



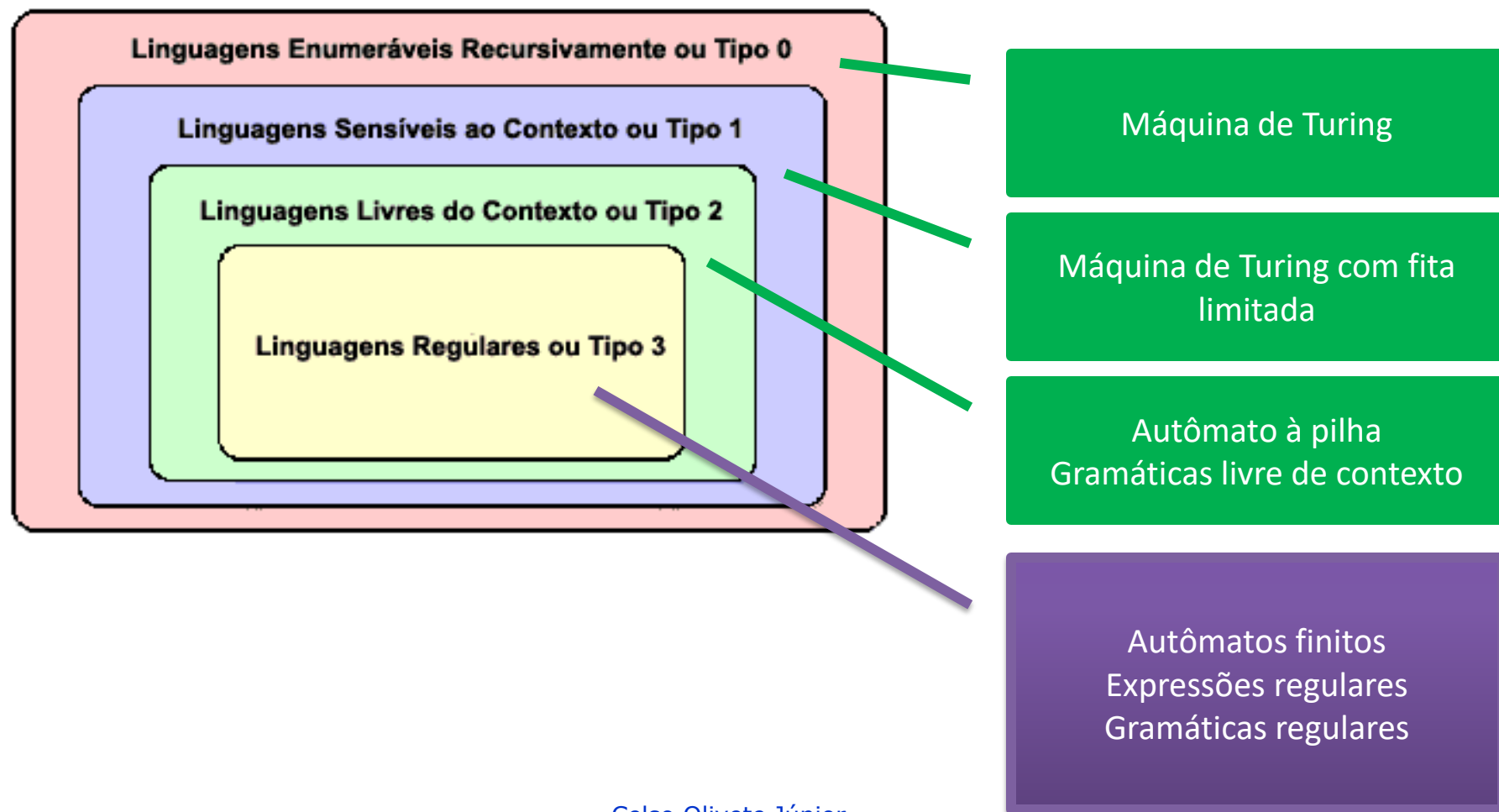
LFA - Aula 04

Autômatos Finitos

Celso Olivete Júnior

celso.olivete@unesp.br

Classificação das Linguagens segundo Hierarquia de Chomsky



Na aula passada...

- Gramáticas regulares
 - linear à direita
 - $A \rightarrow wB$ ou $A \rightarrow w$ com $|w| \geq 0$
 - linear unitária à direita
 - $A \rightarrow wB$ ou $A \rightarrow w$ com $|w| \leq 1$
 - linear à esquerda
 - $A \rightarrow Bw$ ou $A \rightarrow w$ com $|w| \geq 0$
 - linear unitária à esquerda
 - $A \rightarrow wB$ ou $A \rightarrow w$ com $|w| \leq 1$

Na aula de hoje:

- Linguagens regulares: **Autômatos finitos (AFs)**
 - Da mesma forma como ocorre com as expressões regulares (ER's) e com as gramáticas lineares (GL's), os **AFs** também possibilitam a formalização das linguagens regulares.
 - ER's e GL's são dispositivos de geração de sentenças
 - **AF's são dispositivos de aceitação de sentenças - reconhecedores**



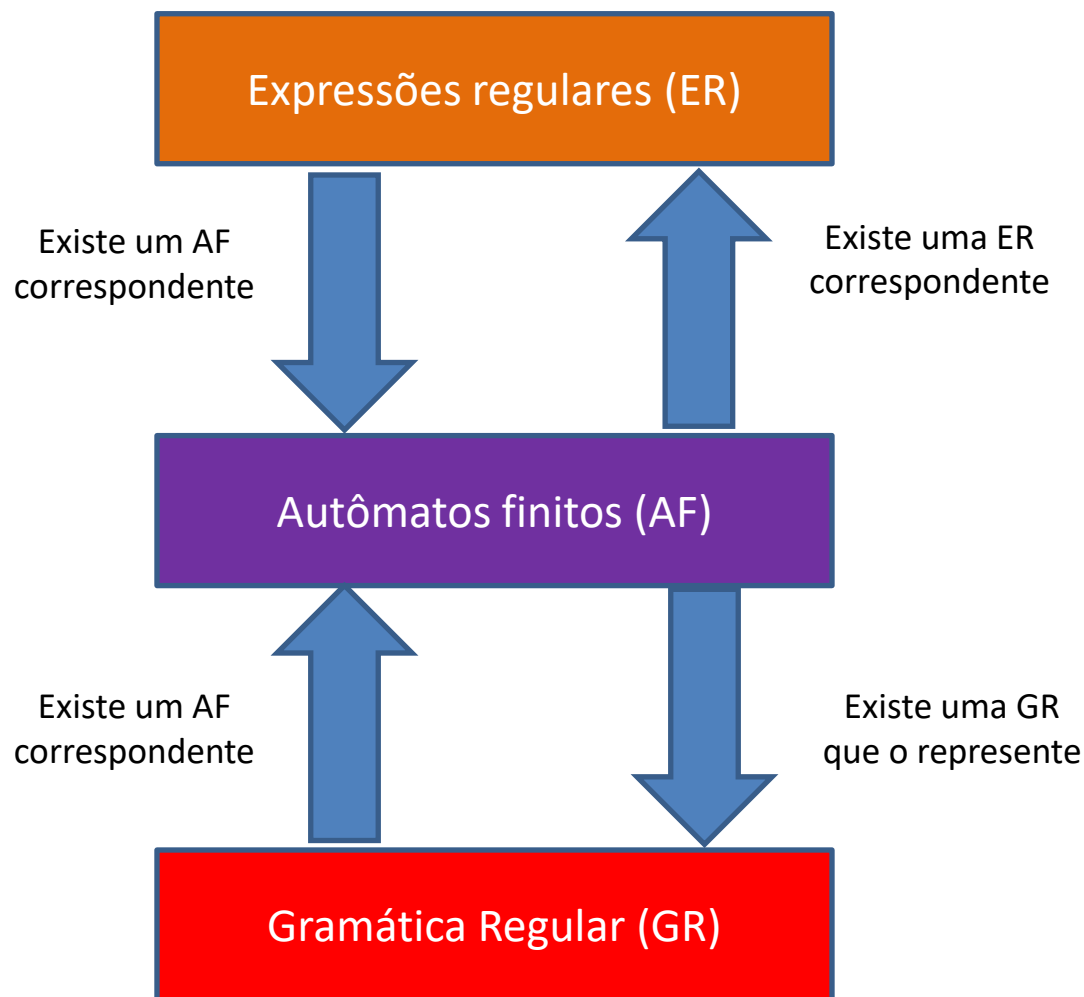
Linguagens regulares: Autômatos finitos(AF's)

- Referências bibliográficas

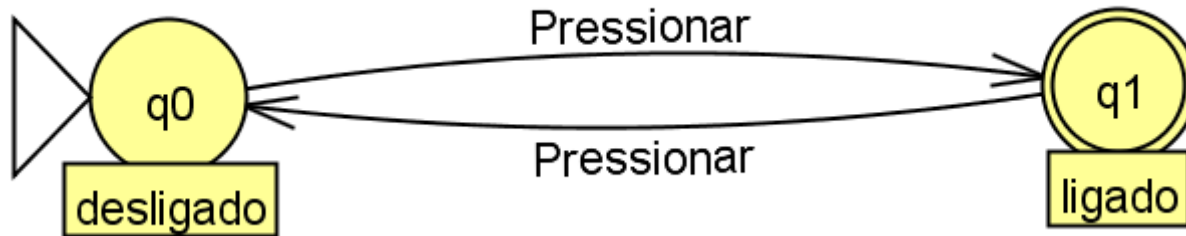
→ HOPCROFT, J. E.; ULLMAN, J. D.; MOTWANI, R. *Introdução à Teoria de Autômatos, Linguagens e Computação*. Editora Campus, 2002 → Capítulo 2

→ RAMOS, M.V.M.; NETO, J.J.; VEGA, I.S. *Linguagens Formais: Teoria, Modelagem e Implementação*. Editora Bookman 2009. → Capítulo 3

Equivalências

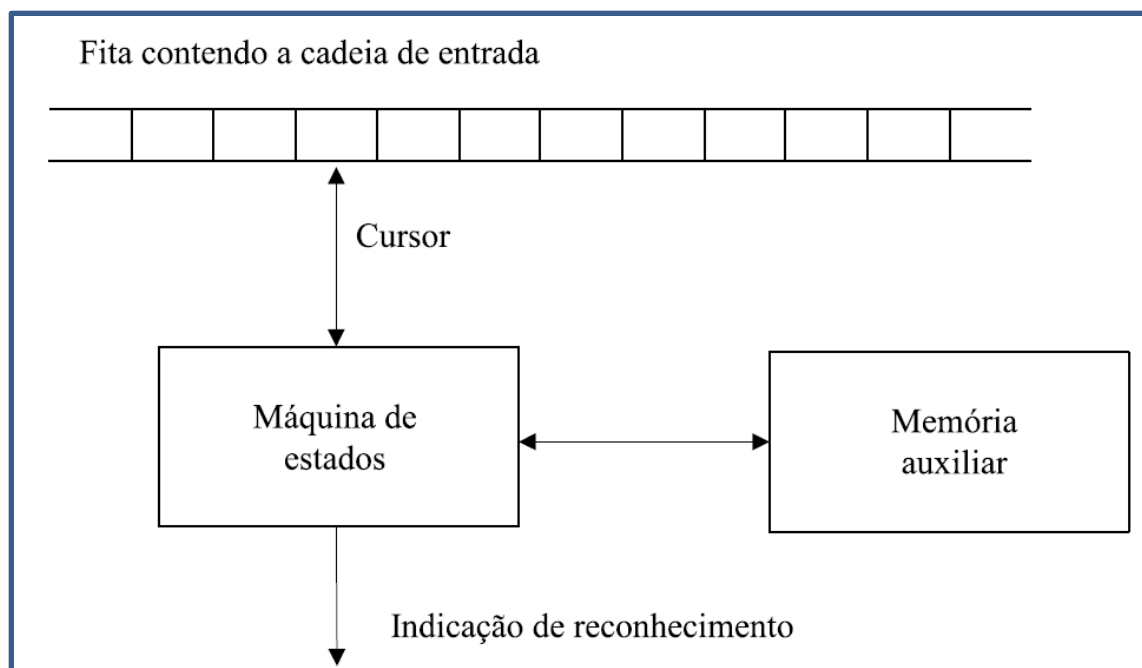


Autômatos finitos: exemplo clássico do interruptor



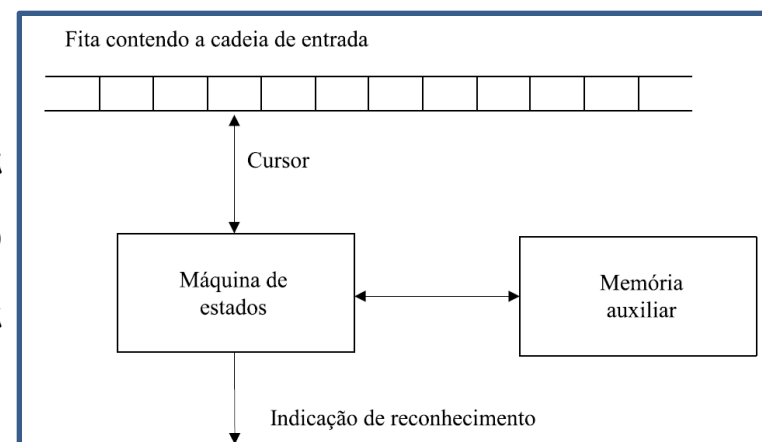
A estrutura de um reconhecedor genérico

- um reconhecedor **genérico** apresenta: uma **memória (fita)** contendo o texto de entrada do reconhecedor, um **cursor**, que indica o próximo elemento da fita a ser processado, uma **máquina de estados finitos**, sem memória, e uma **memória auxiliar opcional**.



Fita de entrada

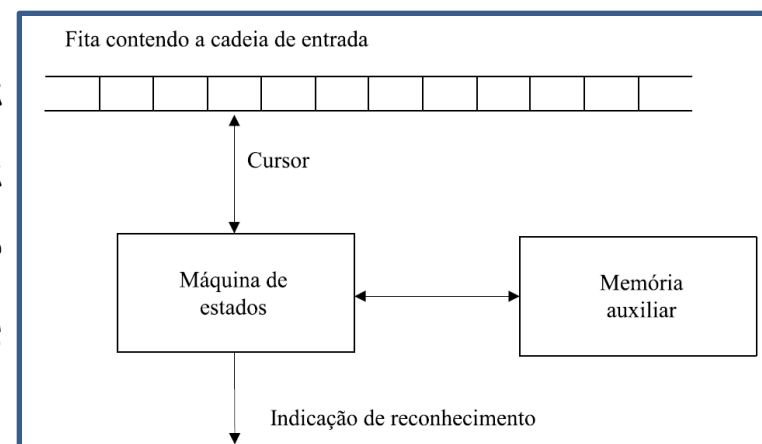
- **Contém a cadeia a ser analisada pelo reconhecedor.** Ela é dividida em células, e **cada célula pode conter um único símbolo da cadeia de entrada**, pertencente ao alfabeto Σ de entrada escolhido para o reconhecedor.
- A cadeia de entrada é **disposta da esquerda para a direita**, sendo o seu **primeiro símbolo** colocado na posição **mais à esquerda da fita**.
- Dependendo do tipo de reconhecedor considerado, **a fita** (ou o conjunto de fitas) de entrada **pode apresentar comprimento finito ou infinito**. Neste último caso, a fita pode ter ou não limitação à esquerda e/ou à direita.
- A **cadeia de entrada** registrada na fita de entrada **pode estar delimitada por símbolos especiais**, não pertencentes ao alfabeto de entrada, à sua esquerda e/ou à sua direita, porém isso não é obrigatório



Cursor

- A **leitura dos símbolos** gravados **na fita** de entrada é **feita através de um cabeçote de acesso**, normalmente denominado **cursor**, o qual sempre **aponta o próximo símbolo** da cadeia a ser processado. Os **movimentos do cursor são controlados pela máquina de estados**, e **podem**, dependendo do tipo de reconhecedor, **ser unidirecionais** (podendo deslocar-se para um lado apenas, tipicamente para a direita) **ou bidirecionais** (podendo deslocar-se para a esquerda e para a direita).

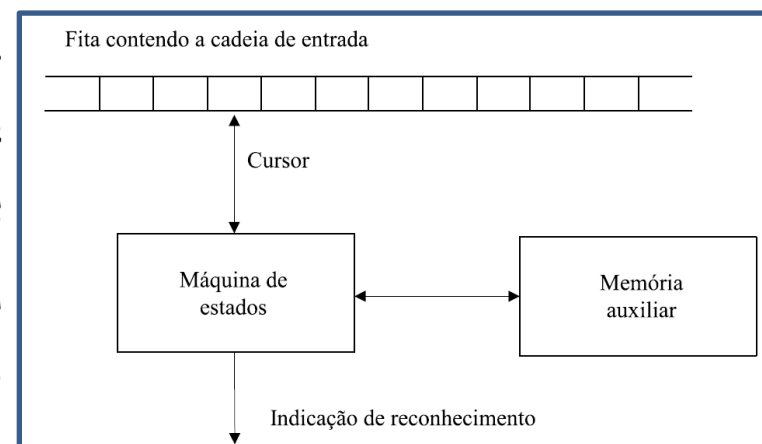
- Determinados tipos de reconhecedores utilizam o **cursor** não apenas para **lerem os símbolos** da fita de entrada, mas também para **escreverem** sobre a **fita**, substituindo símbolos nela presentes por outros, de acordo com comandos determinados pela máquina de estados.



Máquina de estados

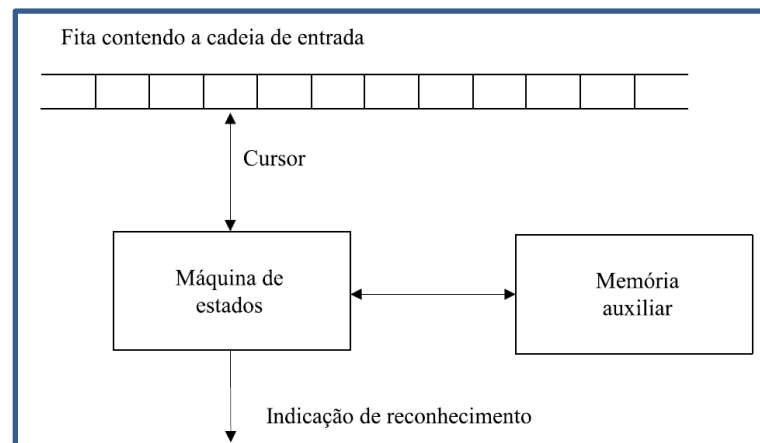
• A **máquina de estados** funciona como um **controlador** central do reconhecedor, e contém uma coleção finita de **estados**, responsáveis pelo registro de informações colhidas no passado, mas consideradas relevantes para decisões futuras, e transições, que **promovem as mudanças de estado** da máquina em sincronismo com as operações efetuadas **através do cursor** sobre a fita de entrada.

• Além disso, a máquina de estados finitos **pode utilizar uma memória auxiliar para armazenar e consultar outras informações**, também coletadas ao longo do processamento, que sejam eventualmente necessárias ao completo reconhecimento da cadeia de entrada.



Memória auxiliar

- A memória auxiliar é **opcional**, e torna-se necessária apenas em reconhecedores de linguagens que apresentam uma certa complexidade. Normalmente, ela **assume a forma de uma estrutura de dados** de baixa complexidade, como, por exemplo, uma **pilha** (no caso do reconhecimento de linguagens livres de contexto ou **TIPO 2**).
- As informações registradas na memória auxiliar são codificadas com base em um alfabeto de memória, e todas as operações de manipulação da memória auxiliar (leitura e escrita) fazem referência apenas aos símbolos que compõem esse alfabeto.
- Os elementos dessa memória são referenciados através de um cursor auxiliar que, eventualmente, poderá coincidir com o próprio cursor da fita de entrada. Seu tamanho não é obrigatoriamente limitado e, por definição, seu conteúdo pode ser consultado e modificado.



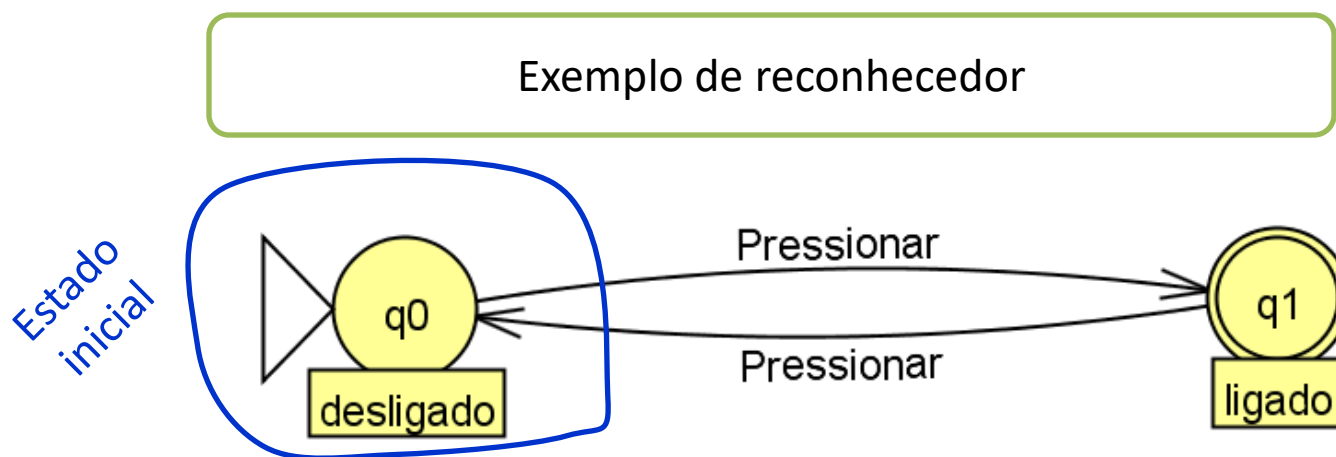
Funcionamento do reconhecedor

- A operação de um reconhecedor baseia-se em uma sequência de **movimentos** que o **conduzem**, de uma **configuração inicial única**, para alguma **configuração de parada**, indicativa do sucesso ou do fracasso da **tentativa de reconhecimento** da cadeia de entrada.
- A configuração de um reconhecedor genérico é caracterizada pela quádrupla:
 1. Estado;
 2. Conteúdo da fita de entrada;
 3. Posição do cursor;
 4. Conteúdo da memória auxiliar.

Configuração inicial

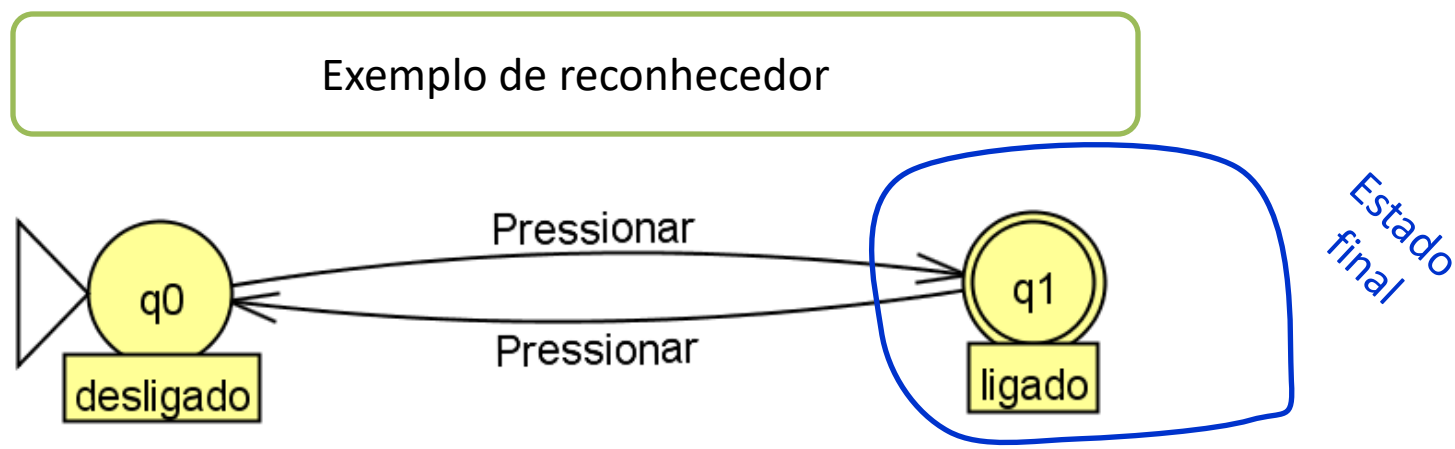
• A **configuração inicial** de um **reconhecedor** é definida como sendo aquela em que as seguintes condições são verificadas:

1. Estado: inicial, único para cada reconhecedor;
2. Conteúdo da fita de entrada: com a cadeia completa a ser analisada;
3. Posição do cursor: apontando para o símbolo mais à esquerda da cadeia;
4. Conteúdo da memória auxiliar: inicial, predefinido e único.



Configuração final

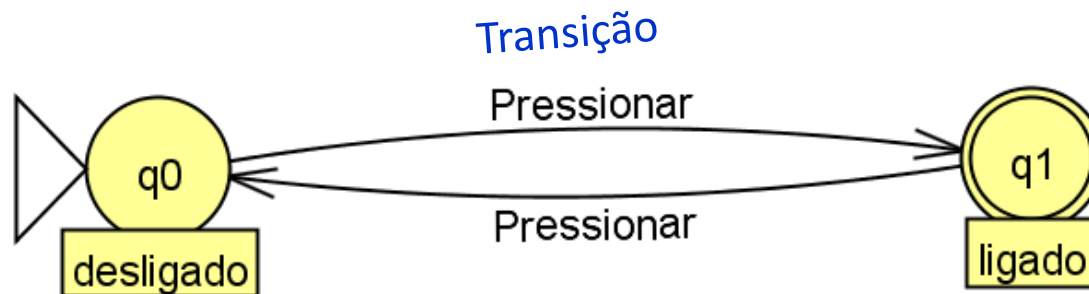
1. Estado: algum dos estados finais, que não são necessariamente únicos no reconhecedor;
2. Conteúdo da fita de entrada: inalterado ou alterado, em relação à configuração inicial, conforme o tipo de reconhecedor;
3. Posição do cursor: apontando para a direita do último símbolo da cadeia de entrada ou apontando para qualquer posição da fita, conforme o tipo de reconhecedor;
4. Conteúdo da memória auxiliar: final e predefinido, não necessariamente único ou idêntico ao da configuração inicial, ou apenas indefinido.



Movimentação

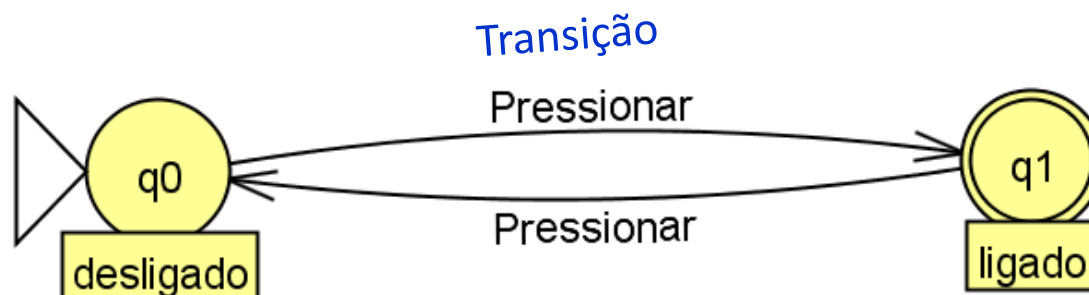
- A especificação de uma possibilidade de movimentação entre uma configuração (estado) e outra é denominada **transição**. A movimentação do reconhecedor da configuração corrente para uma configuração seguinte é feita, portanto, levando-se em conta todas as transições passíveis de serem aplicadas pelo reconhecedor à configuração corrente.

Exemplo de reconhecedor



Movimentação

- Uma transição mapeia operações formadas por:
 - Estado corrente;
 - Símbolo correntemente apontado pelo cursor da fita de entrada;
 - Símbolo correntemente apontado pelo cursor da memória auxiliar;
- em operações formadas por:
 - Próximo estado;
 - Símbolo que substituirá o símbolo correntemente apontado pelo cursor da fita de entrada e o sentido do deslocamento do cursor;
 - Símbolo que substituirá o símbolo correntemente apontado pelo cursor da memória auxiliar.



Linguagem

- Diz-se que um reconhecedor **aceita (ou reconhece) uma cadeia se lhe for** possível **atingir alguma configuração final** a partir de sua configuração inicial única, através de movimentos executados sobre tal cadeia.
- **Caso contrário**, diz-se que o reconhecedor **rejeita a cadeia**.
- A **maneira como tais configurações sucedem umas às outras** durante o reconhecimento (ou aceitação) da cadeia de entrada **define uma característica fundamental dos reconhecedores**, conforme explicado a seguir.
 - Seja o reconhecedor determinístico ou não-determinístico, a linguagem por ele aceita (ou definida) corresponde ao conjunto de todas as cadeias que ele aceita.

Autômatos finitos e linguagens regulares

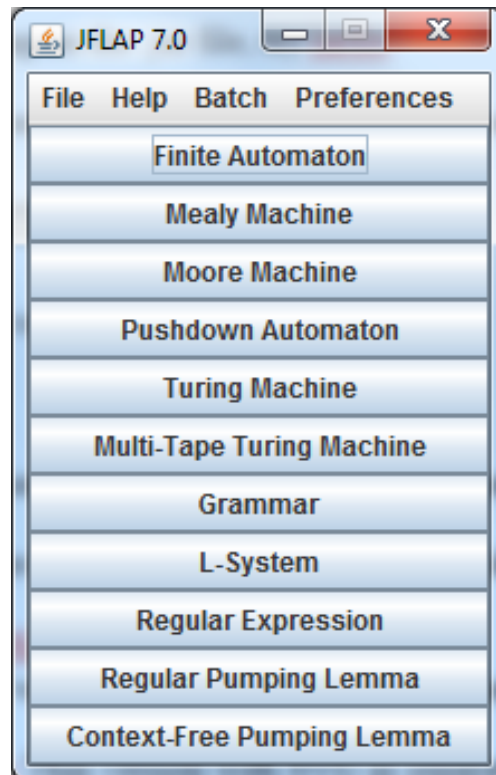
- Da mesma forma como ocorre com as ER's e GR's, os **autômatos finitos** também possibilitam a formalização das linguagens regulares. No entanto, diferentemente daquelas notações, que constituem dispositivos de geração de sentenças, os **autômatos finitos são dispositivos de aceitação de sentenças** e constituem um caso particular do modelo geral de **reconhecedores**

Autômatos finitos - particularidades

• Correspondem à **instância mais simples do modelo geral de reconhecedores** apresentado. As suas principais particularidades em relação ao modelo geral são:

1. **Inexistência de memória auxiliar;**
2. Utilização do **cursor** da fita de entrada **apenas para leitura de símbolos**, não havendo operações de escrita sobre a fita;
3. **Movimentação do cursor** de leitura em apenas um sentido, **da esquerda para a direita;**
4. A **fita** de entrada possui **comprimento limitado**, suficiente apenas para acomodar a cadeia a ser analisada.

Para construir e simular os AF's utilizaremos a ferramenta JFLAP

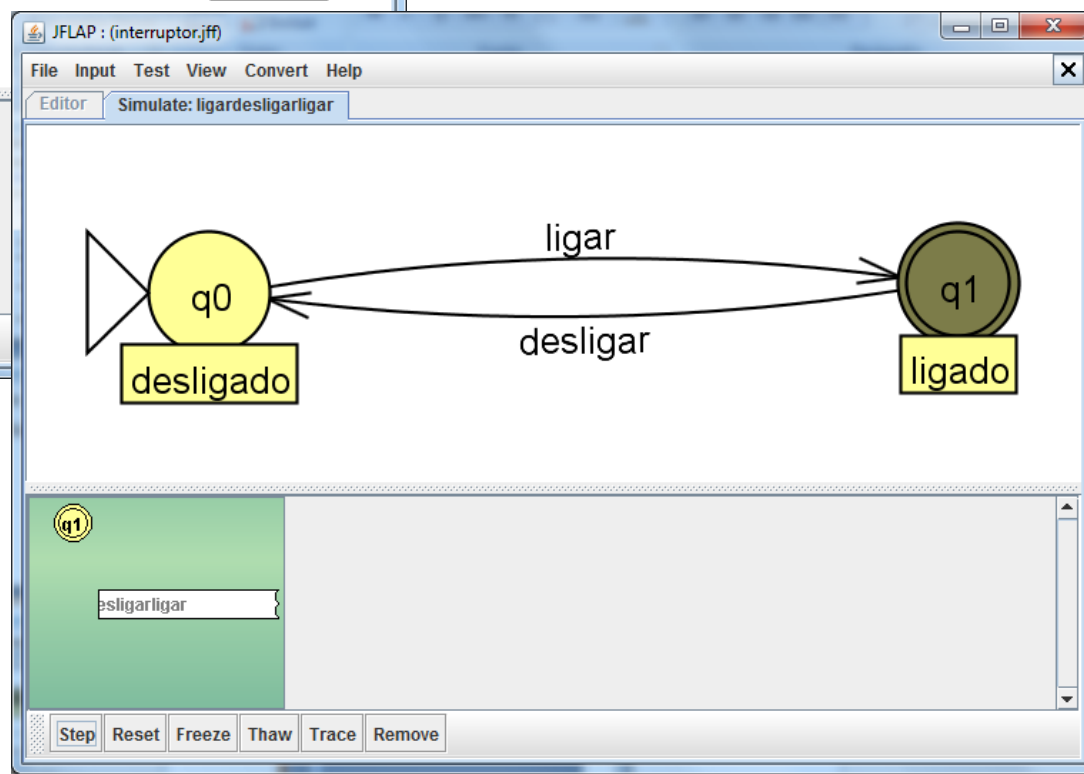
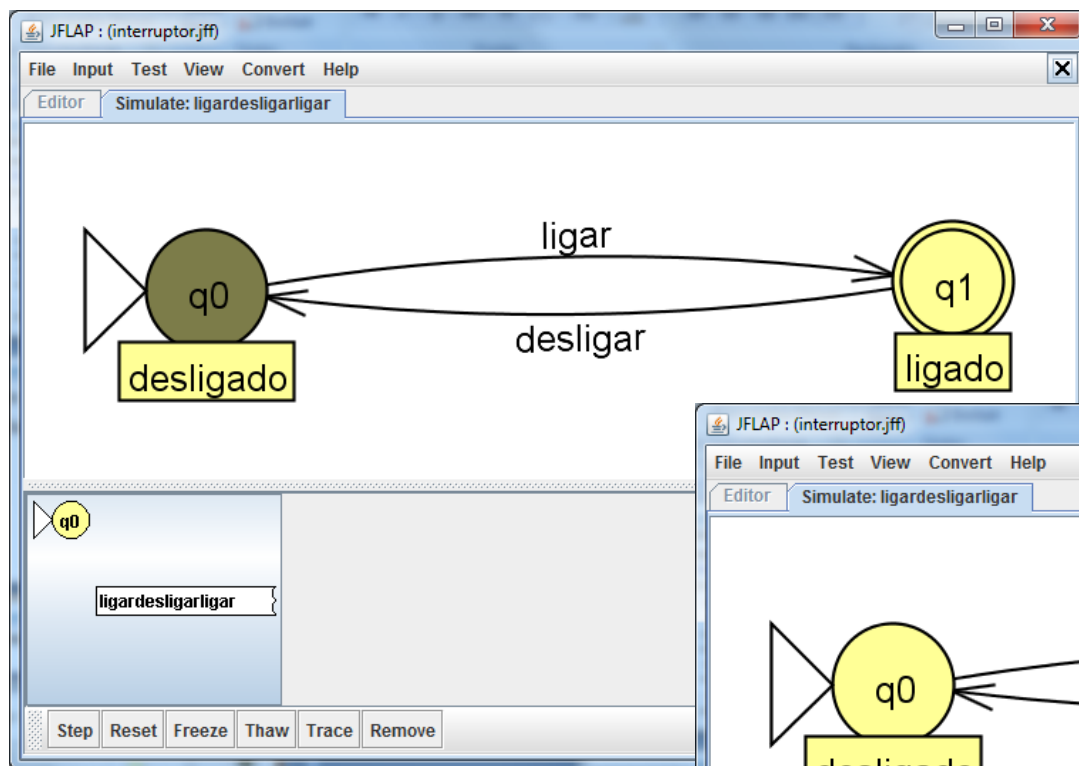


Tutorial JFLAP

<http://pt.scribd.com/doc/75454773/JFLAP>

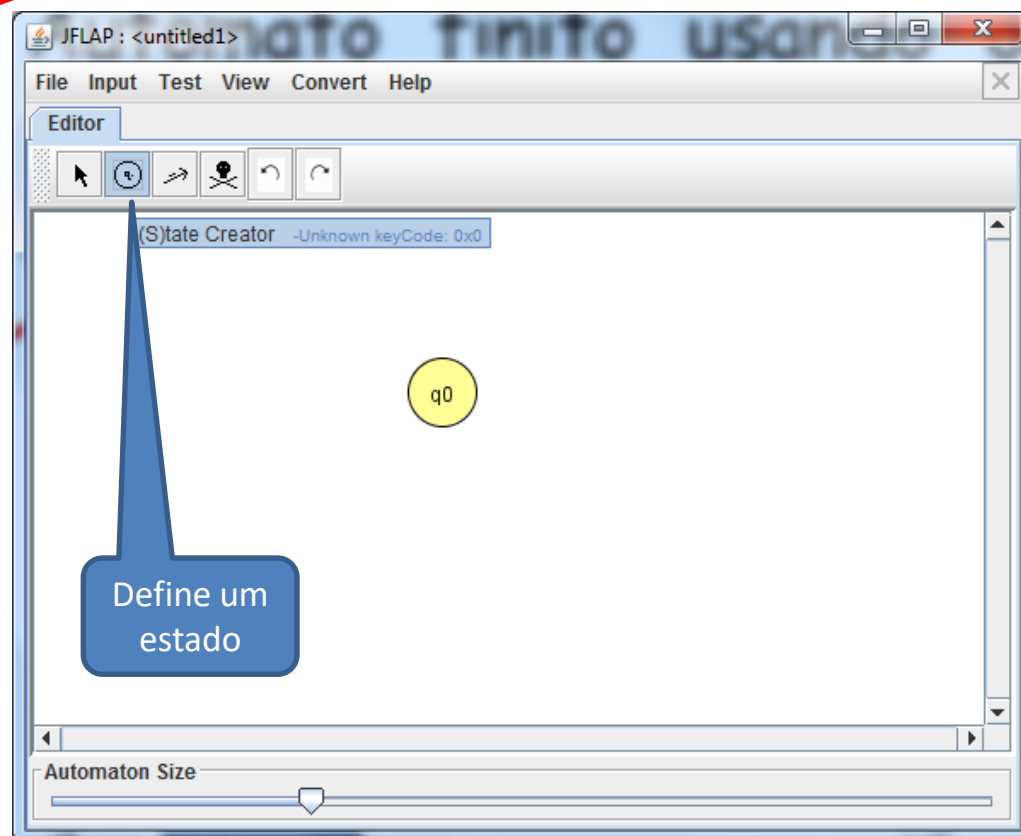
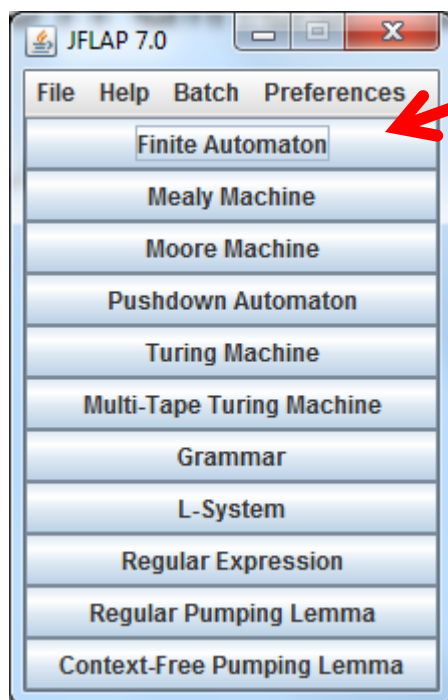
<http://www.jflap.org/>

Exemplo de Autômato finito usando o JFLAP



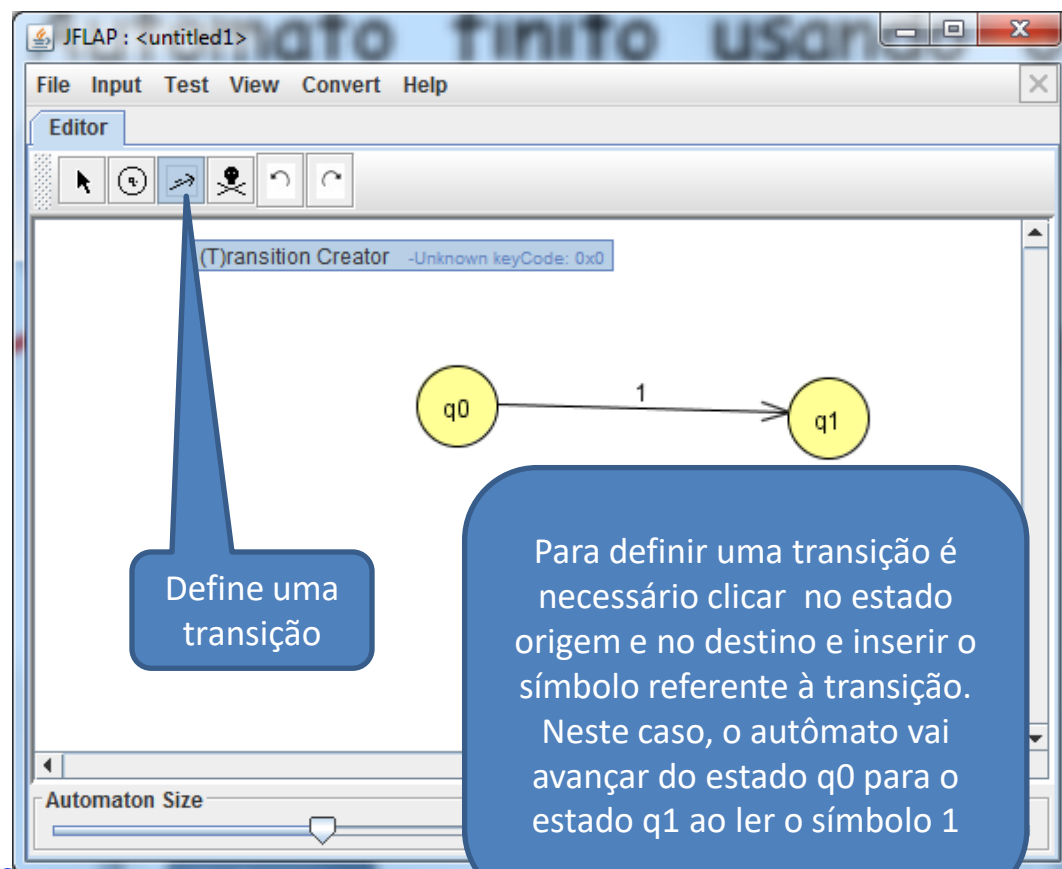
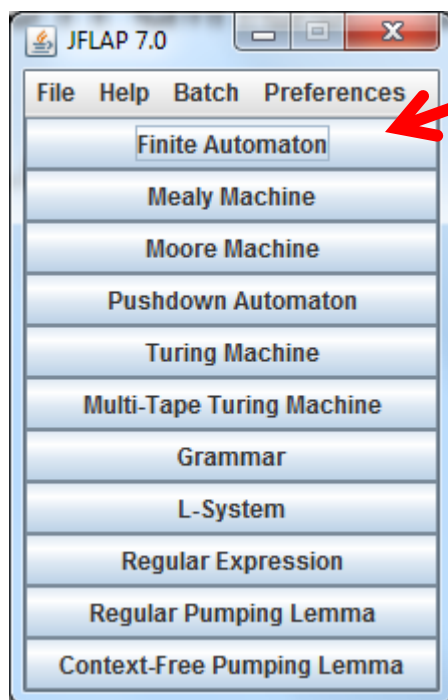
Construindo um Autômato finito usando o JFLAP

Opção
para
construir
AF



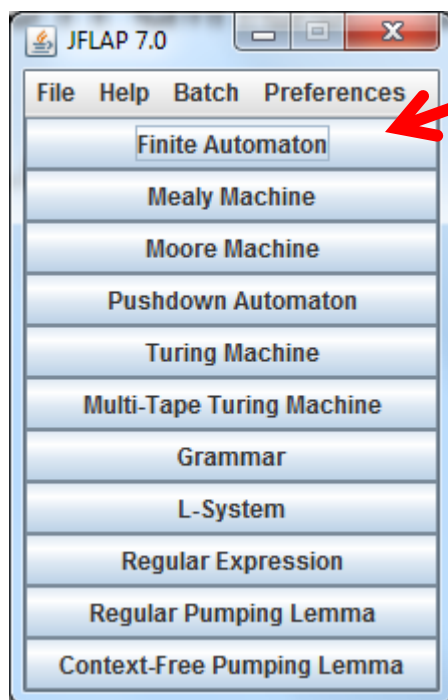
Construindo um Autômato finito usando o JFLAP

Opção
para
construir
AF



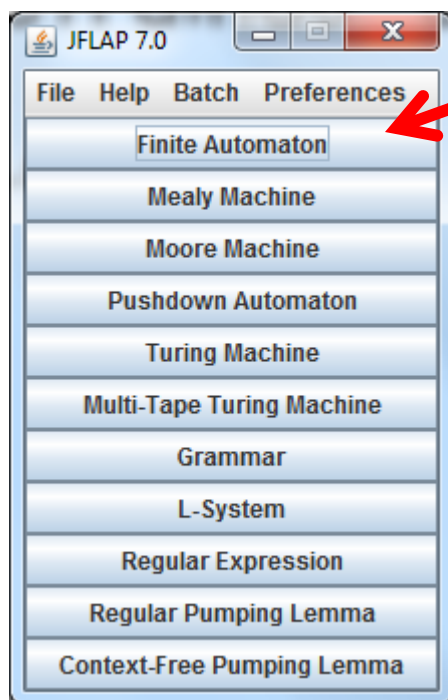
Construindo um Autômato finito usando o JFLAP

Opção
para
construir
AF



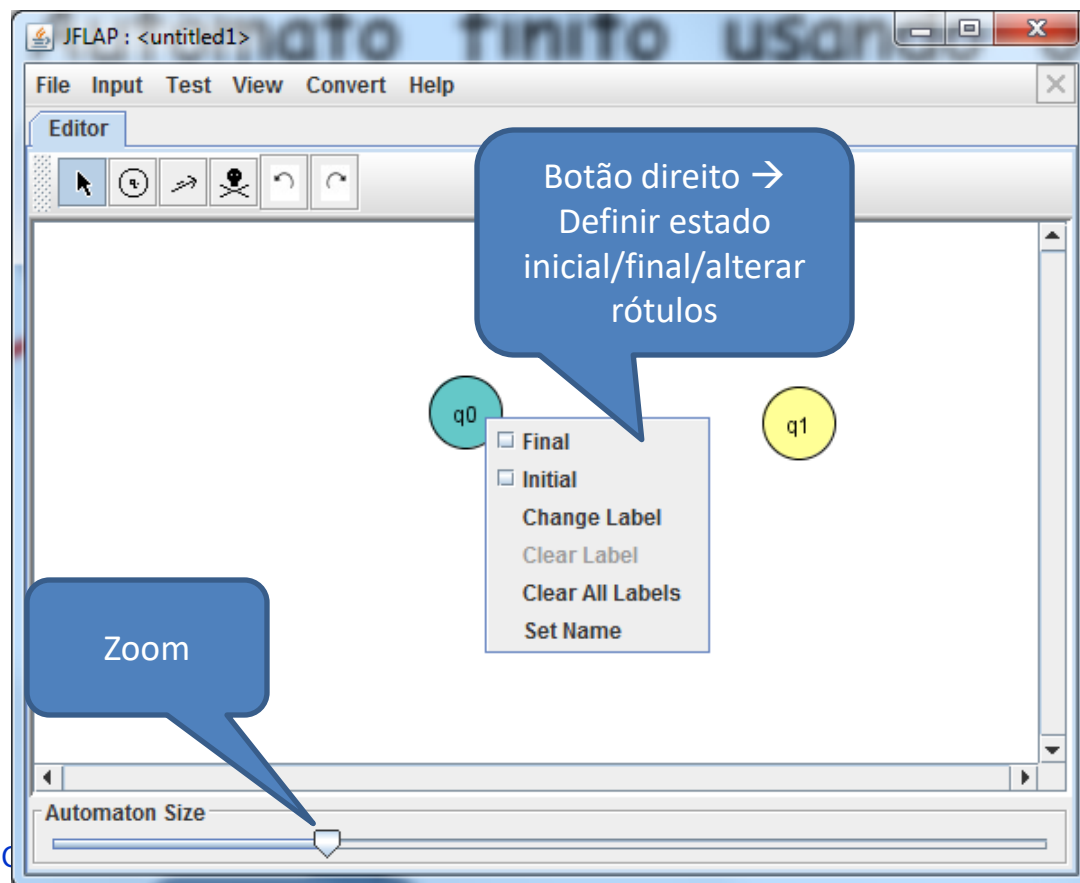
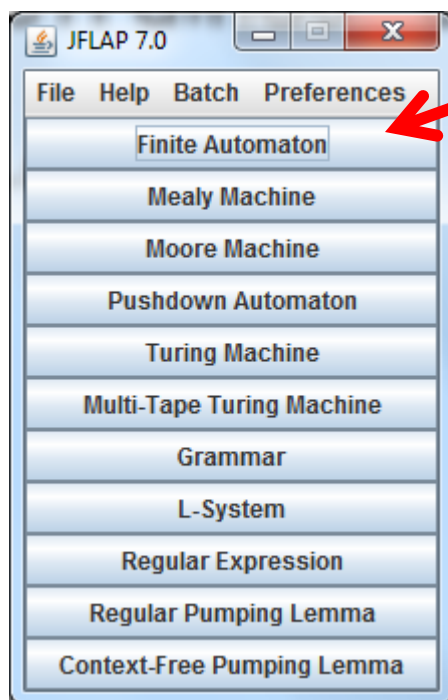
Construindo um Autômato finito usando o JFLAP

Opção
para
construir
AF



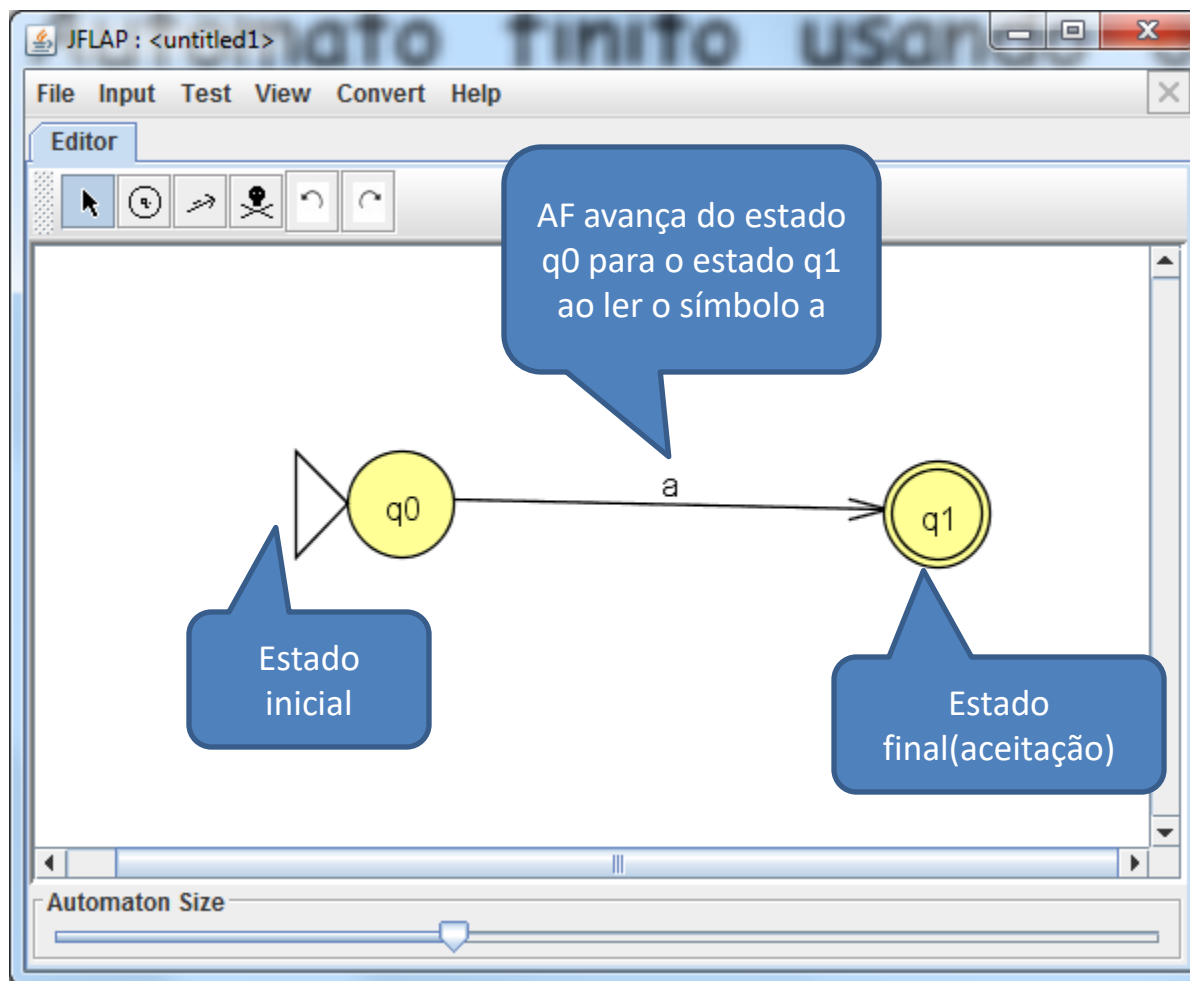
Construindo um Autômato finito usando o JFLAP

Opção
para
construir
AF

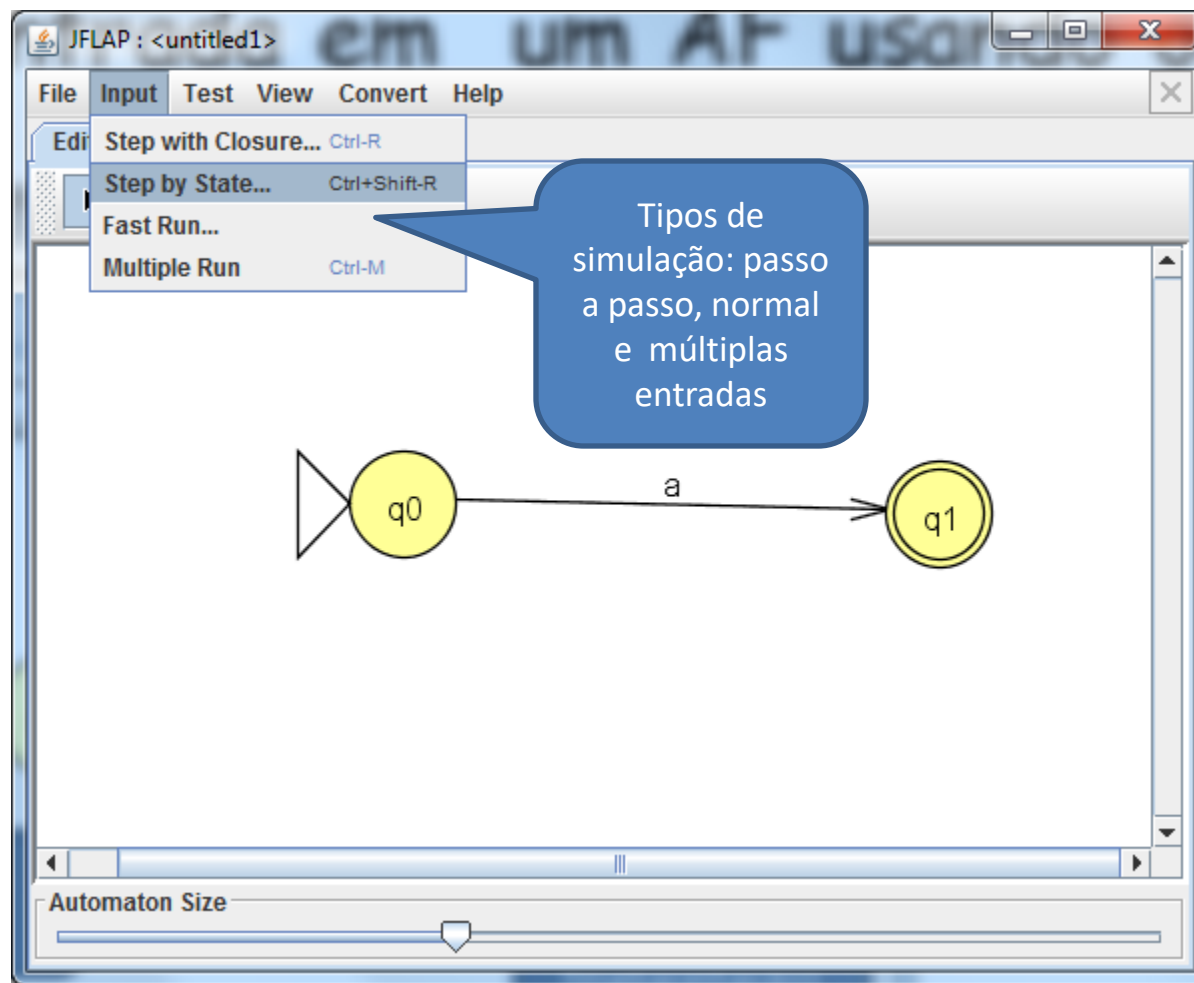


Construindo um Autômato finito usando o JFLAP

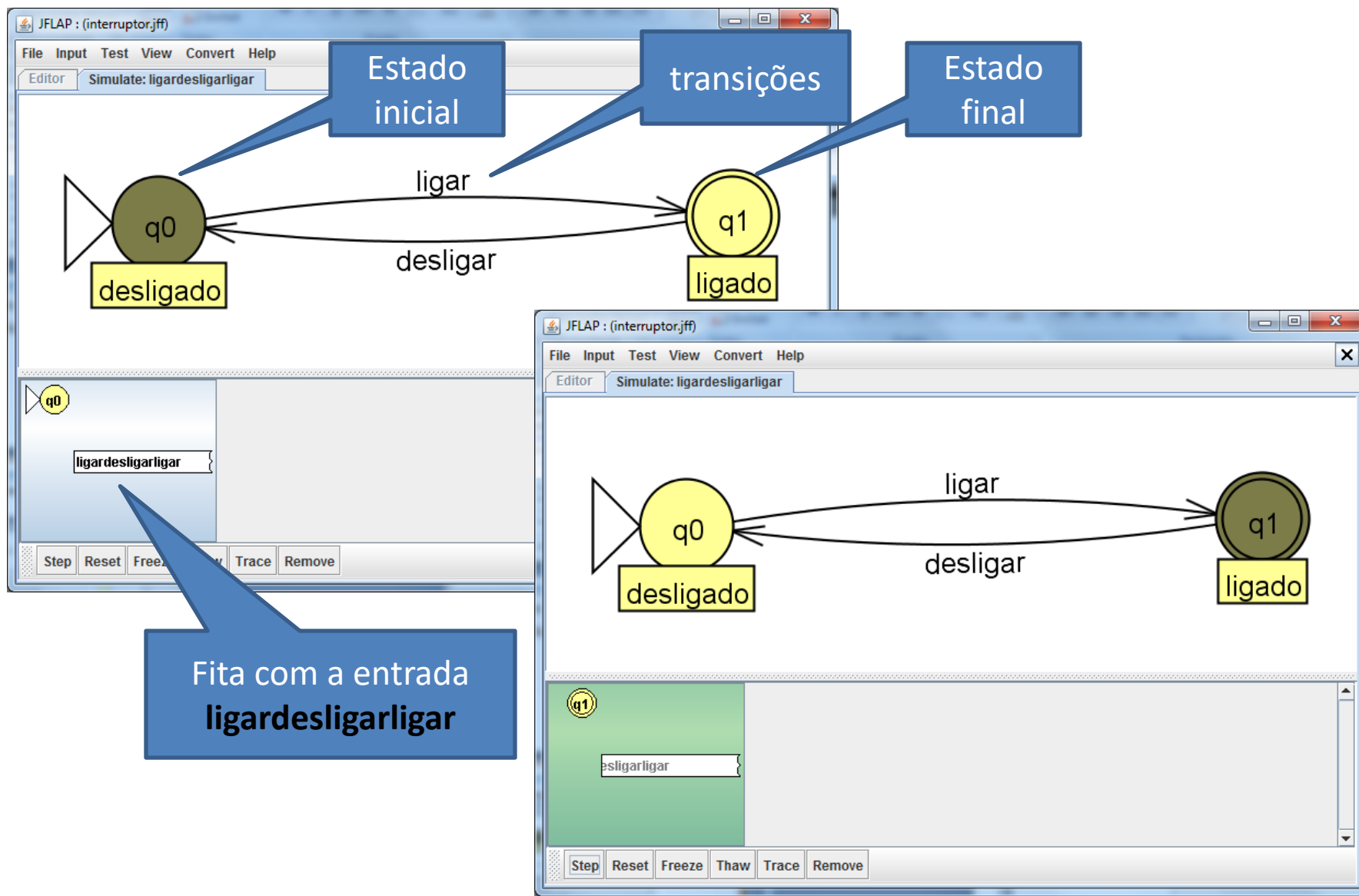
JFLAP



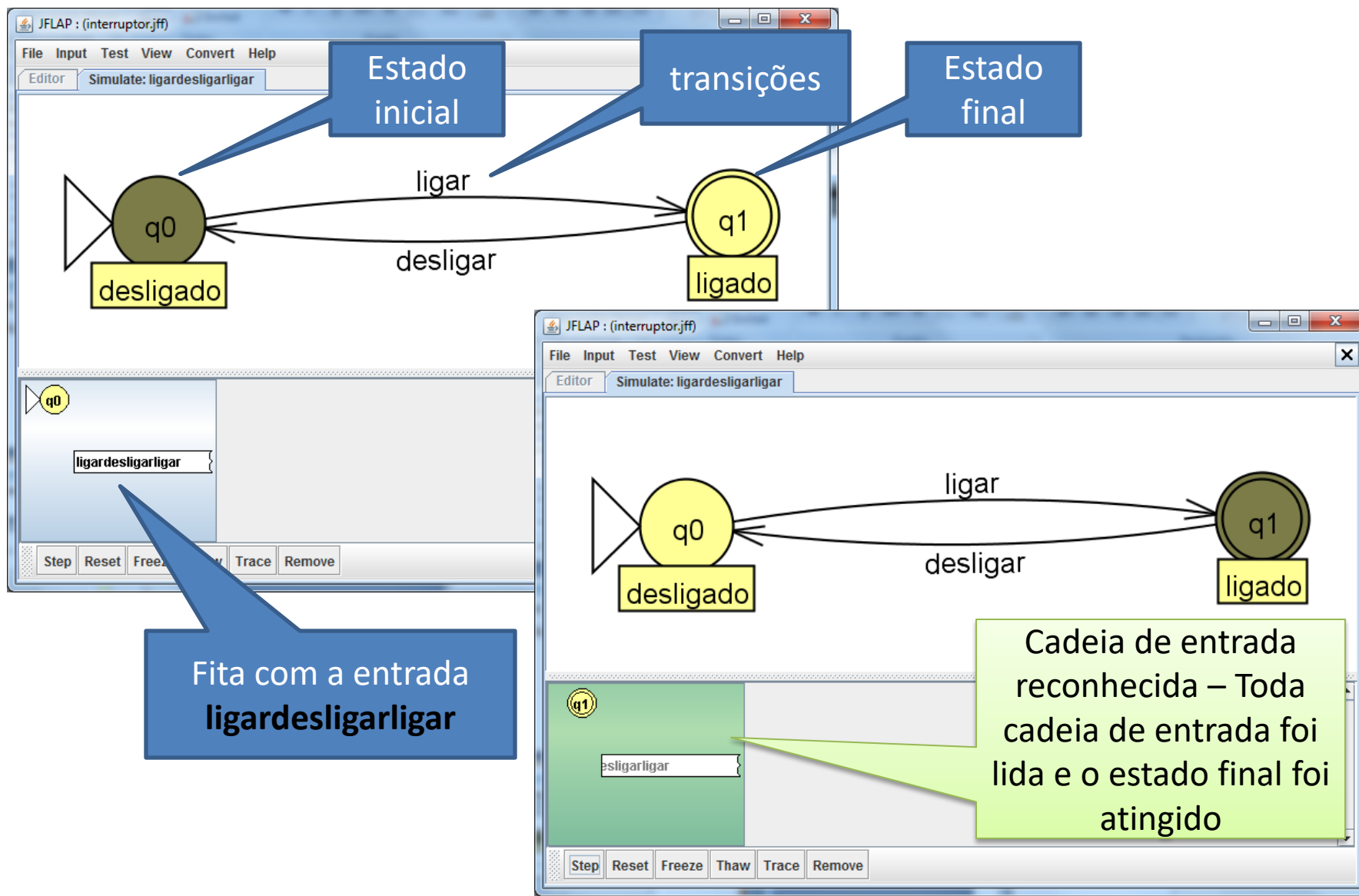
Simulando uma entrada em um AF usando o JFLAP



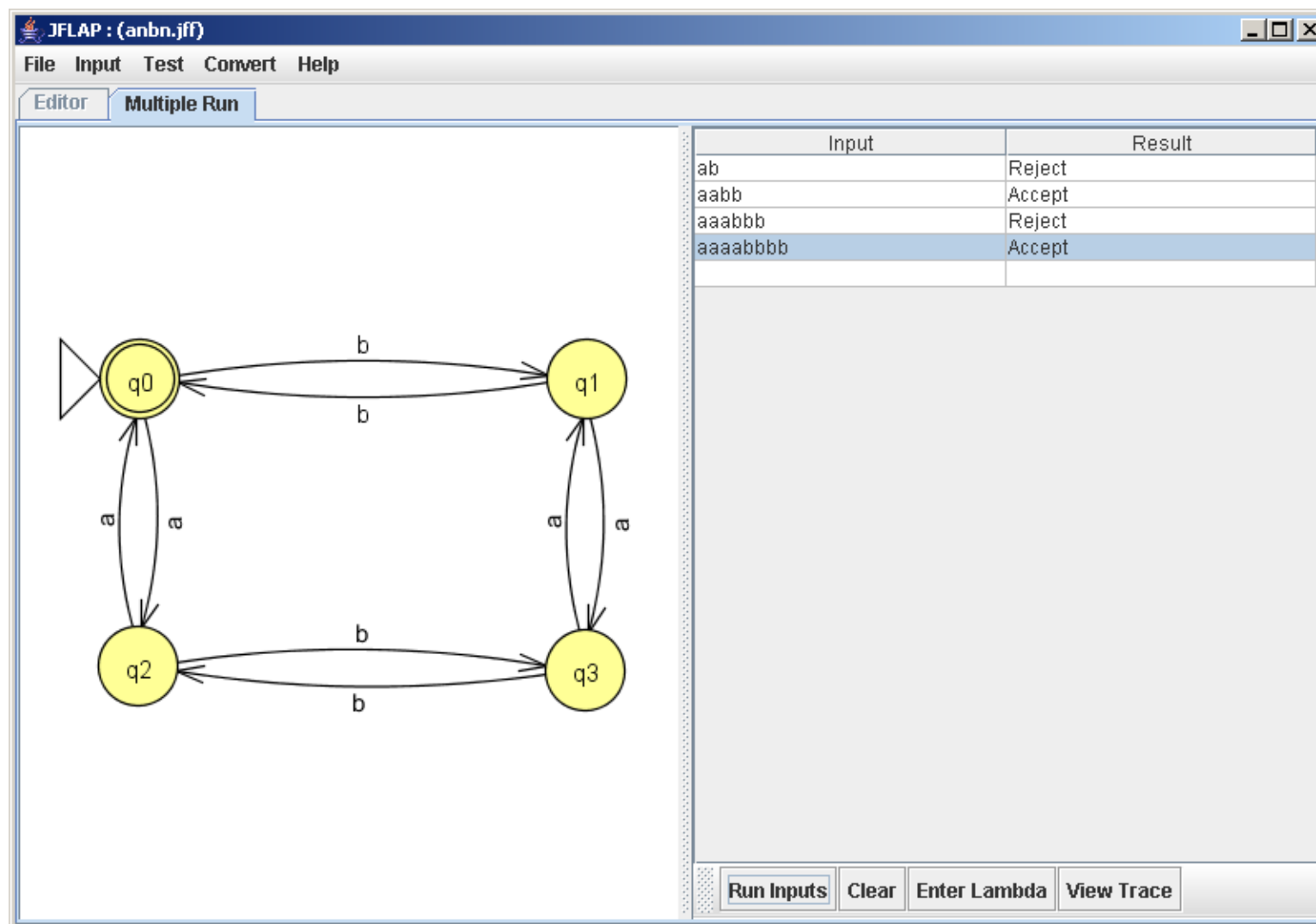
Exemplo de Autômato finito usando o JFLAP



Exemplo de Autômato finito usando o JFLAP



Exemplo 2 - $L = \{w \mid w \text{ tenha o número par de } a \text{ e } b\}$ Ex: aabb aaaabbbb



Exemplo 2 - $L = \{w \mid w \text{ tenha o número par de } a \text{ e } b\}$ Ex: aabb aaaabbbb

JFLAP : (anbn.jff)

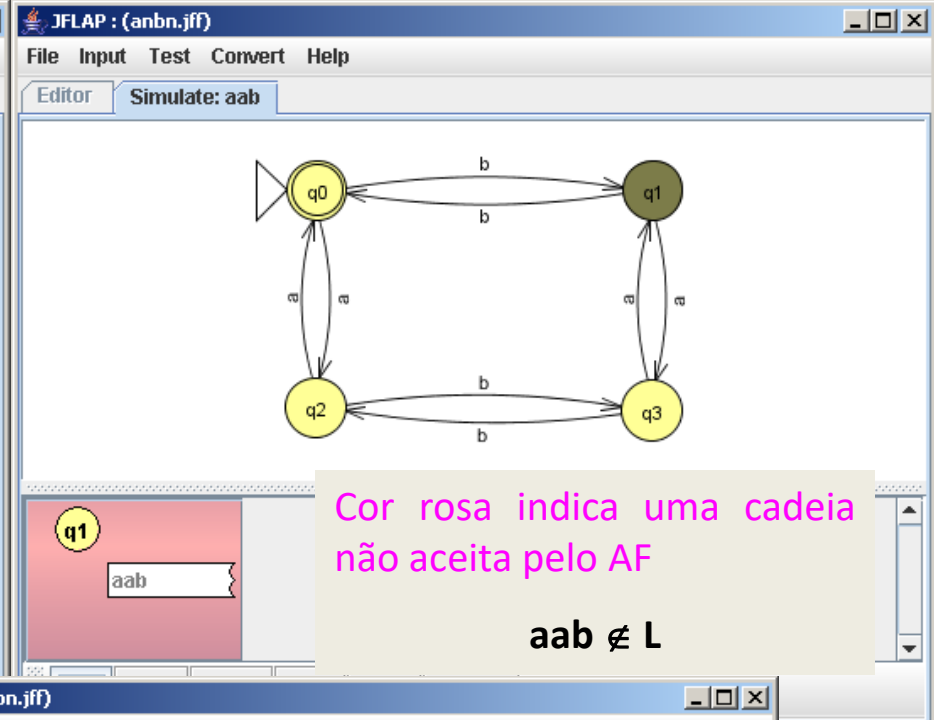
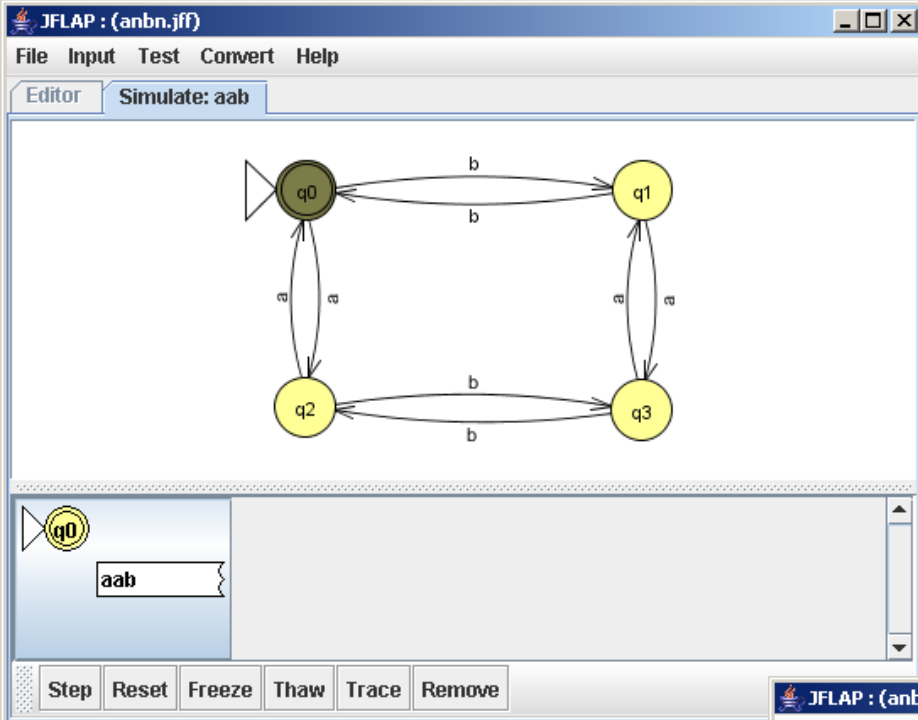
File Input Test Convert Help

Editor Multiple Run

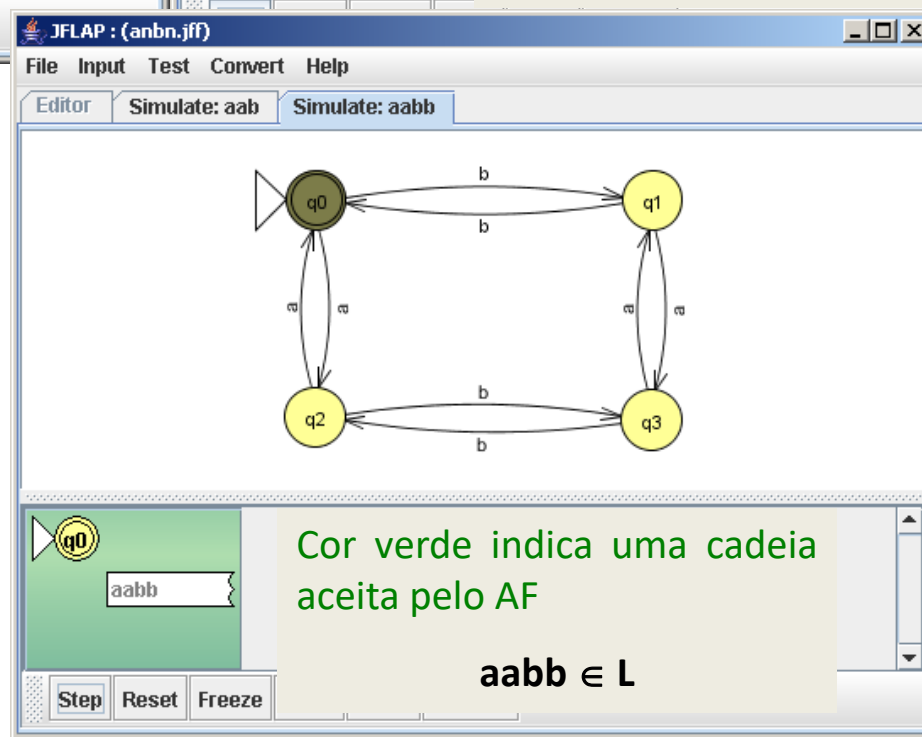
Input	Result
ab	Reject
aabb	Accept
aaaabbb	Reject
aaaabbbb	Accept

Run Inputs Clear Enter Lambda View Trace

Simulando com múltiplas entradas.
JFLAP: Input → Multiple Run



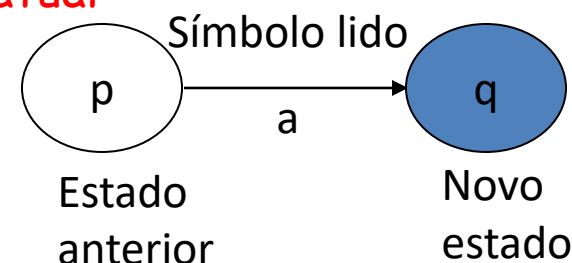
Exemplo 2 - $L = \{w | w \text{ tenha o número par de } a \text{ e } b\}$ Ex: aabb aaaabbbb



Classificação dos AF's

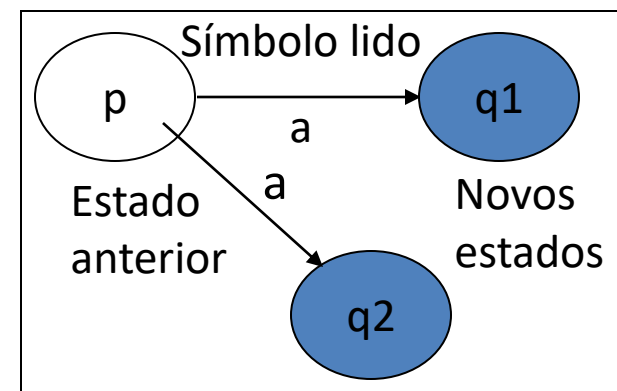
• AF Determinístico (AFD)

- Para cada entrada (símbolo) existe **um e somente um estado** ao qual o autômato pode transitar **a partir de seu estado atual e do símbolo lido**



• AF Não - Determinístico (AFND)

- O autômato tem o poder de estar em vários estados ao mesmo tempo



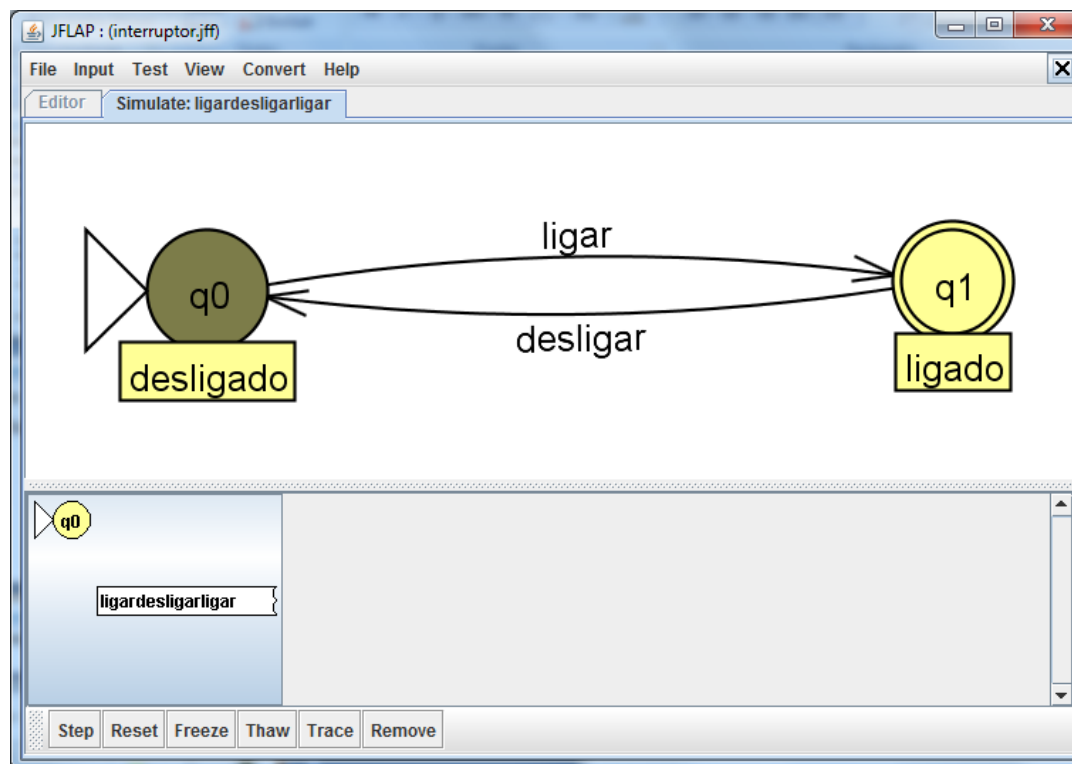
AFD

- Denotado por $A = (\{Q\}, \Sigma, \delta, q_0, \{F\})$
 - Um conjunto finito de estados, frequentemente denotado por Q
 - Um alfabeto de entrada, denotado pelo Σ
 - Uma função transição δ que toma como argumentos um estado e um símbolo de entrada e retorna um novo estado.
 - q_0 - Um estado inicial (pertencente a Q)
 - Um conjunto de estados finais ou de aceitação F (subconjunto de Q)

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

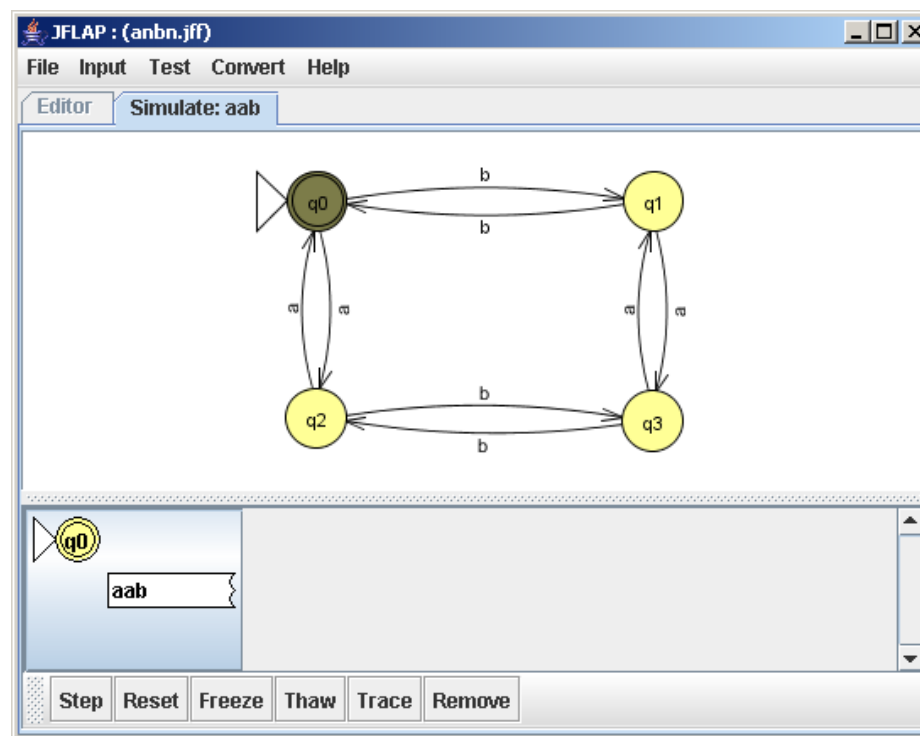
Baseado no exemplo do interruptor

$$A = (\{q_0, q_1\}, \{\text{ligar}, \text{desligar}\}, \delta, q_0, \{q_1\})$$



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

$$A = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_0\})$$

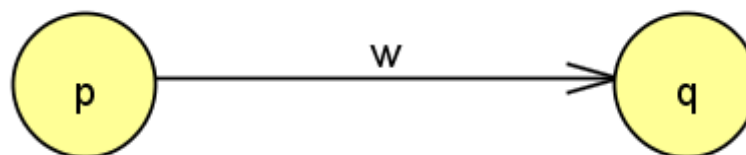


AFD - Função de Transição de Estados

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

$\delta(p, w) = q \rightarrow$ no estado p ao ler o símbolo w , o AFD atingirá o estado q .

- Isto é, existe um caminho no diagrama de transições de p para q denominado w



AFD - Linguagem Aceita

- Uma cadeia **x** é dita ser aceita pelo AFD $A = (Q, \Sigma, \delta, q_0, F)$ se $\delta(p_i, x) = q$ para algum $q \in F$.

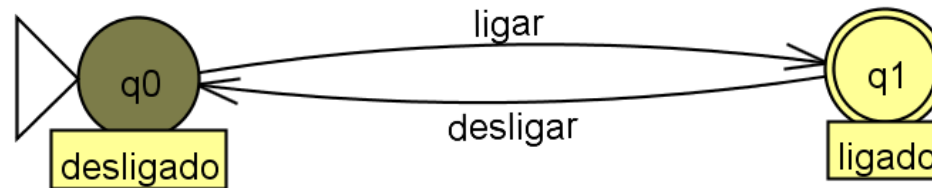
Ou $L(M) = \{x \mid \delta(p_i, x) \in F\}$

- **Definição 1:** Uma linguagem aceita por um AFD é uma linguagem regular (ou do tipo 3)
- **Definição 2:** Dois AFD A_1 e A_2 são equivalentes se $L(\text{AFD1}) = L(\text{AFD2})$

AFD - Formas de representação

1. Diagramas de Transições

Grafo



AFD - Formas de representação

2. Tabelas de Transições

Qual o diagrama de transições correspondente?

Estado Inicial,
representado por



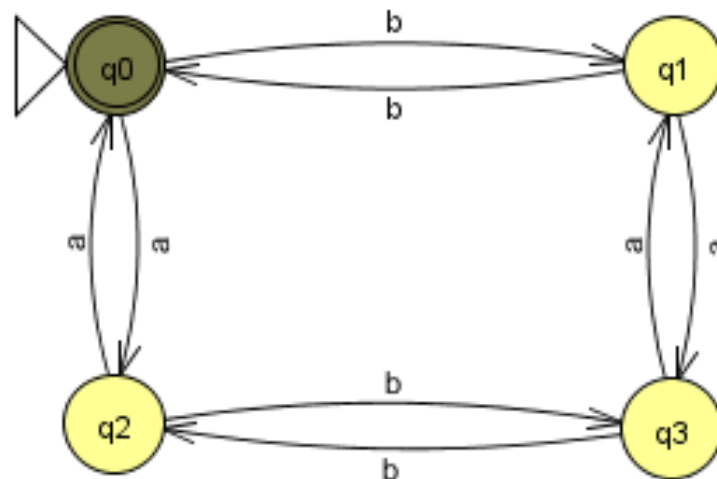
Estado de
aceitação,
representado por
*

Símbolos de entrada

	0	1
→q0	q2	q1
*q1	q1	q1
q2	q2	q1

Conjunto de Estados

Reconhecimento de bbaa



$$A = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_0\})$$

Tabela de Transição de Estados

	a	b
$\rightarrow^* q_0$	q2	q1
q1	q3	q0
q2	q0	q3
q3	q1	q2

Função de Transição de Estados

$$\delta(q_0, a) = q_2 \quad \delta(q_0, b) = q_1$$

$$\delta(q_1, a) = q_3 \quad \delta(q_1, b) = q_0$$

$$\delta(q_2, a) = q_0 \quad \delta(q_2, b) = q_3$$

$$\delta(q_3, a) = q_1 \quad \delta(q_3, b) = q_2$$

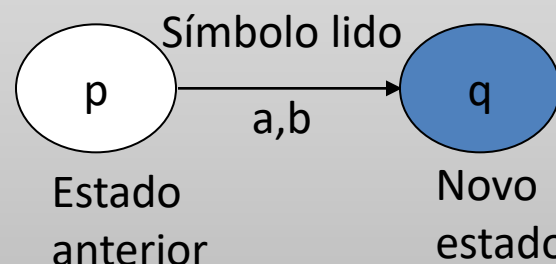
Exercícios

1. Construir AFD's (**determinísticos**) que reconhecem as linguagens sobre o alfabeto $\{a,b\}$ e cujas sentenças estão descritas a seguir:

- Começam com aa;
- Não começam com aa;
- Terminam com bbb;
- Não terminam com bbb;
- Contém a subcadeia aabbb;
- Possuem comprimento maior ou igual a 3;
- Possuem comprimento menor ou igual a 3;
- Possuem comprimento diferente de 3;
- Possuem comprimento par;
- Possuem comprimento ímpar;
- Possuem comprimento múltiplo de 4;
- Possuem quantidade par de símbolos a;
- Possuem quantidade ímpar de símbolos b.

Relembrando AFD:

UM E SOMENTE um estado a seguir a partir de seu estado atual e do símbolo lido



Exercícios

2. Construa ER's, GR's e AFD's, na forma de diagrama de estados e tabela de transições, que reconhecem as seguintes linguagens

$L1 = \{x \in \{0,1\}^* \mid \text{número de } 1's \text{ em } x \text{ é múltiplo de } 3\}$

$L2 = \{x \in \{0,1\}^* \mid x \text{ contém a subcadeia } 001\}$

$L3 = \text{um AFD que reconhece a formação de identificadores em Pascal } (\underline{L}D), \text{ onde: } \underline{} \text{ representa underline; } L [a-z] \text{ e } D = [0-9]$

$L4 = \text{um AFD que reconhece a formação de números inteiros em Pascal } (0\dots 9)$

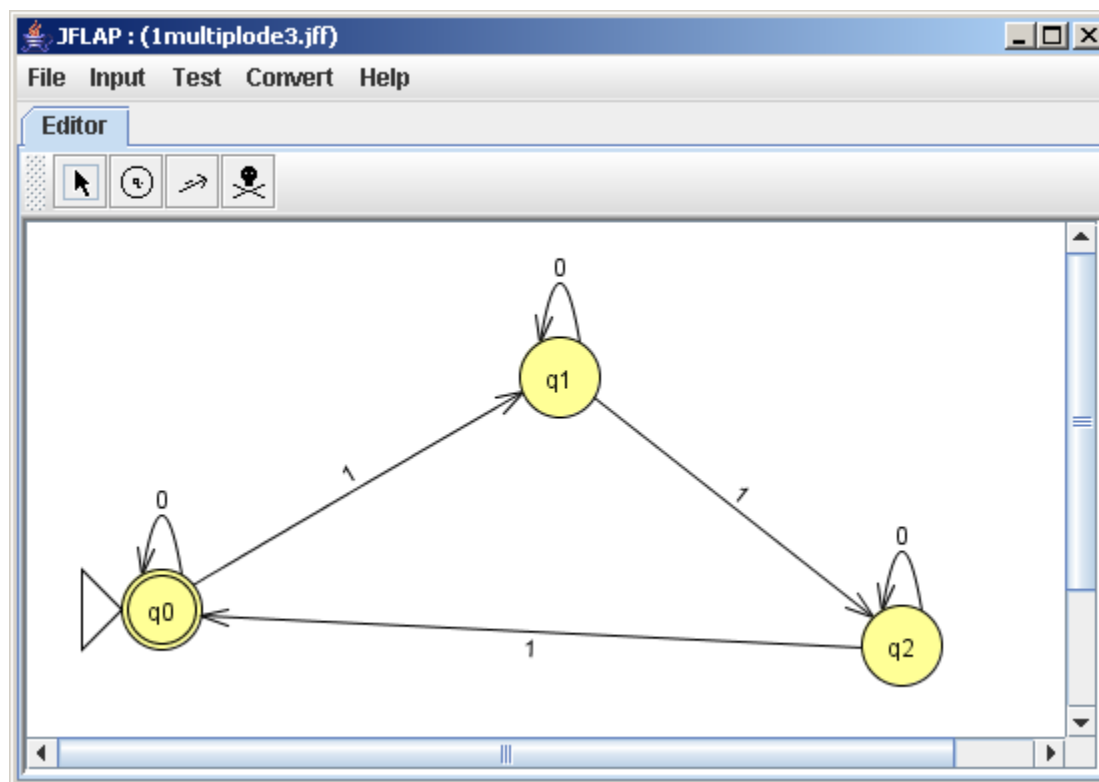
$L5 = \text{um AFD que reconhece a formação de Operadores relacionais em Pascal } (<, <=, =, >, >=)$



Resolução dos AFD's referentes ao exercício 2

$L1 = \{x \in \{0,1\}^* \mid \text{número de 1's em } x \text{ é múltiplo de } 3\}$

$A = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_0\})$

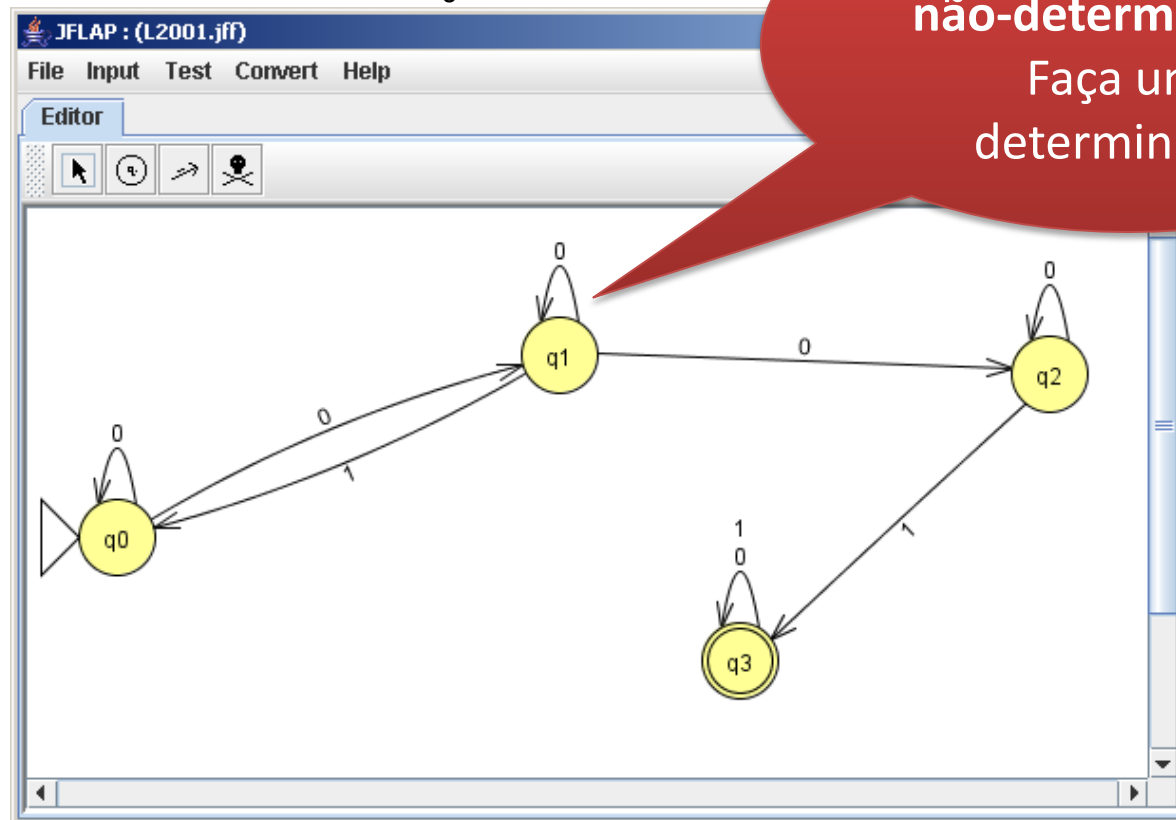


Aceita: 000111 0111111 111111111

Rejeita: 011 00001111 00001

$L2 = \{x \in \{0,1\}^* \mid x \text{ contém a subcadeia } 001\}$

$A = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$

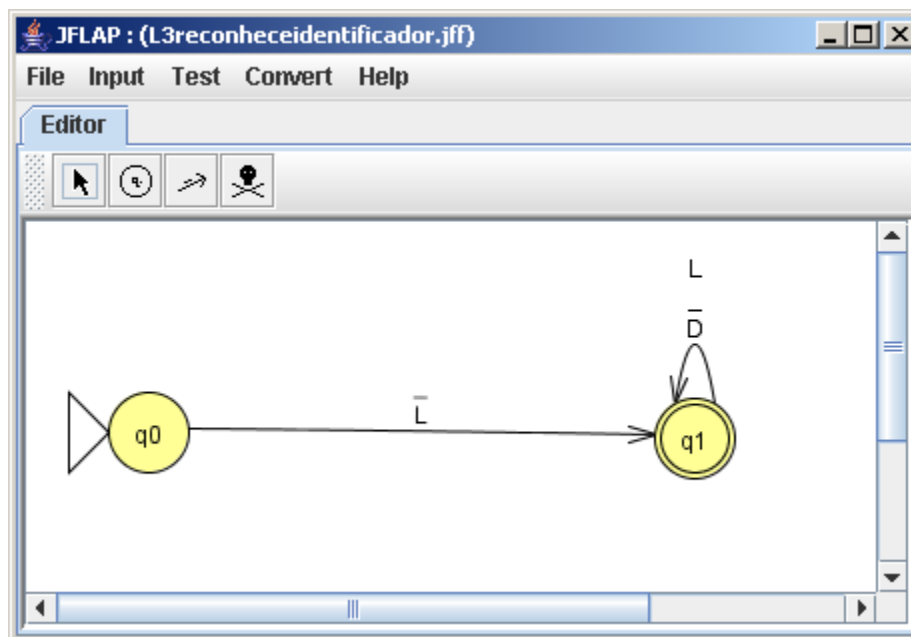


Aceita: 00111 1001111 11100111111

Rejeita: 011 01111 000

L3 = um AFD que reconhece a formação de identificadores em Pascal

$A = (\{q_0, q_1\}, \{A..Z, a..z, 0..9\}, \delta, q_0, \{q_1\})$



L é o conjunto das letras

D é o conjunto dos dígitos

_ é o sublinhado

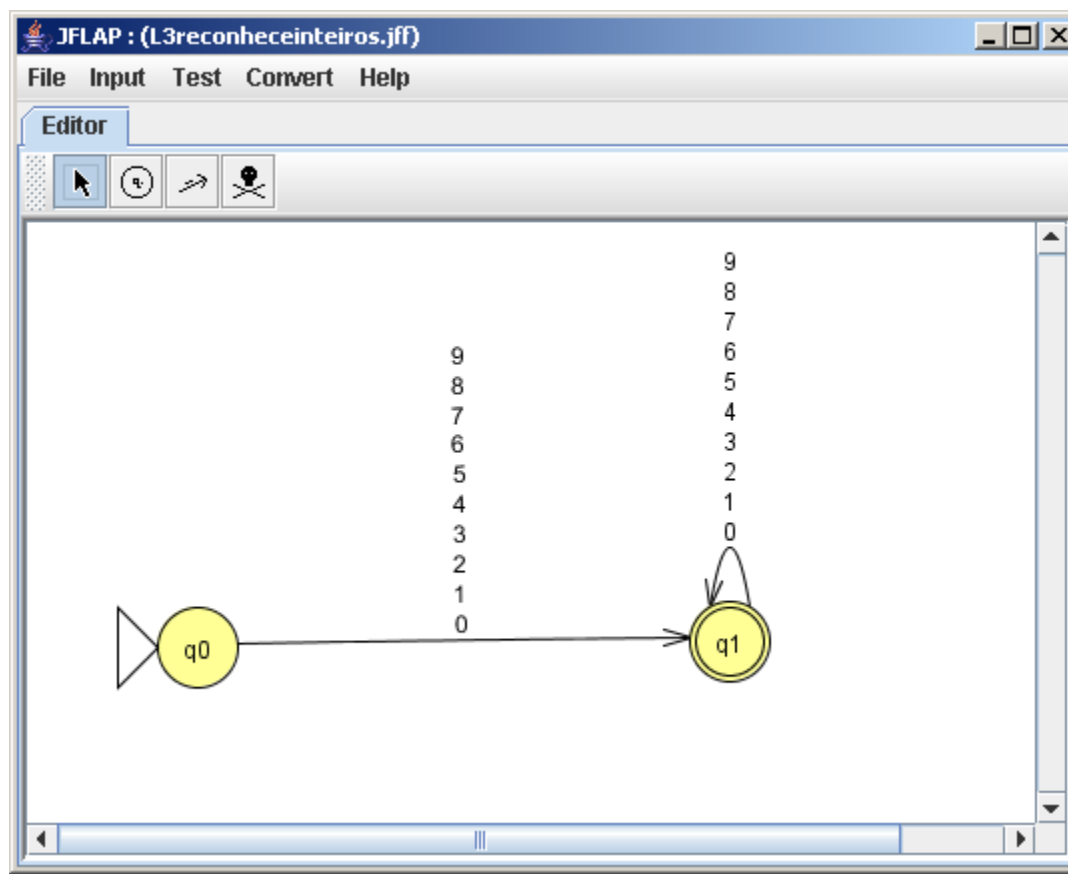
$_LLL \rightarrow _xyz$

$LDD \rightarrow A123$

$L_D \rightarrow x_2$

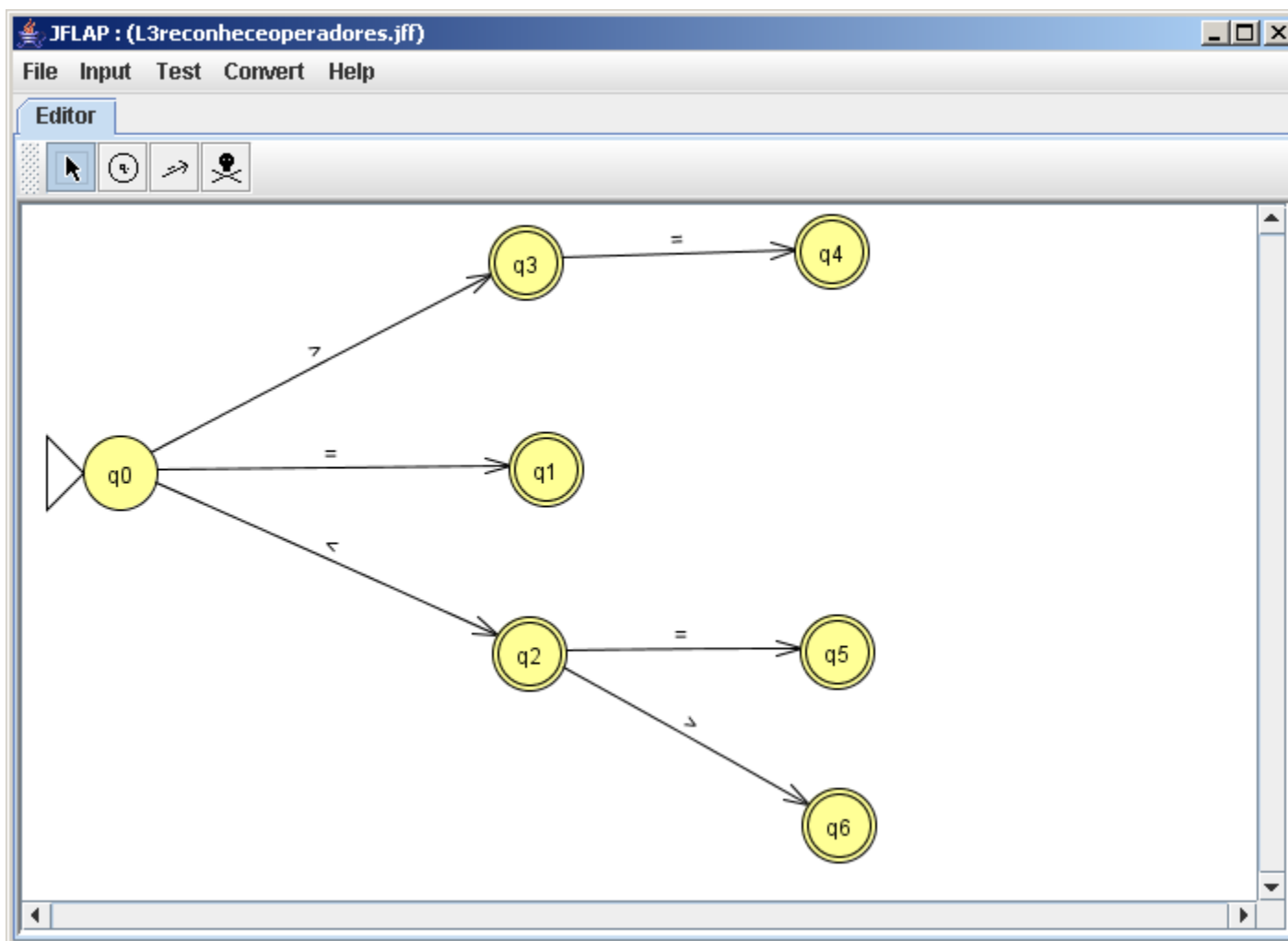
L4 = um AFD que reconhece a formação de números inteiros em Pascal

$A = (\{q_0, q_1\}, \{0, \dots, 9\}, \delta, q_0, \{q_1\})$



L5 = um AFD que reconhece a formação de operadores relacionais em Pascal

$A = (\{q_0, q_1\}, \{<, <=>, <=, =, >, >= \}, \delta, q_0, \{q_1, q_2, q_3, q_4, q_5, q_6\})$





Na próxima aula...

- AFND