EAD

EDUCAÇÃO A DISTÂNCIA

UNIDADE I

Linguagens Formais e Autômatos

Prof. Me. Roberto Leminski

Por que estudar Linguagens?

- Ao criarmos um programa de computador, nos o fazemos em uma Linguagem de Computador.
- A maioria das Linguagens de Programação é utilizada em um Ambiente de Desenvolvimento Integrado (IDE – *Integrated Development Environment*), que inclui um compilador.

O compilador, antes de executar a função de tradução propriamente dita, realiza três tarefas na programação em que foi escrito:

- Análise Léxica: verificação das palavras.
- Análise Sintática: verificação das frases (linhas de código).
- Análise Semântica: verificação do significado e contexto.

Por que estudar Linguagens?

- As Análises Léxica, Sintática e Semântica estão associadas ao processamento das componentes Regular, Livre de Contexto e Dependente de Contexto, respectivamente, da Linguagem de Programação.

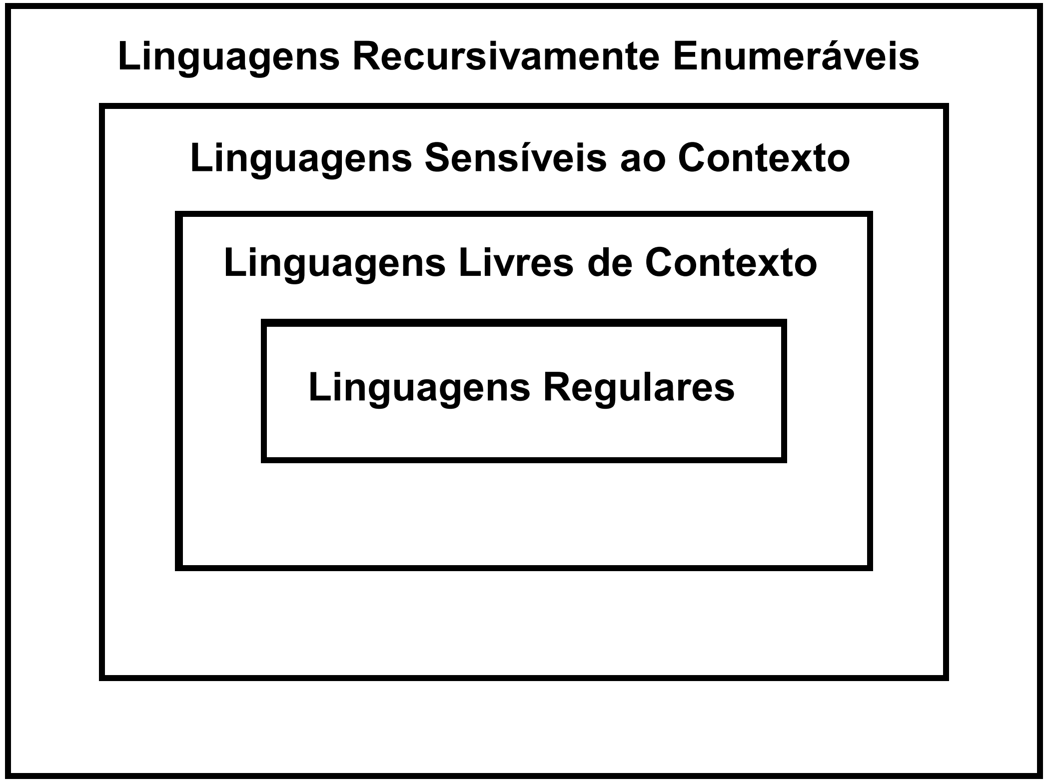
- Estes componentes foram classificados pelo linguista Noam Chomsky, que propôs, em 1857, uma classificação de linguagens em uma estrutura hierárquica que recebe o nome de Hierarquia de Chomsky.

Fonte: https://commons.wikimedia.org/wiki/File:NoamChomsky2020.png

Hierarquia de Chomsky

A Hierarquia de Chomsky classifica as linguagens formais em quatro níveis:

- Linguagens do tipo 0 ou recursivamente enumeráveis;
- Linguagens do tipo 1 ou sensíveis a contexto;
- Linguagens do tipo 2 ou livres de contexto;
- Linguagens do tipo 3 ou regulares.



Fonte: Adaptado de: livro-texto.

Componentes de uma linguagem formal

- Símbolo: é uma unidade atômica empregada na construção de cadeias e trata-se de um conceito primitivo; sua representação visual é irrelevante.
- Alfabeto: é um conjunto finito de símbolos.

Exemplos de alfabetos:

- $\Sigma = \{a, b, c, d, e\}$
- $\Sigma = \{0, 1, 2, 3\}$
- $\Sigma = \{*, +, \#, @\}$
- $\Sigma = \{\ \}$ (alfabeto vazio)

- Cadeia: é uma sequência finita de símbolos de um alfabeto.

  Exemplos de cadeias no alfabeto $\Sigma = \{a, b, c, d, e\}$:
  - aaabcd
  - ecba
  - ccabaeedaad

- Uma cadeia vazia é indicada pelo símbolo ε

Componentes de uma linguagem formal – Cadeias

- O comprimento de uma cadeia w é o número de símbolos presentes na cadeia e é representado pela notação |w|.

  Exemplos:

  - w = aaabcd; |w| = 6
  - w = ecba; |w| = 4
  - w = ccabaeedaad; |w| = 11

- Duas cadeias sobre o mesmo alfabeto podem ser combinadas e formarem uma terceira, a partir da operação de <u>concatenação</u>. A concatenação das cadeias v e w é representada por vw e formada pela justaposição de v e w.
  - Exemplo: seja w1 = 123 e w2 = 456, temos as concatenações w1w2 = 123456 e w2w1 = 456123.

Gramáticas

- Gramáticas são dispositivos geradores das cadeias que pertencem a uma linguagem formal.

Uma gramática é uma quádrupla (composta por quatro elementos) G = (V, Σ, P, S), em que:

- V é um conjunto finito de símbolos não terminais.
- Σ é um conjunto finito de símbolos terminais da gramática. Esse conjunto também é denominado alfabeto.

(Símbolos terminais são aqueles que aparecem nas cadeias finais produzidas pela gramática; símbolos não terminais aparecem como auxiliares durante a construção das cadeias, mas não nas cadeias finais).

- P é conjunto de produções ou regras de substituição da gramática.
- S é um elemento de V denominado símbolo inicial ou raiz da gramática.

Gramáticas – Regras de substituição

- Uma regra de substituição significa que um símbolo não terminal será substituído por um conjunto de símbolos não terminais e terminais.

- Esta substituição é indicada por uma seta ($\rightarrow$) que é sempre unidirecional.

- Regras de substituição serão aplicadas enquanto houver símbolos não terminais na cadeia que está sendo formada.

  - Pela aplicação de diferentes regras, podemos formar diferentes cadeias a partir de um conjunto de símbolos.

  - Estas regras devem ser tais que permitam a construção de toda e qualquer cadeia da linguagem definida pela gramática, e não devem permitir a construção de cadeias que não pertençam à linguagem.

Gramáticas – Exemplo

Considere a gramática formada por:

$V = \{S, P, Q\}$

$\Sigma = \{0,1,2,3\}$

Raiz: S

P (regras de substituição):

$S \rightarrow 0S$ (1)

$S \rightarrow 1S$ (2)

$S \rightarrow P$ (3)

$P \rightarrow 2Q$ (4)

$Q \rightarrow 2R$ (5)

$R \rightarrow 2R$ (6)

$R \rightarrow 2$ (7)

(As regras foram numeradas apenas para facilitar o exemplo)

Gramáticas – Exemplo

Considere a gramática formada por:

$V = \{S, P, Q\}$

$\Sigma = \{0,1,2,3\}$

Raiz: S

P (regras de substituição):

$S \rightarrow 0S \quad (1)$

$S \rightarrow 1S \quad (2)$

$S \rightarrow P \quad (3)$

$P \rightarrow 2Q \quad (4)$

$Q \rightarrow 2R \quad (5)$

$R \rightarrow 3R \quad (6)$

$R \rightarrow 3 \quad (7)$

Começamos com a raiz:

S

Regra 1:

0S

Regra 3:

0P

Regra 4:

02Q

Regra 5:

022R

Regra 7:

0223

(Não há mais símbolos não terminais)

Interatividade

Considere a gramática definida a seguir. Qual das cadeias indicadas **não** pode ser formada por meio dela?

$V = \{S, A\}$ $\Sigma = \{x, y, z\}$ Raiz: S

P (regras de substituição): $S \rightarrow xS$, $S \rightarrow yS$, $S \rightarrow A$, $A \rightarrow zA$, $A \rightarrow \varepsilon$

a) xxyyzz.
b) xxyy.
c) xxzzy.
d) zzz.
e) x.

Resposta

Considere a gramática definida a seguir. Qual das cadeias indicadas **<u>não</u>** pode ser formada por meio dela?

$V = \{S, A\}$ $\Sigma = \{x, y, z\}$ Raiz: S

P (regras de substituição): $S \rightarrow xS$, $S \rightarrow yS$, $S \rightarrow A$, $A \rightarrow zA$, $A \rightarrow \varepsilon$

a) xxyyzz.
b) xxyy.
c) xxzzy.
d) zzz.
e) x.

Linguagens regulares

- Uma Linguagem Regular é definida por uma Gramática Regular.
- Uma Gramática Regular é qualquer Gramática Linear.
- Existem Gramáticas Lineares à direita e à esquerda.

Gramáticas Lineares são aquelas em que todas as regras de substituição se dão da seguinte forma (sendo A e B símbolos não terminais e w um símbolo terminal da Linguagem):

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

$$A \rightarrow wB \text{ ou } A \rightarrow w$$

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

$$A \rightarrow Bw \text{ ou } A \rightarrow w$$

Expressões regulares

- Uma Gramática Linear à direita pode ser representada na forma de uma expressão regular.
- Uma expressão regular é uma notação alternativa para expressar uma Linguagem Regular, baseada em três operações.
- O termo deriva do trabalho do matemático norte-americano Stephen Cole Kleene, que desenvolveu as expressões regulares como uma notação ao que ele chamava de álgebra de conjuntos regulares.

Estas operações são:

- Alternância;
- Concatenação;
- Fecho de Kleene (ou, simplesmente, fechamento).

Expressões regulares – Operações

- Alternância indica todos os possíveis subconjuntos unitários com um dos elementos associados (não incluindo o conjunto vazio), sem repetição.

- Notação: a U b   ou   a + b   ou   a | b

Exemplos:

A | B = {A, B}

A | B | C = {A, B, C}

A alternância possui a propriedade associativa:

(A | B) | C = A | (B | C) = A | B | C = {A, B, C}

Expressões regulares – Operações

- <u>Concatenação</u> indica a sequência ordenada dos elementos, na ordem apresentada, sem repetição.

- <u>Notação</u>: a · b   ou   ab

<u>Exemplos</u>:

(A | B) C = {AC, BC}

(A | B) (C | D) = {AC, BC, AD, BD}

- Assim como na alternância, a concatenação irá resultar em conjuntos finitos de cadeias.

Expressões regulares – Operações

- Fecho de Kleene é o conjunto de todas as cadeias de símbolos que podem ser construídas ao concatenar zero ou mais cadeias com os símbolos envolvidos.

- Notação: a*

Exemplo:

A* = {ε, A, AA, AAA, AAAA, ...}

(ε indica a cadeia vazia)

- Diferente das duas outras operações, o Fecho de Kleene irá sempre resultar em conjuntos com infinitos elementos.

Expressões regulares – Exemplos

Mais alguns exemplos:

- a | b* define o conjunto de cadeias {ε, a, b, bb, bbb, ...}
- (a | b)* define o conjunto de todas as cadeias que contêm os símbolos a e b, incluindo a cadeia vazia: {ε, a, b, aa, ab, ba, bb, aaa, ...}
- ab*(c | ε) define o conjunto de cadeias começando com a, seguindo então com zero ou mais b e terminando, em alguns casos, com um único c: {a, ac, ab, abc, abb, ...}

- Se uma Linguagem regular possuir um Fecho de Kleene, ela irá sempre possuir infinitas cadeias.

Expressões regulares

Para transformar uma gramática regular à direita em uma expressão regular, seguimos os seguintes passos:

- Caso haja duas ou mais substituições a partir de um mesmo símbolo terminal, estas serão equivalentes a uma alternância;
- Um símbolo terminal que seja substituído por si mesmo corresponde a um fechamento;
- Os elementos da expressão regular serão uma concatenação a partir da substituição da raiz da gramática.

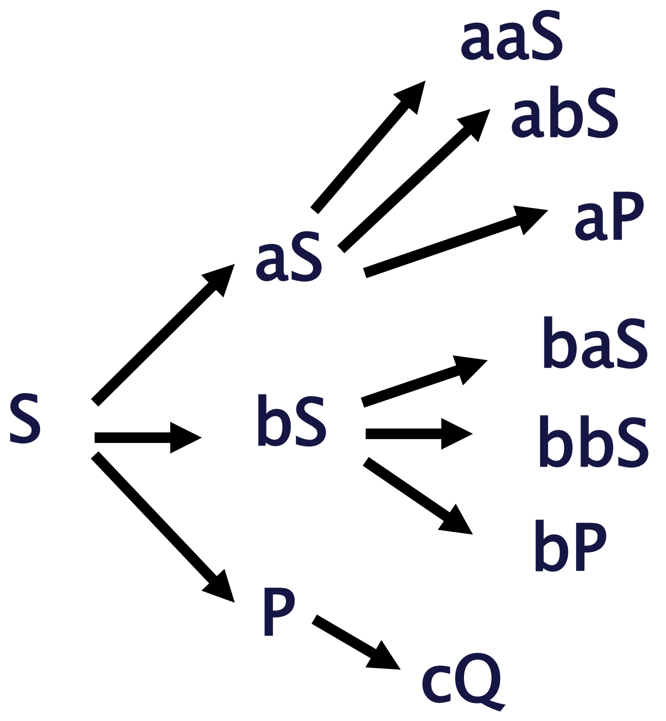
Expressões regulares

Exemplo: Considere a gramática a seguir:

$V = \{S, P, Q, R\}$ $\Sigma = \{a, b, c, d\}$ Raiz: S

P: $S \rightarrow aS$, $S \rightarrow bS$, $S \rightarrow P$, $P \rightarrow cQ$, $Q \rightarrow cR$, $R \rightarrow dR$, $R \rightarrow d$

(a | b)*cc dd*



Interatividade

Considere a linguagem regular definida pela expressão regular a seguir. Quais das cadeias indicadas podem ser formadas por meio dela?

L = (0 | 1)*(X | YY)*

a) 11XX, 0010, YYXX
b) 0XXX, XYX1, YYXX
c) 0XXX, XYX1, YXXX
d) XYY0, XXXY, 000Y
e) XXXX, 0000, 0YYY

Resposta

Considere a linguagem regular definida pela expressão regular a seguir. Quais das cadeias indicadas podem ser formadas por meio dela?

L = (0 | 1)*(X | YY)*

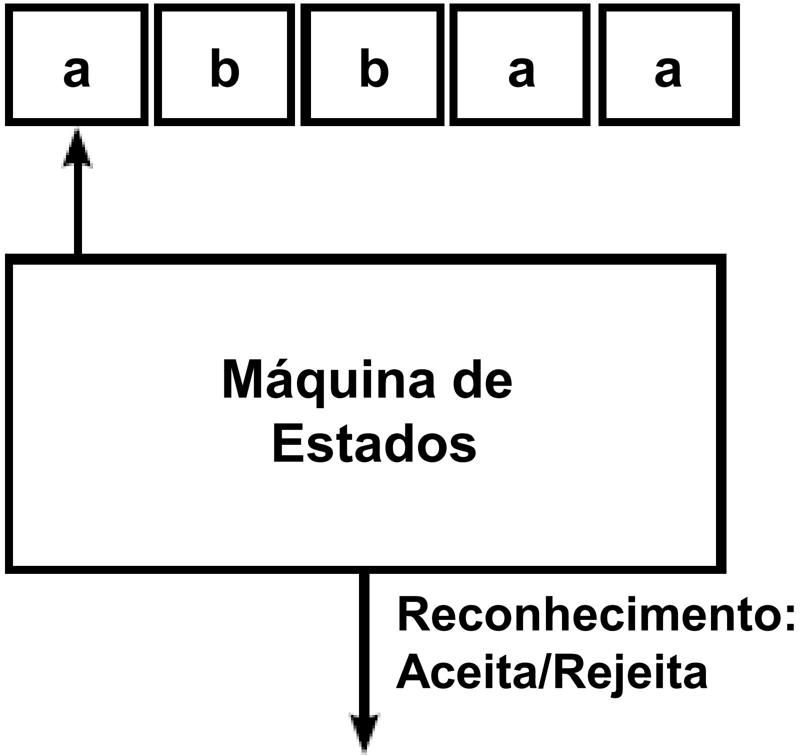
a) 11XX, 0010, YYXX
b) 0XXX, XYX1, YYXX
c) 0XXX, XYX1, YXXX
d) XYY0, XXXY, 000Y
e) XXXX, 0000, 0YYY

Autômatos finitos

- Os autômatos finitos são mecanismos de aceitação das sentenças das Linguagens Regulares.
- Assim, 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.

Um modelo de autômato finito é composto por:

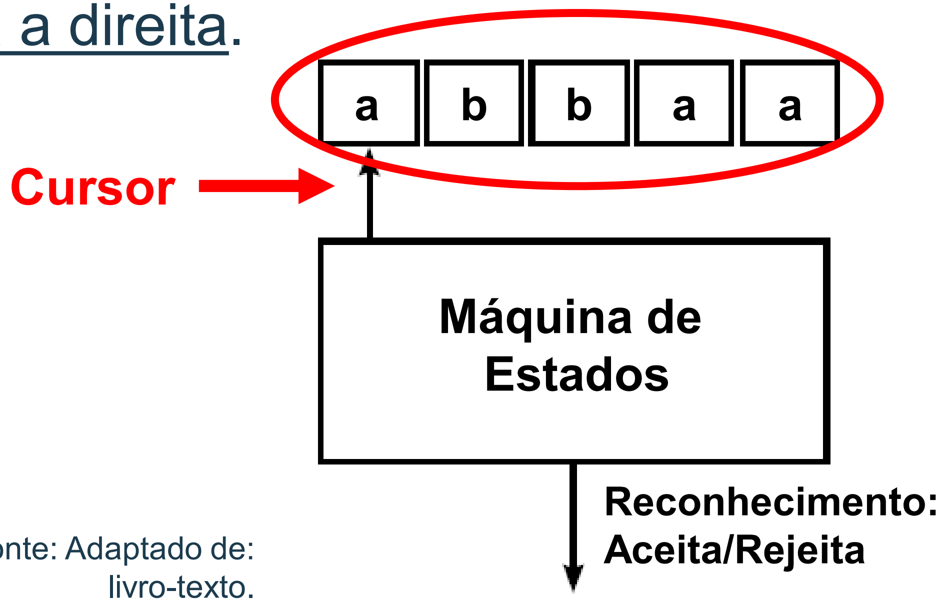
- Fita de entrada.
- Máquina de estados.



Fonte: Adaptado de: livro-texto.

Autômatos finitos – Fita de entrada

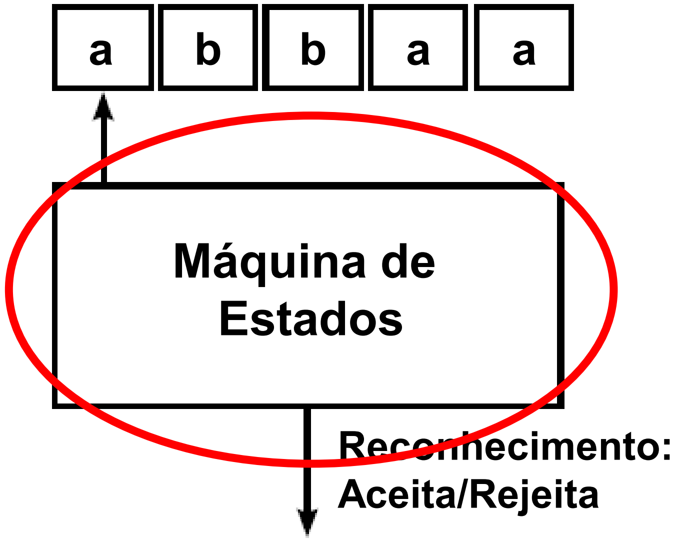
- Trata-se de um dispositivo de armazenamento, uma memória, que contém a cadeia a ser analisada pelo reconhecedor.
- A fita é finita, <u>dividida em células</u>, e cada célula armazena um símbolo da cadeia de entrada.
- Assim, o comprimento da fita de entrada é igual ao comprimento da cadeia de entrada.
- 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.
- <u>Não há operações de escrita sobre a fita</u>.
- <u>O cursor movimenta-se exclusivamente da esquerda para a direita</u>.



Fonte: Adaptado de: livro-texto.

Autômatos finitos – Máquina de estados

- A máquina de estados é o 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.
- Inicialmente, o cursor aponta para o símbolo mais à esquerda da cadeia.
- Neste início do processamento, em que a cadeia ainda será analisada, o controlador encontra-se no estado inicial, que deve ser único.
- Uma transição de estado pode ocorrer ou não, dependendo do estado atual e do último símbolo lido.



Fonte: Adaptado de: livro-texto.

Autômatos finitos – Máquina de estados

- Os autômatos finitos podem ser <u>determinísticos</u> ou <u>não determinísticos</u>.

Um Autômato Finito Determinístico (AFD), indicado por M, é uma quíntupla:

<u>M = (Q, Σ, g, q0, F)</u>

- Q é um conjunto finito de estados;
- Σ é um alfabeto (finito e não vazio) de entrada;
- g é um conjunto de funções de transição, g: $Q \times S \rightarrow Q$;
- q0 é o estado inicial, $q0 \in Q$;
- F é um conjunto de estados finais, $F \subseteq Q$.

Autômatos finitos

Exemplo: considere o autômato finito M definido por:

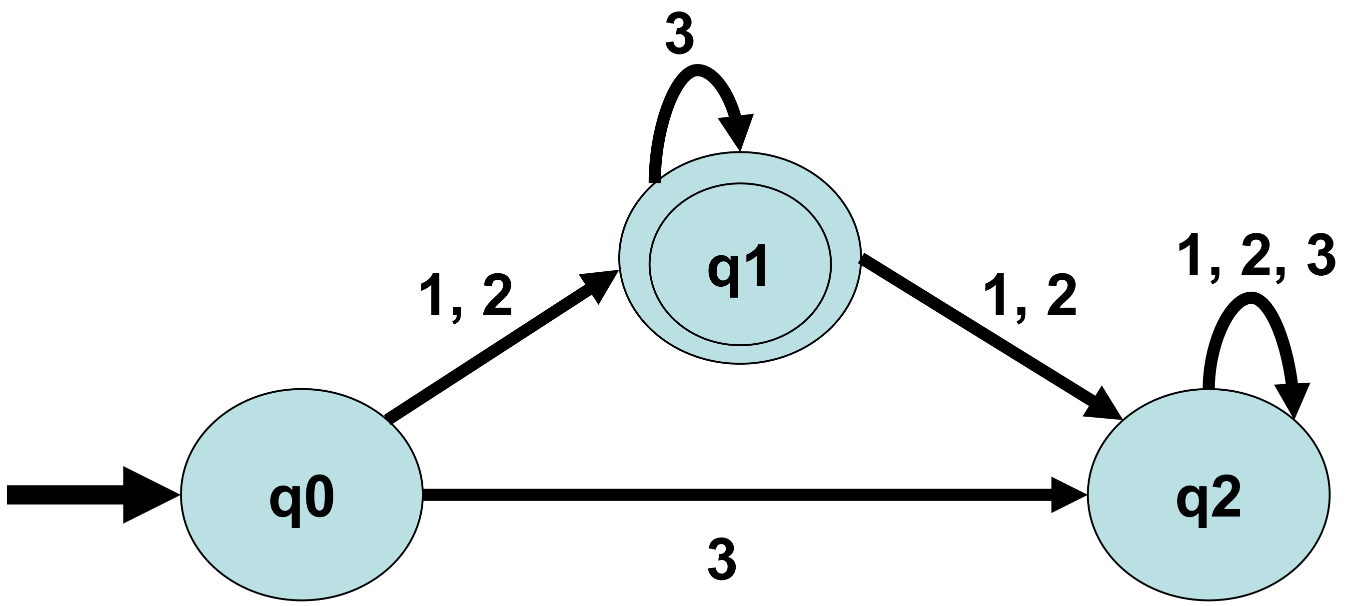
- Q = {q0, q1, q2)
- Σ = {1, 2, 3}
- g = {((q0,1), q1), ((q0,2), q1), ((q0,3), q3), ((q1,1), q2), ((q1,2), q2), ((q1,3), q1), ((q2,1), q2), ((q2,2), q2), ((q2,3), q2)}
- F = {q1}

Autômatos finitos

Exemplo: considere o autômato finito M definido por:

- Q = {q0, q1, q2}
- Σ = {1, 2, 3}
- g = {((q0,1), q1), ((q0,2), q1), ((q0,3), q3), ((q1,1), q2), ((q1,2), q2), ((q1,3), q1), ((q2,1), q2), ((q2,2), q2), ((q2,3), q2)}
- F = {q1}

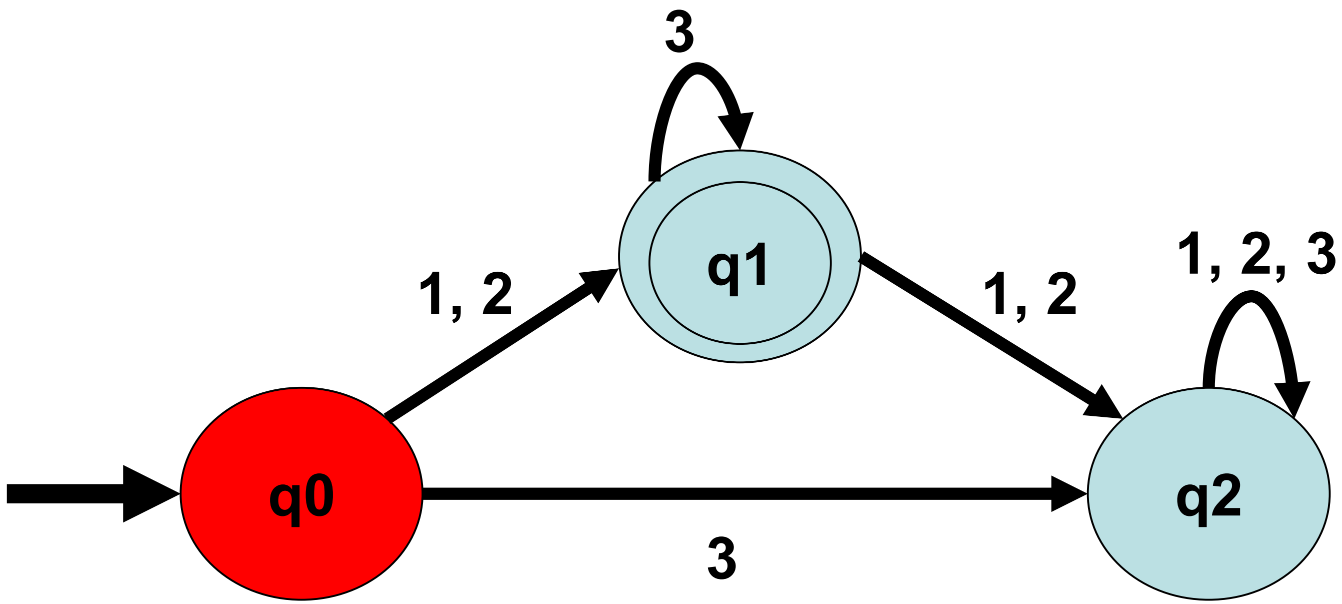
Graficamente:



Autômatos finitos

Testando a cadeia 1231:

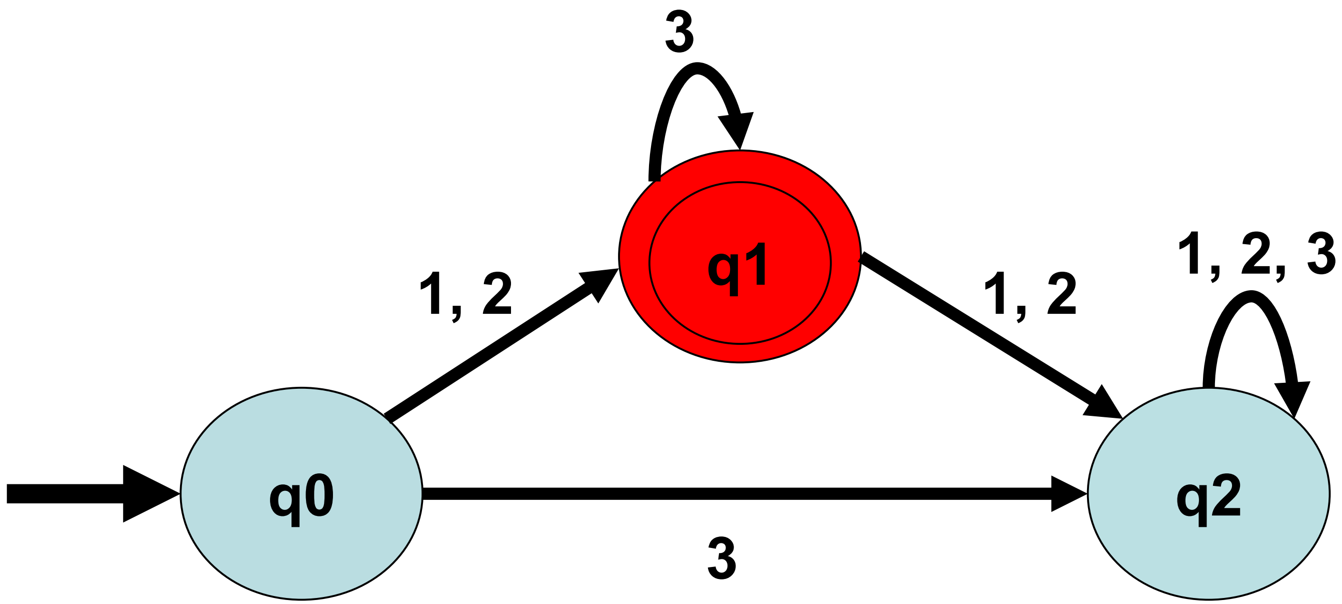
1	2	3	1



Autômatos finitos

Testando a cadeia 1231:

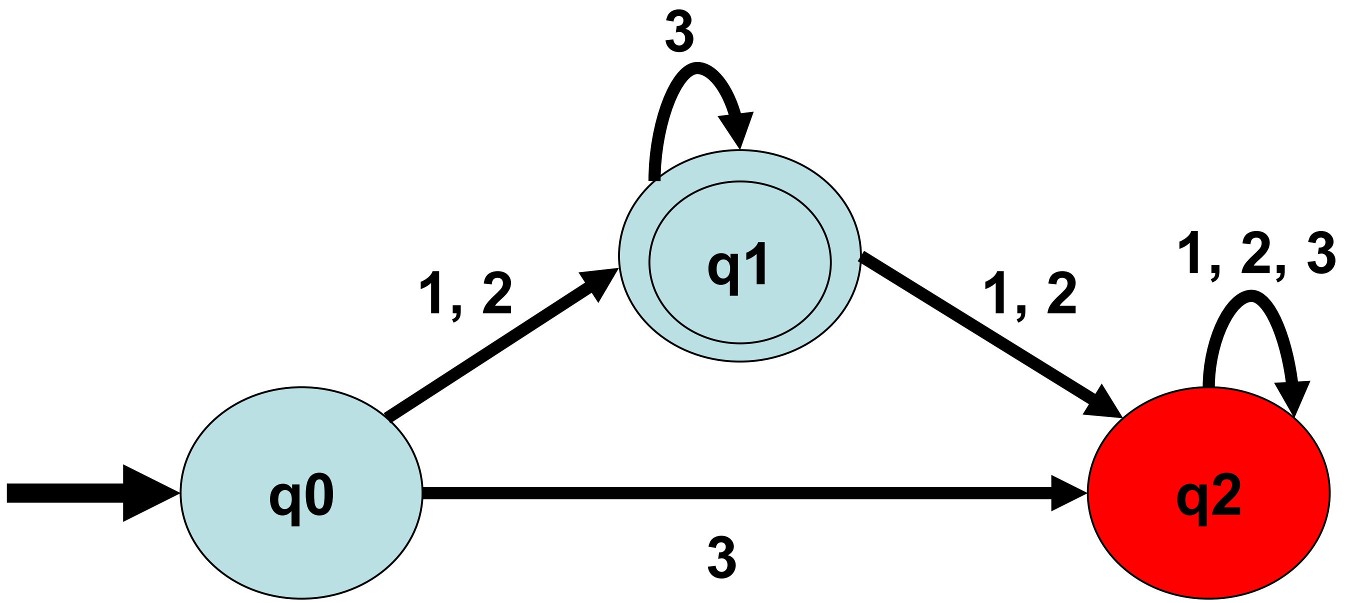
1	2	3	1



Autômatos finitos

Testando a cadeia 1231:

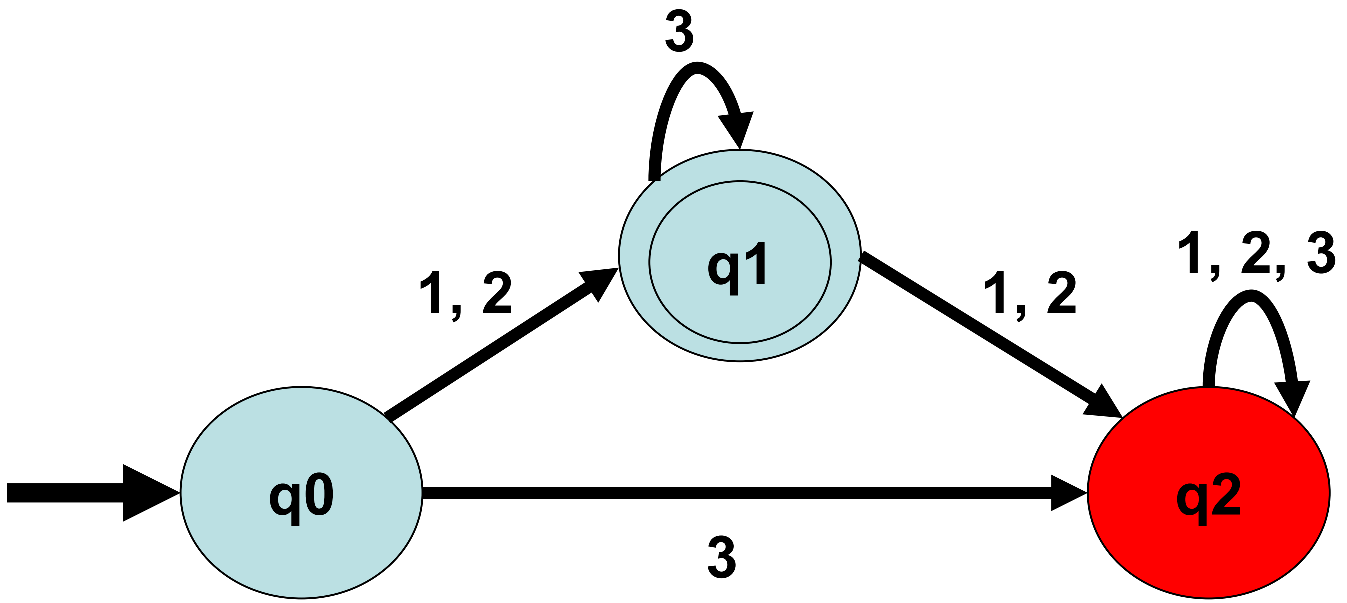
1	2	3	1



Autômatos finitos

Testando a cadeia 1231:

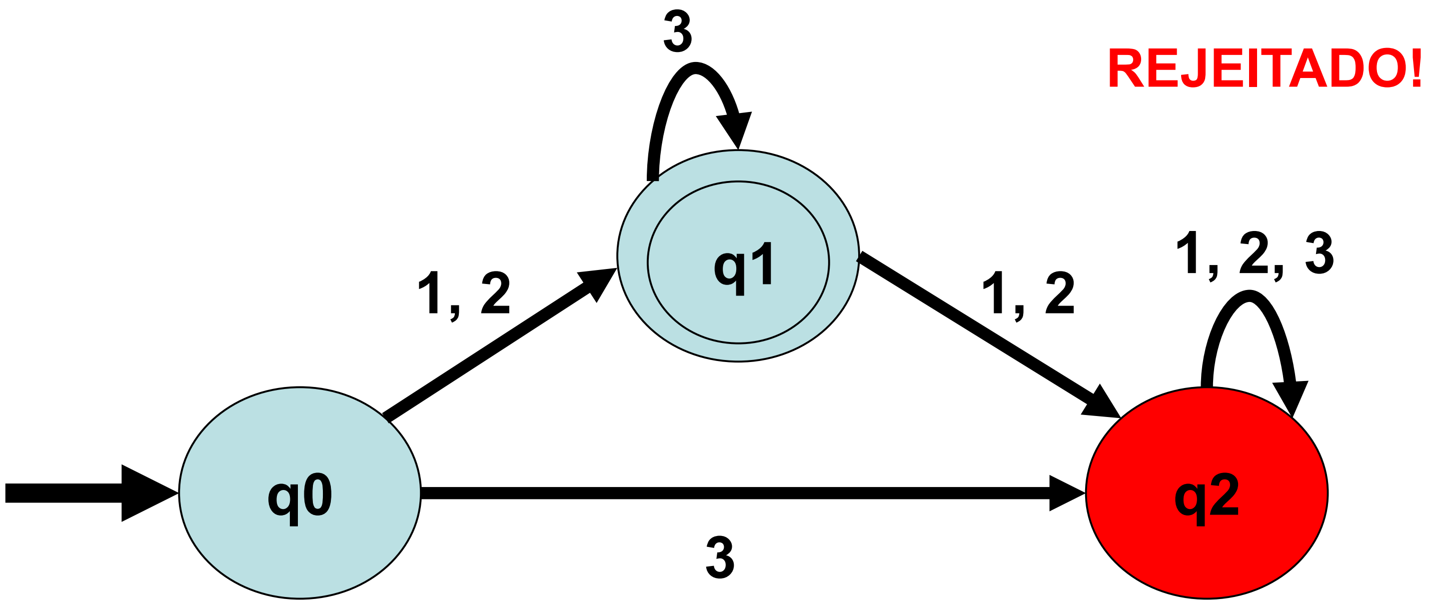
1	2	3	1



Autômatos finitos

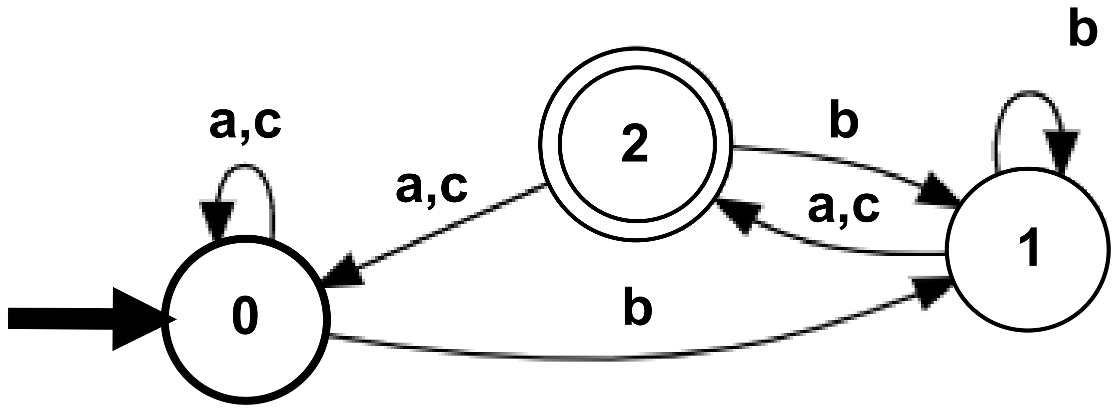
Testando a cadeia 1231:

1	2	3	1



Interatividade

Considere o autômato finito determinístico indicado na imagem, no qual Q = {0, 1, 2}, Σ = {a, b, c}, qo= {0}, F = {1}.



Autômato criado com o FSM simulator.

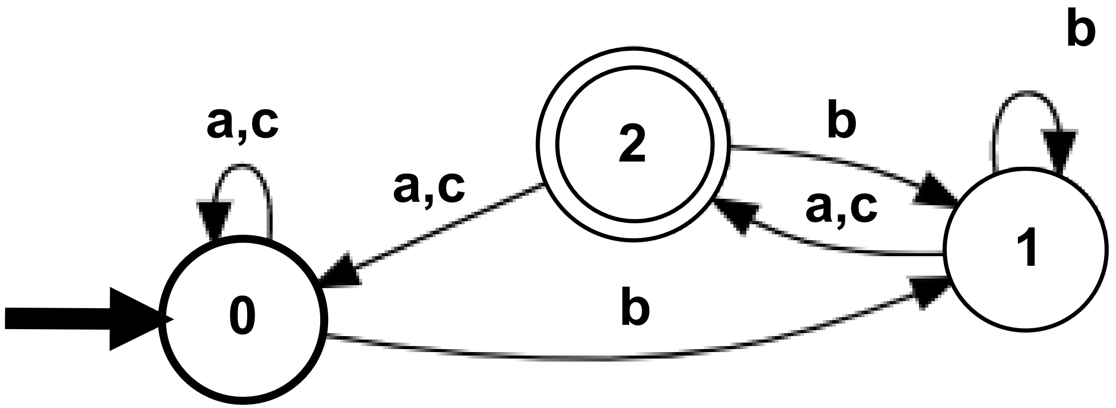
Fonte: Adaptado de: https://ivanzuzak.info/noam/webapps/fsm_simulator)

<u>Não</u> é uma cadeia aceita por ele:

a) babc
b) babbc
c) acbab
d) bbbcaba
e) ccbbc

Resposta

Considere o autômato finito determinístico indicado na imagem, no qual Q = {0, 1, 2}, Σ = {a, b, c}, qo= {0}, F = {1}.



Autômato criado com o FSM simulator.

Fonte: Adaptado de: https://ivanzuzak.info/noam/webapps/fsm_simulator)

Não é uma cadeia aceita por ele:

a) babc
b) babbc
c) acbab
d) bbbcaba
e) ccbbc

Obtenção de autômato finito a partir de gramáticas

- Seja G = (V, Σ, P, S) uma Gramática Regular linear à direita.

Pode-se obter o autômato finito determinístico seguindo-se as etapas:

- Para cada símbolo não terminal da gramática, há um correspondente estado para o autômato;
- O símbolo inicial S da gramática G corresponde ao estado inicial do autômato M;
- O alfabeto de entrada Σ de M é o mesmo alfabeto Σ de G.

As funções de transição g do autômato são construídas a partir das regras de substituição da gramática, conforme a tabela (maiúsculas são não terminais e minúsculas são terminais):

Substituição	Transição
$S \rightarrow aA$	g(S, a) = A, S estado inicial
$A \rightarrow bB$	g(A, b) = B
$A \rightarrow \varepsilon$	A é estado final
$A \rightarrow a$	A é estado final

Obtenção de autômato finito a partir de gramáticas

Por exemplo, consideremos a gramática linear definida por:

$V = \{S, A, B\}$ $\Sigma = \{x, y, z\}$ Raiz: S

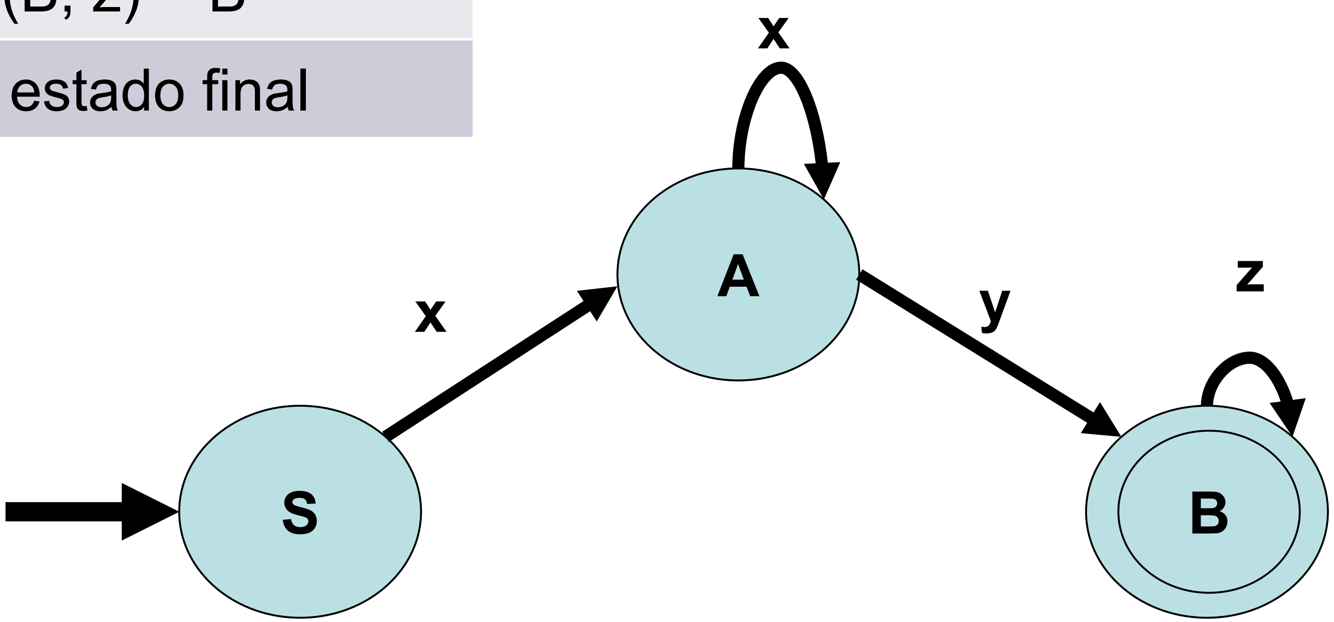
P: $S \rightarrow xA$, $A \rightarrow xA$, $A \rightarrow yB$, $B \rightarrow zB$, $B \rightarrow \varepsilon$

Substituição	Transição
$S \rightarrow xA$	$g(S, x) = A$
$A \rightarrow xA$	$g(A, x) = A$
$A \rightarrow yB$	$g(A, x) = B$
$B \rightarrow zB$	$g(B, z) = B$
$B \rightarrow \varepsilon$	B é estado final

Todas as demais transições levam para um estado extra, sem saída (vamos chamá-lo de estado C).

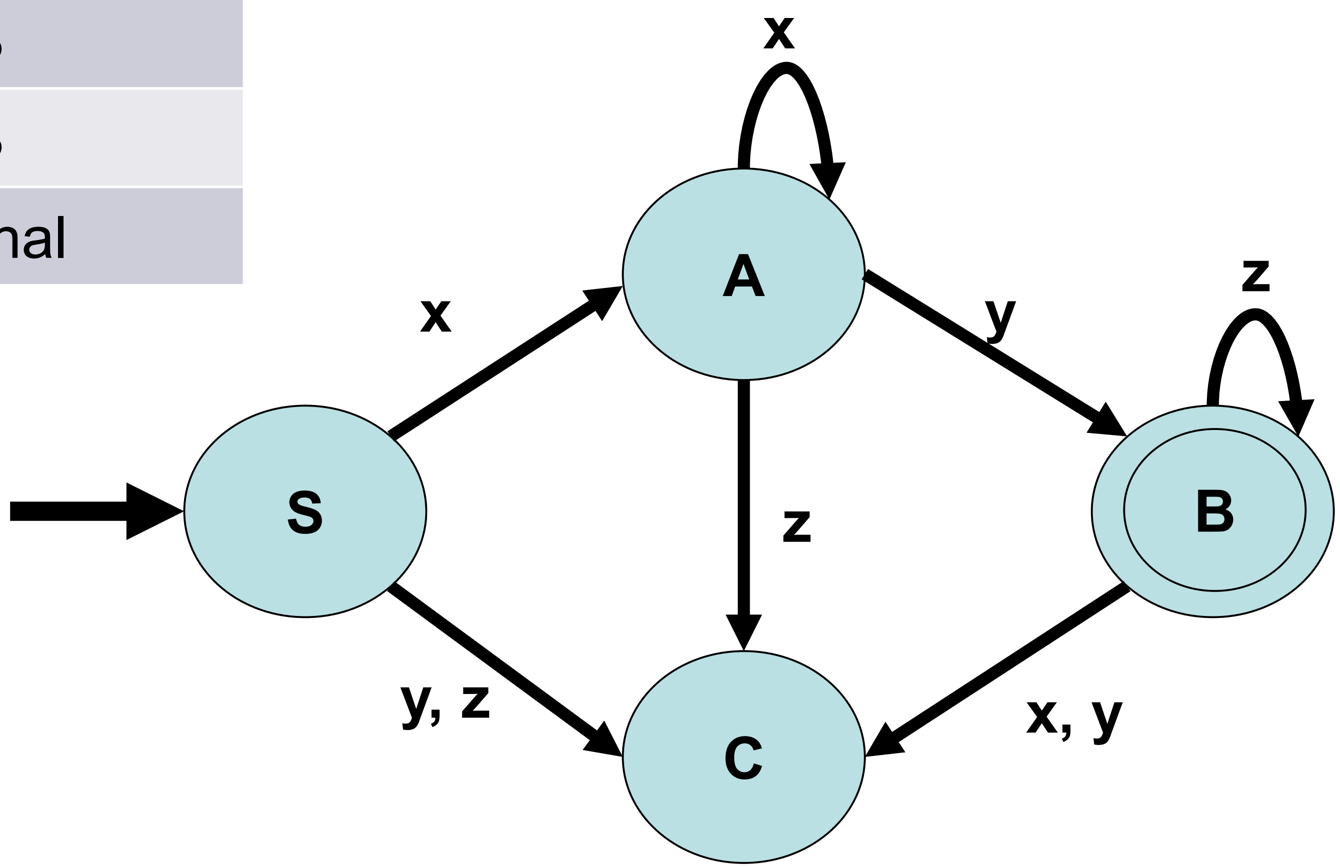
Obtenção de autômato finito a partir de gramáticas

Substituição	Transição
$S \rightarrow xA$	$g(S, x) = A$
$A \rightarrow xA$	$g(A, x) = A$
$A \rightarrow yB$	$g(A, y) = B$
$B \rightarrow zB$	$g(B, z) = B$
$B \rightarrow \varepsilon$	B é estado final



Obtenção de autômato finito a partir de gramáticas

Substituição	Transição
$S \rightarrow xA$	g(S, x) = A
$A \rightarrow xA$	g(A, x) = A
$A \rightarrow yB$	g(A, y) = B
$B \rightarrow zB$	g(B, z) = B
$B \rightarrow \varepsilon$	B é estado final



Obtenção de autômato finito a partir da expressão regular

- De forma semelhante, pode-se obter o autômato finito a partir de uma expressão regular de uma gramática.

Neste caso, as etapas são as seguintes:

- Desconsiderar incialmente todos os fechamentos;
- O número de estados será o número de elemento restantes na expressão mais dois;
- Traçar uma trajetória direta, partindo do estado inicial até o final, em que cada transição correspondente a cada alternância e concatenação da expressão regular;
- Os fechamentos serão transições que começam e terminam no mesmo estado;
- Quaisquer outras transições levarão para um estado sem saída.

Obtenção de autômato finito a partir da expressão regular

- Exemplo: construir o autômato para a expressão regular (a | b)cc*dd*

- Desconsiderando por enquanto os fechamentos, temos (a | b) c d; como são três elementos, nosso autômato terá cinco estados.

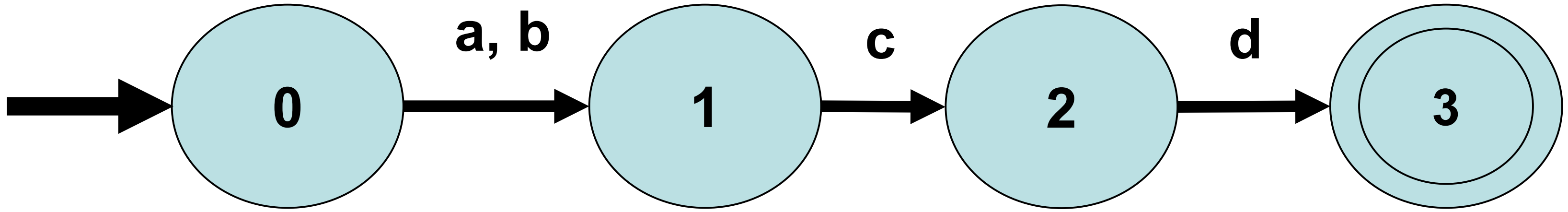
Traçar uma trajetória do estado inicial até o final:

Obtenção de autômato finito a partir da expressão regular

- Exemplo: construir o autômato para a expressão regular (a | b)cc*dd*

- Desconsiderando por enquanto os fechamentos, temos (a | b) c d; como são três elementos, nosso autômato terá cinco estados.

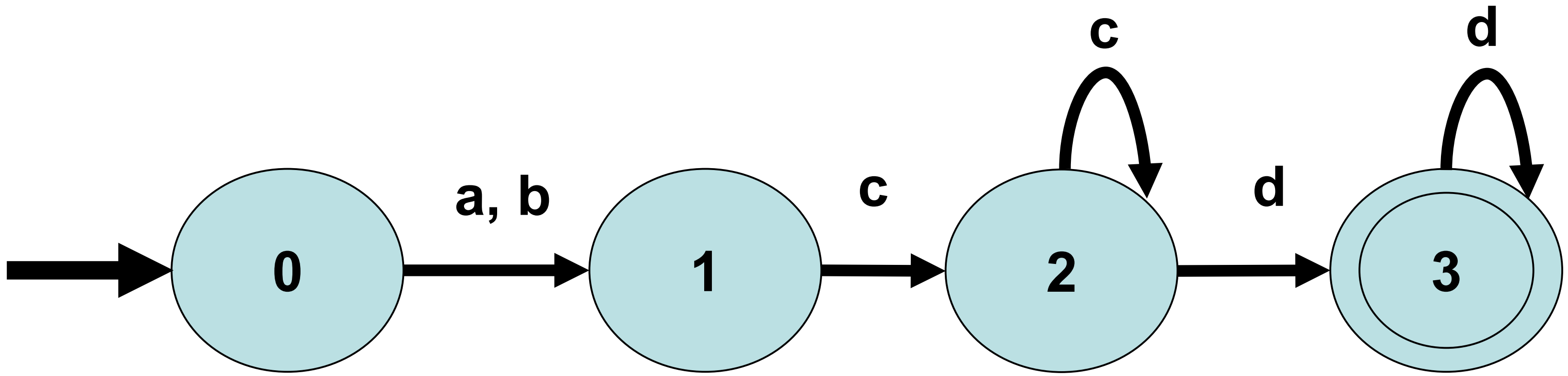
Traçar uma trajetória do estado inicial até o final:



Obtenção de autômato finito a partir da expressão regular

- Exemplo: construir o autômato para a expressão regular (a | b)cc*dd*
- Desconsiderando por enquanto os fechamentos, temos (a | b) c d; como são três elementos, nosso autômato terá cinco estados.
- Traçar uma trajetória do estado inicial até o final;

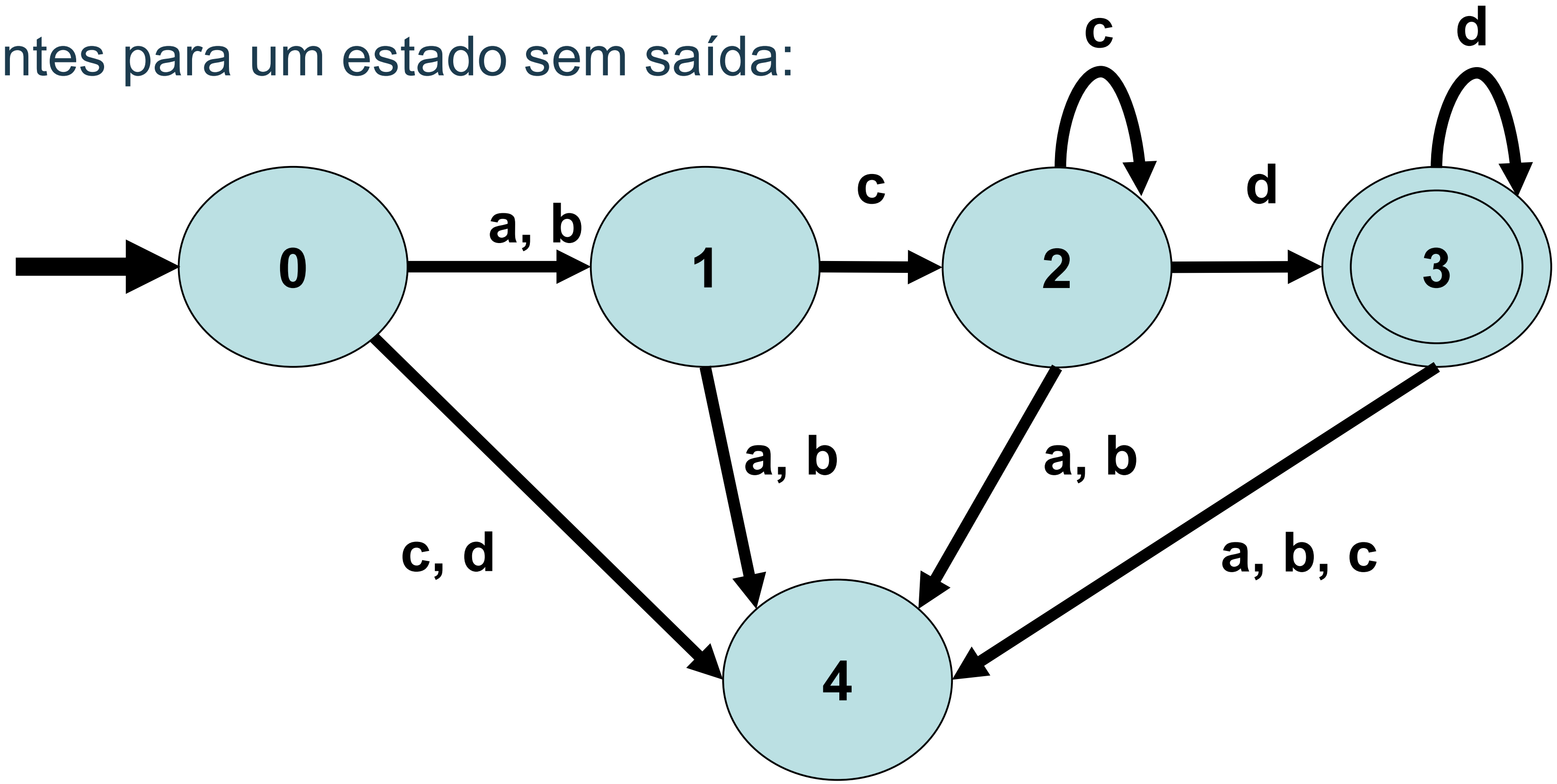
Acrescentar os fechamentos:



Obtenção de autômato finito a partir da expressão regular

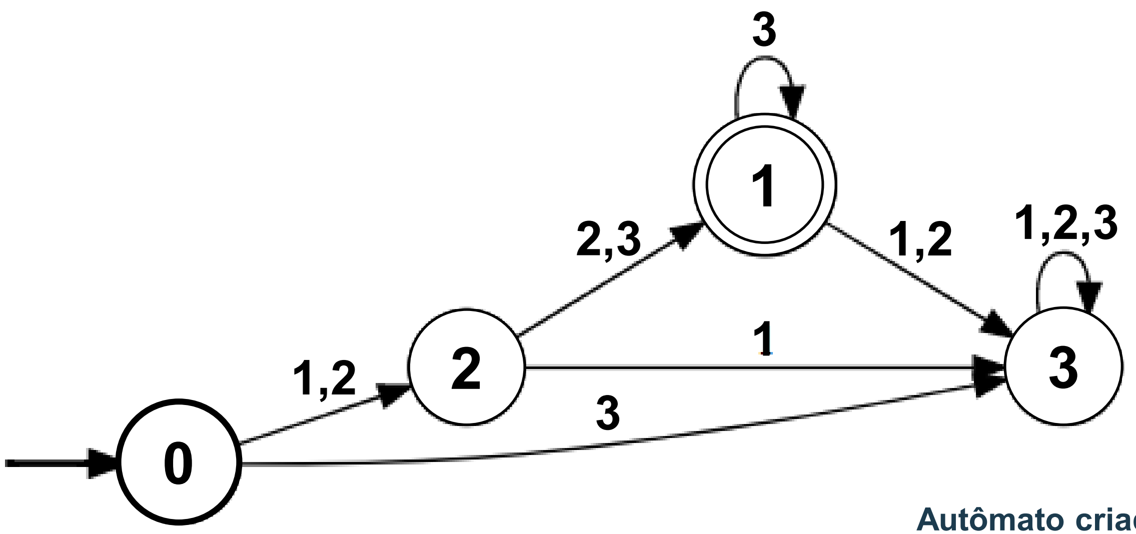
- Exemplo: construir o autômato para a expressão regular (a | b)cc*dd*
- Desconsiderando por enquanto os fechamentos, temos (a | b) c d; como são três elementos, nosso autômato terá cinco estados.
- Traçar uma trajetória do estado inicial até o final;
- Acrescentar os fechamentos;

Acrescentar as transições restantes para um estado sem saída:



Interatividade

Considere o autômato finito determinístico indicado na imagem. Ele foi gerado a partir de qual expressão regular?

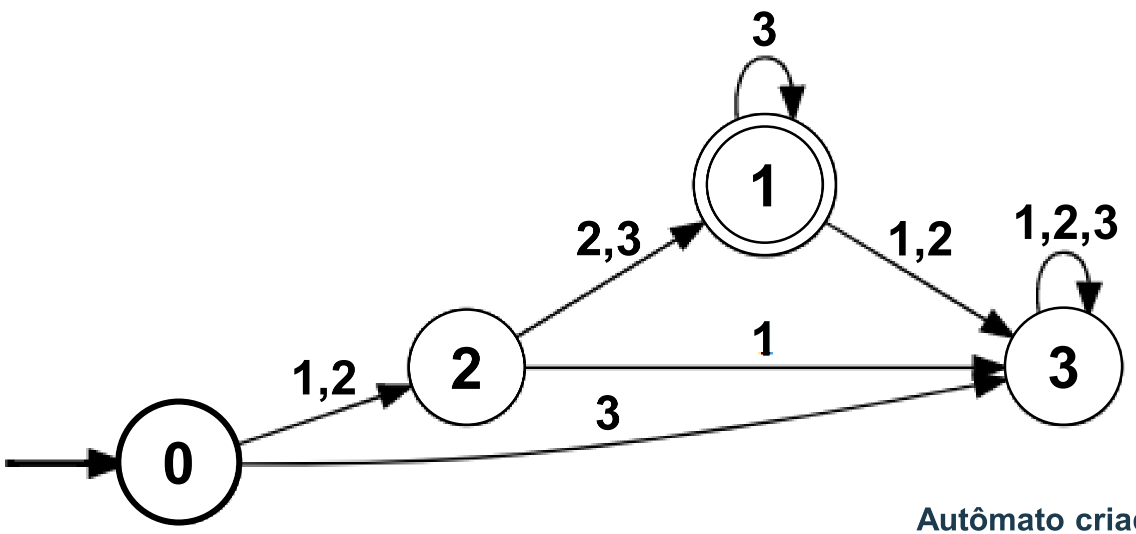


Autômato criado com o FSM simulator.
Fonte: Adaptado de: https://ivanzuzak.info/noam/webapps/fsm_simulator)

a) 12(2 | 3)3*
b) (1 | 2)23*
c) (1 | 2)233*
d) (1 | 2)(2 | 3)
e) (1 | 2)(2 | 3)3*

Resposta

Considere o autômato finito determinístico indicado na imagem. Ele foi gerado a partir de qual expressão regular?



Autômato criado com o FSM simulator.
Fonte: Adaptado de:
https://ivanzuzak.info/noam/webapps/fsm_simulator)

a) 12(2 | 3)3*
b) (1 | 2)23*
c) (1 | 2)233*
d) (1 | 2)(2 | 3)
e) (1 | 2)(2 | 3)3*

Referências

- LEWIS, H. R. P.; CHRISTOS, H. *Elementos de teoria da computação*. 2. ed. Porto Alegre: Bookman, 2000.
- MENEZES, P. B. *Linguagens formais e autômatos*. 2. ed. Porto Alegre: Sagra Luzzatto, 1998.
- RAMOS, M. V. M.; NETO, J. J.; VEGA, I. S. *Linguagens formais*. Porto Alegre: Bookman, 2009.

ATÉ A PRÓXIMA!