Aspectos Teóricos da Computação

Prof. Rodrigo Martins
rodrigo.martins@francomontoro.com.br

Cronograma da Aula

- Gramáticas Regulares: dispositivos geradores das Linguagens Regulares
- Expressões Regulares
- Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares
- Exemplos no Jflap
- Exercícios

Gramáticas Regulares: dispositivos geradores das Linguagens Regulares

- As Linguagens regulares são geradas pelas Gramáticas Regulares.

- Seja G = (V, ∑, P, S) uma gramática e sejam A e B símbolos não terminais e w uma cadeia de ∑*.

a) G é uma gramática linear à direita, se todas as produções são da forma:

  - $A \rightarrow wB$ ou $A \rightarrow w$

b) G é uma gramática linear à esquerda, se todas as produções são da forma:

  - $A \rightarrow Bw$ ou $A \rightarrow w$

Gramáticas Regulares: dispositivos geradores das Linguagens Regulares

- Uma Gramática Regular é qualquer Gramática Linear.

- **Exemplo:** A gramática $\mathbf{G_1}$ **é Linear à direita.**

$G_1 = (V, \sum, P, S)$, onde:

$V = \{S, T\}$

$\sum = \{x, y\}$

$P = \{ S \rightarrow xyT$

$\quad T \rightarrow xyT \mid xy\}$

Gramáticas Regulares: dispositivos geradores das Linguagens Regulares

- **Exemplo:** A gramática **G_2 é Linear à esquerda.**

$G_2 = (V, \sum, P, S)$, onde:

$V = \{S\}$

$\sum = \{x, y\}$

$P = \{ S \rightarrow Sxy \mid xy\}$

S é o símbolo inicial.

Gramáticas Regulares: dispositivos geradores das Linguagens Regulares

- Os seguintes teoremas mostram que a Classe das Gramáticas Regulares denota exatamente a Classe das Linguagens Regulares.

- **Teorema 1**: Se L é uma linguagem gerada por uma Gramática Regular, então L é uma Linguagem Regular.

- **Teorema 2**: Se L é uma Linguagem Regular, então existe G, Gramática Regular que gera L.

Gramáticas Regulares: dispositivos geradores das Linguagens Regulares

- Exemplo: Considere-se a **Gramática G_3 Linear à Direita**:

$G_3 = (V, \sum, P, S)$, onde:

$V = \{S, A, B\}$

$\sum = \{x, y\}$

$P = \{ S \rightarrow xA$

$A \rightarrow yB$

$B \rightarrow xA \mid \varepsilon \}$

S é o símbolo inicial.

$S \rightarrow xA \rightarrow xyB \rightarrow xy\varepsilon = xy$

$S \rightarrow xA \rightarrow xyB \rightarrow xyxA \rightarrow xyxyB \rightarrow xyxy\varepsilon = xyxy$

$S \rightarrow xA \rightarrow xyB \rightarrow xyxA \rightarrow xyxyB \rightarrow xyxyxA \rightarrow xyxyxyB \rightarrow xyxyxy\varepsilon = xyxyxy$

Uma vez que G3 é regular, a linguagem L(G3) é regular.

Esta gramática gera a seguinte linguagem.

$L(G_3) = \{xy, xyxy, xyxyxy, xyxyxyxy, ...\}$

Gramáticas Regulares: dispositivos geradores das Linguagens Regulares

- Exemplo: Considere-se a **Gramática G_4 Linear à Esquerda**:

G_4 = (V, ∑, P, S), onde:

V = {S, A, B }

∑ = {x, y}

P = { S → Ay

A → Bx

B → Ay | ε}

S é o símbolo inicial.

S → Ay → Bxy → εxy = xy
S → Ay → Bxy → Ayxy → Bxyxy → εxyxy = xyxy
S → Ay → Bxy → Ayxy → Bxyxy → Ayxyxy → Bxyxyxy → εxyxyxy = xyxyxy

A gramática G_4 é equivalente a G_3 e também gera a linguagem regular L(G_4) = {xy, xyxy, xyxyxy,xyxyxyxy,...}

Expressões Regulares

- As expressões regulares, notação desenvolvida por Kleene na década de 50, constituem-se em uma alternativa para representar as linguagens regulares.
- Uma expressão regular sobre um alfabeto $\sum$ é indutivamente definida como se segue:

a) O conjunto vazio $\emptyset$, que é uma Linguagem vazia, é uma expressão regular;

b) A cadeia vazia ε é uma expressão regular e portanto, a linguagem L = $\{\varepsilon\}$ é regular.

c) Qualquer símbolo $x \in \sum$ é uma expressão regular e portanto, a linguagem L = {x} é regular.

d) Se r e s são expressões regulares e consequentemente as linguagens R e S são regulares, então:

1. (r | s) é uma expressão regular e a linguagem R U S é regular.
2. (rs) é uma expressão regular e a linguagem RS = $\{xy \mid x \in R \text{ e } y \in S\}$ é regular.
3. (r*) é uma expressão regular.

Expressões Regulares

Exemplo 1: A expressão regular x* representa a linguagem:

$L_1 = \{\varepsilon, x, xx, xxx, ...\}$

Exemplo 2: A expressão regular (x|y*) representa a linguagem :

$L_2 = \{\varepsilon, x, y, yy, yyy, ...\}$

Exemplo 3: A expressão regular (xy*) representa a linguagem:

$L_3 = \{\varepsilon, x, xy, xyy, xyyy, xyyyy, ...\}$

Exemplo 4: A expressão regular (x | xy+) representa a linguagem:

$L_4 = \{x, xy, xyy, xyyy ...\}$

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

- Os autômatos finitos são formalismos de aceitação das sentenças das Linguagens Regulares, ou seja, o autômato finito aceita toda e qualquer cadeia pertencente à linguagem para o qual foi projetado e rejeita todas as cadeias não pertencentes à mesma.

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

- Os autômatos finitos apresentam os seguintes componentes:
  - **Fita de entrada.** Trata-se de um dispositivo de armazenamento, uma memória, que contém a cadeia a ser analisada pelo reconhecedor. A fita é finita, dividida em células, e cada célula armazena um símbolo da cadeia de entrada. O comprimento da fita de entrada, portanto, é igual ao comprimento da cadeia de entrada. A leitura dos símbolos gravados na fita de entrada é efetuada mediante o uso de um cursor, o qual sempre aponta o próximo símbolo da cadeia a ser processado. Não há operações de escrita sobre a fita. O cursor movimenta-se exclusivamente da esquerda para a direita.

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

- **Unidade de controle finito ou máquina de estados:** Trata-se de um controlador central do reconhecedor. A unidade de controle dispõe da especificação de movimentações possíveis do cursor, descritas por um conjunto finito de estados e transições. O conceito de estado diz respeito ao registro de informações capturadas no passado e relevantes para o posterior processamento da cadeia de entrada. Inicialmente, o cursor aponta para o símbolo mais à esquerda da cadeia. Nesta configuração, em que a cadeia completa ainda será analisada, o controlador se encontra no estado inicial, que deve ser único. Uma transição em um autômato finito é definida pela tripla (estado corrente, símbolo corrente, próximo estado).

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

- O símbolo pode ser cadeia vazia ε, ou qualquer elemento do alfabeto da cadeia de entrada sobre o qual Linguagem é definida.

- Os autômatos finitos podem ser **determinísticos** ou **não determinísticos**.

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

- Um Autômato Finito Determinístico (AFD) ou simplesmente um Autômato Finito (AF) M é uma quíntupla:

$M = (Q, \sum, g, q0, F)$

Q é um conjunto finito de estados;

$\sum$ é um alfabeto (finito e não-vazio) de entrada;

g é uma função de transição;

q0 é o estado inicial;

F é um conjunto de estados finais.

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

- Seja M um autômato finito determinístico (Q, ∑, g, q0, F), onde:

Q = {q0, q1, q2}

∑ = {a, b, c}

q0 é o estado inicial

F = {q2}

e a função g pode ser apresentada como a tabela abaixo:

g	x	y	z
q0	q1	-	-
q1	-	q2	-
q2	-	-	q2

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

- Pode-se também denotar a função g, como:

g= {((q0,x), q1), ((q1,y), q2), ((q2, z), q2)

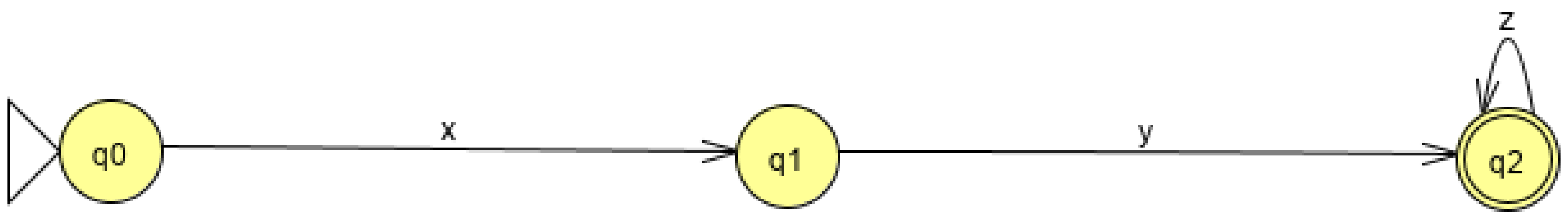
- Ainda, a função g, pode ser especificada como:

g(q0, x) = q1; g(q1, y) = q2; g(q2, z) = q2.

g	x	y	z
q0	q1	-	-
q1	-	q2	-
q2	-	-	q2

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

- M pode ser representado como na figura seguinte.

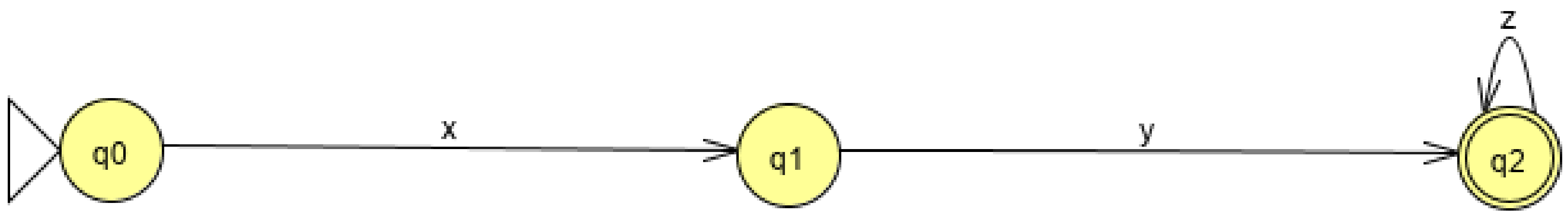


Autômato Finito Reconhecedor da Linguagem
$L_1 = \{w \mid w = xyz^*\}$

g	x	y	z
q0	q1	-	-
q1	-	q2	-
q2	-	-	q2

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

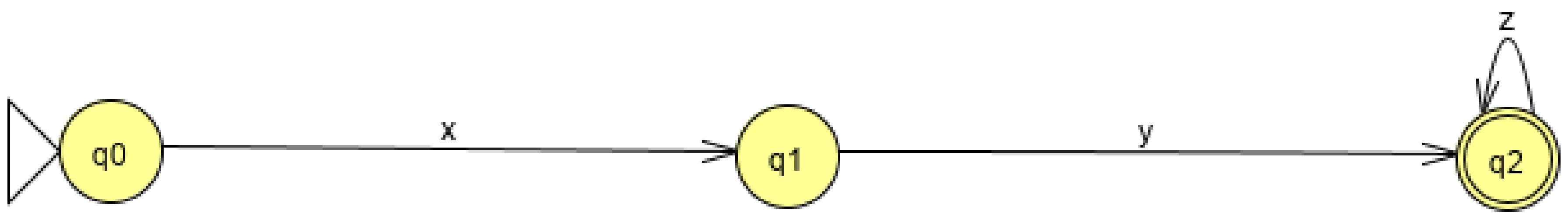
w_1 = xyzz ∈ L1?



Nesta configuração diz-se que o autômato finito, **aceita** a cadeia de entrada, ou seja, a cadeia de entrada pertence à Linguagem reconhecida pelo autômato.

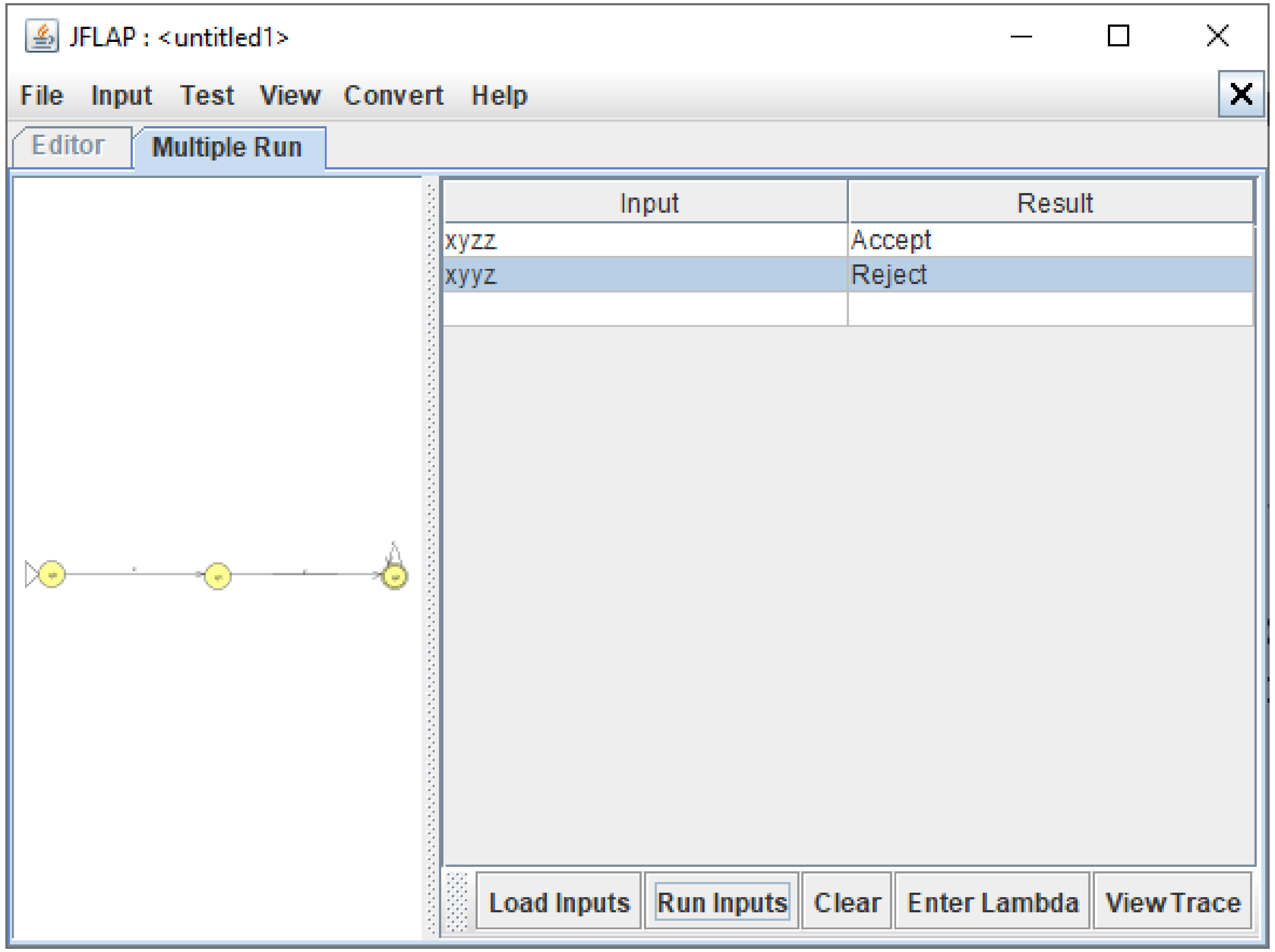
Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

w_1 = xyyz ∈ L1?



Nesta configuração diz-se que o diz-se que o autômato M **rejeita** a cadeia de entrada xyyz

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares



Uma Linguagem aceita por um autômato finito determinístico é uma **Linguagem Regular** ou do **tipo 3**.

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

Exemplo: Sejam ∑ = {x, y} e L, uma linguagem definida sobre o alfabeto ∑, tal que:

L= { w | w apresenta a sub-cadeia xyxy}

A representação algébrica para o autômato finito é:

Q = {q0, q1, q2, q3, q4}

∑ = {x, y}

g = {((q0,x), q1), {((q0,y), q0), ((q1,x), q1), ((q1,y), q2),

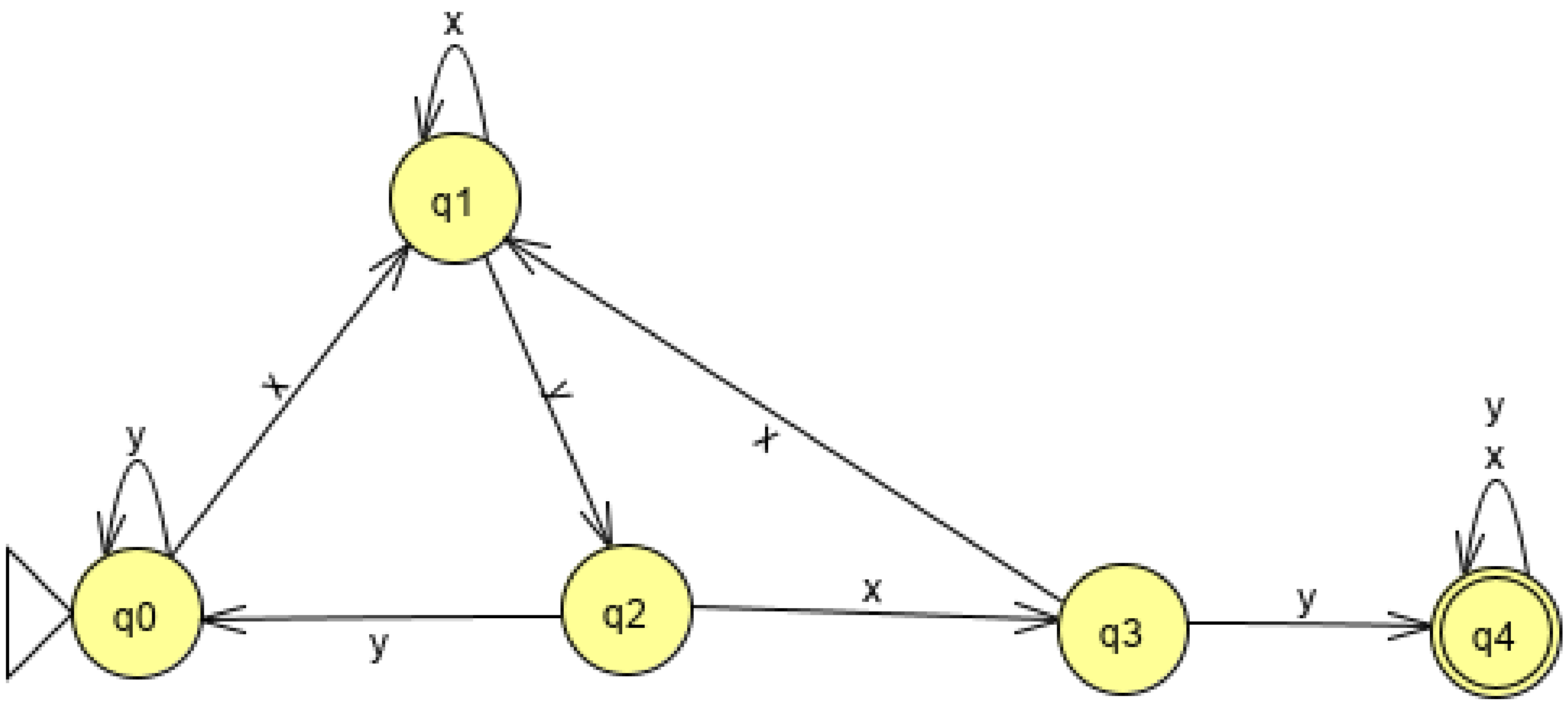
((q2,x), q3), ((q2,y), q0), ((q3,x), q1), ((q3,y), q4), ((q4,x), q4), ((q4,y), q4)}

q0 é o estado inicial.

F = {q4}

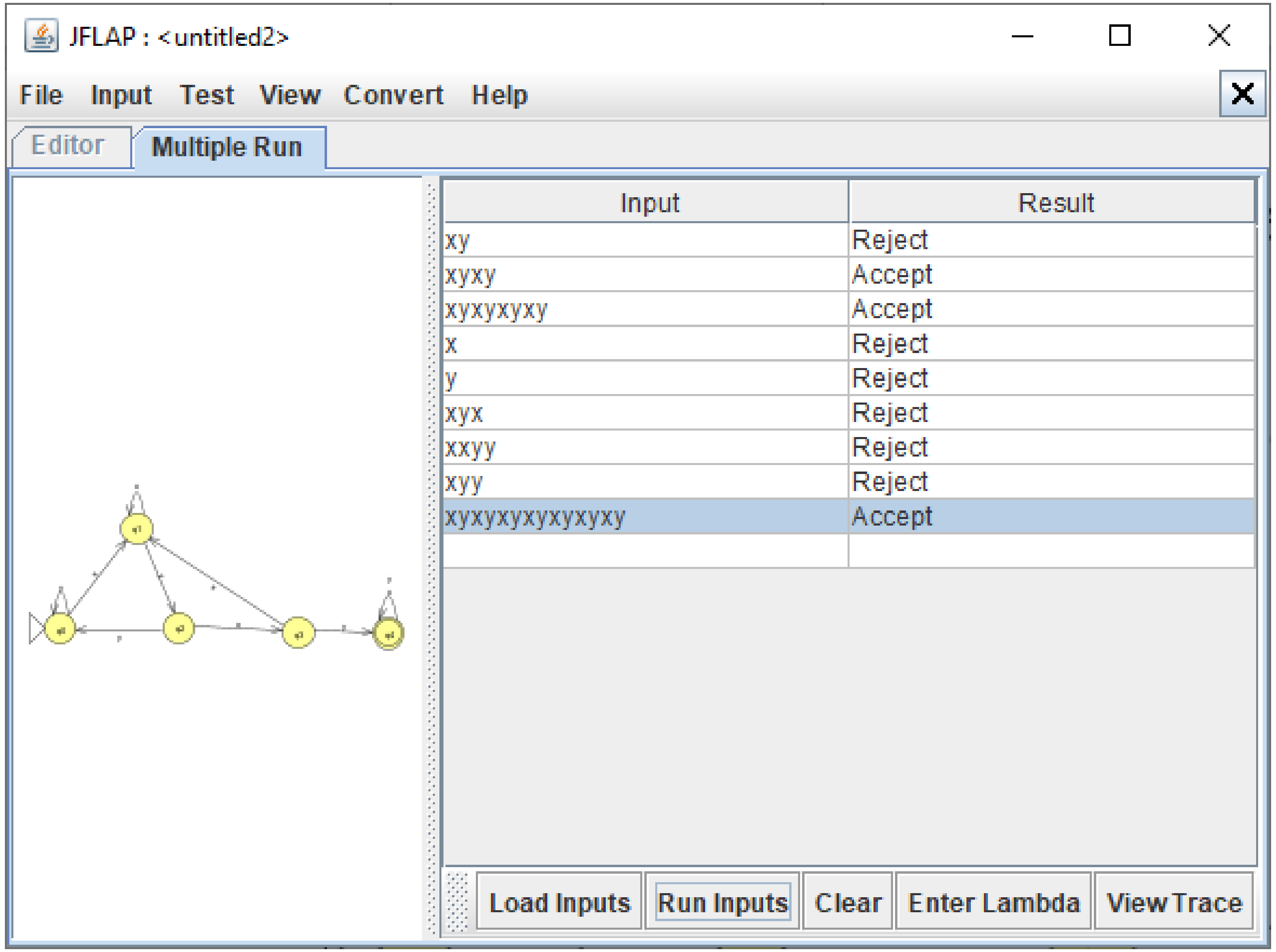
Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

A representação gráfica do autômato é como se ilustra na figura seguinte:



Autômato Finito para reconhecimento de
L= { w | w apresenta a sub-cadeia xyxy}

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares



Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

Exemplo: Considere-se o autômato M, representado em sua forma algébrica como: M = (Q, ∑, g, q0, F)

Onde Q = {q0, q1, q2 }

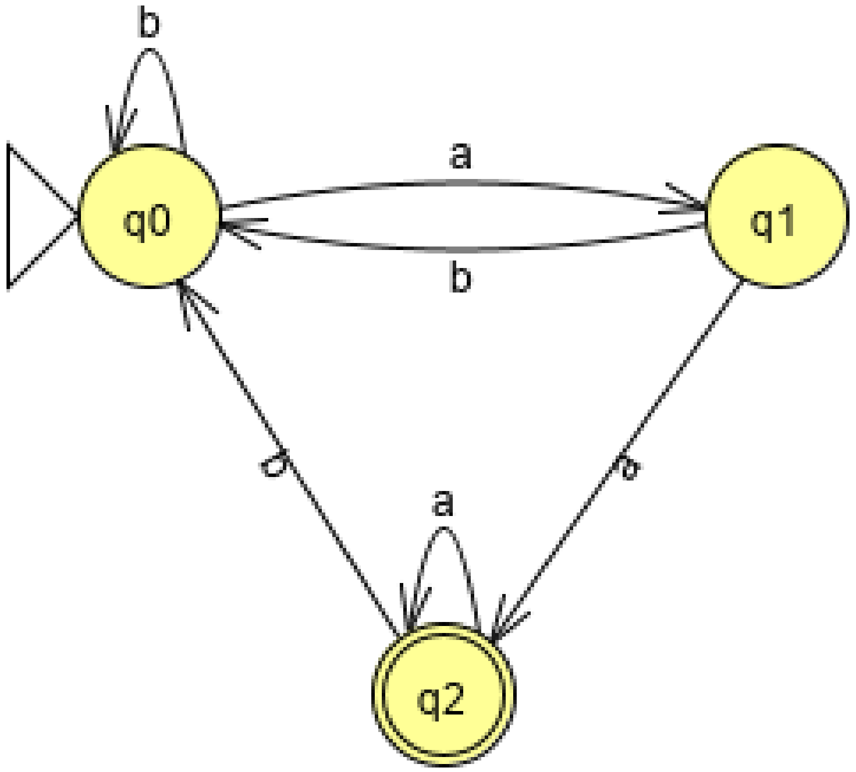
∑ = {a, b}

q0 é o estado inicial e F = {q2}

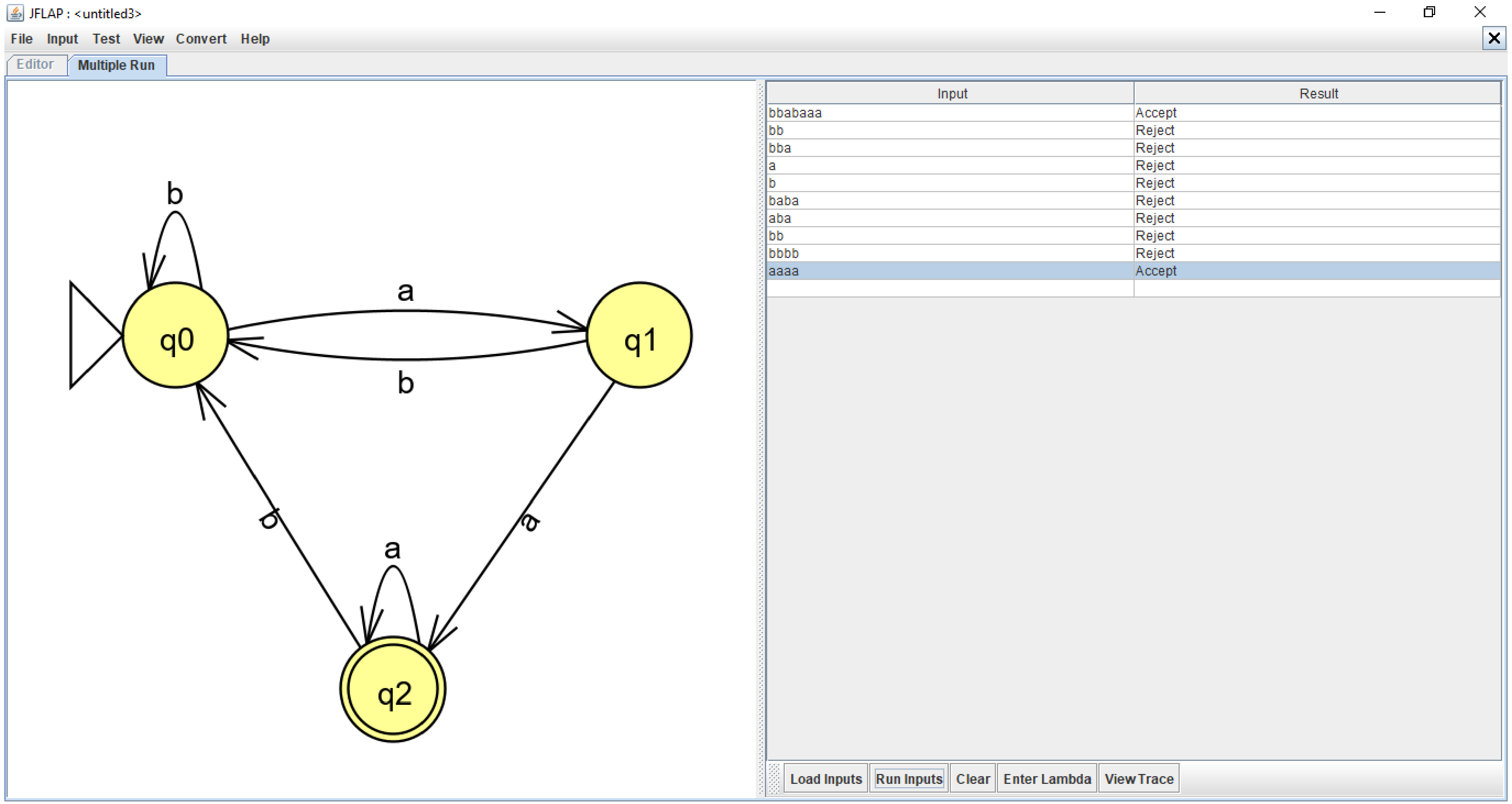
Tem-se que:

g(q0, a) = q1	g(q0,b) = q0	g(q1,a) = q2
g(q1, b) = q0	g(q2,a) = q2	g(q2, b) = q0

Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares



Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares



Exercícios

1) Considere a seguinte linguagem definida sobre o alfabeto ∑ = {a, b, c} e: L(w) = {w |a (a| b)+ c*}

O Autômato M reconhecedor de L(w) é M= (Q, q0, ∑, g, F) com:

Q = {q0, q1, q2, q3}

qo é o estado inicial;

∑ = {a, b, c}

F = {q2, q3}

A função g é dada por:

g = {((q0, a), q1), ((q1, a), q2), ((q1, b), q2), ((q2, a), q2), ((q2, b), q2), ((q2, c), q3), ((q3, c), q3)}

Faça o grafo no JFlap

Exercícios

2) Faça a gramática regular G = (V, T, P, Q0) que gere a linguagem.

G = (V, T, P, S)

V = {Q0, Q1, Q2, Q3}

T = {a, b, c}

P = {Q0 → aQ1;

Q1 → aQ2 | bQ2

Q2 → aQ2 | bQ2| cQ3 | ε

Q3 → cQ3 | ε

Referências desta aula

- AHO, Alfred V. Compiladores: princípios, técnicas e ferramentas. Rio de Janeiro, LTC, 1995

- DIVERIO, Tiaraju Asnuz; MENEZES, Paulo Blauth. Teoria da computação: máquinas universais e computabilidade. Sagra Luzzatto, 2003.

- EAD UNIP - Módulo3 - Ling. Reg -Parte I

FIM

OBRIGADO

RODRIGO