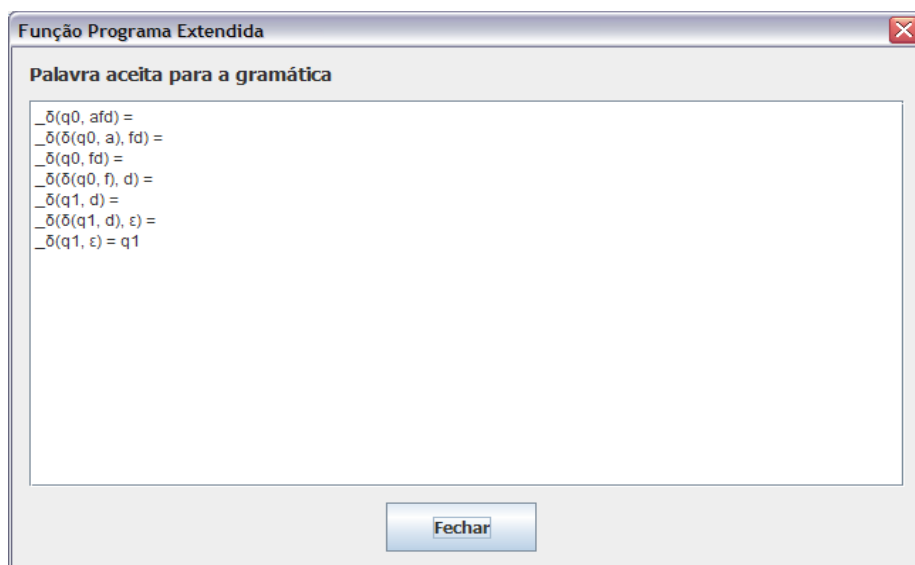


Figura 6. Tela de Resultados de Processamento.

A tela traz como informações se a palavra foi ou não aceita pela gramática e o resultado do processamento da cadeia de entrada em forma de **Função Programa Estendida**.

A Função Programa Estendida, denotada por:

$$\delta: Q \times \Sigma^* \rightarrow Q$$

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

$$\begin{aligned} \underline{\delta}(q_0, \epsilon) &= q \\ \underline{\delta}(q_0, aw) &= \underline{\delta}(\delta(q_0, a), w) \end{aligned}$$

Vejamos um exemplo:

Considere o Autômato $M_1 = (\{a,b\}, \{q_0, q_1, q_2, q_3\}, \delta_1, q_0, \{q_3\})$ um Autômato Finito Determinístico, onde:

δ_1	a	b
q₀	q ₁	q ₂
q₁	q ₃	q ₂
q₂	q ₁	q ₃
q₃	q ₃	q ₃

A Função Programa Estendida aplicada à palavra **abaa** a partir do estado inicial **q₀** 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

Como ao terminar a leitura da cadeia, o autômato convergiu para um estado-final (q_3), então a palavra abaa é aceita por M_1 .

Sugiro que teste o autômato acima, e veja os resultados.

Bibliografia consultada

MENEZES, Paulo 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.