Ciência da Computação
UNESP -Presidente Prudente

LFA – Aula 07

Equivalência entre AFD e AFND
AFND: uma aplicação – busca em textos
Equivalência entre ER's e AF's
Equivalência entre GR's e AF's

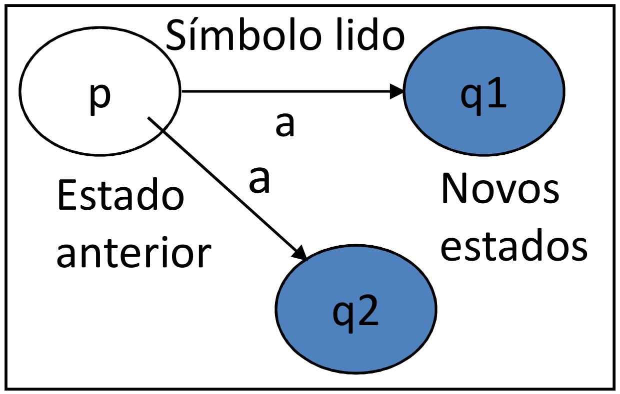
Celso Olivete Júnior

olivete@fct.unesp.br

www.fct.unesp.br/docentes/dmec/olivete/lfa

Na aula passada...

- Autômato finito não-determinístico com e sem movimentos vazios
  - O autômato tem o poder de estar em vários estados ao mesmo tempo



No estado **p** ao ler o símbolo **a** assume **q1** e **q2** como novos estados atuais

Na aula de hoje:

- Equivalência entre AFND e AFD
- AF: uma aplicação – busca em textos
- Conversões:

  ER's em AF's e AF's em ER's

  GR's em AF's e AF's em GR's

- Referência bibliográfica

  HOPCROFT, J. E.; ULLMAN, J. D.; MOTWANI, R. ***Introdução à Teoria de Autômatos, Linguagens e Computação.*** Editora Campus, 2002 → Capítulos 2 e 3

Equivalência entre AFD's e AFND's

Equivalência entre AFD e AFND

- Teorema: Seja L o conjunto aceito por um AFND, então existe um AFD que aceita L - são equivalentes.

- Embora muitas vezes seja mais fácil construir um AFND para uma L, o AFD tem na prática quase o mesmo número de estados que um AFND, embora ele tenha mais transições.

- No pior caso, o menor AFD pode ter 2^n estados, enquanto o menor AFND (para a mesma linguagem) tem apenas n estados

Equivalência entre AFD e AFND

• A prova de que os AFD's podem fazer tudo o que os AFND's podem fazer envolve a construção de subconjuntos

•A construção dos subconjuntos começa a partir de um AFND N = $(Q_n, \Sigma, \delta_n, \{q_0\}, F_n)$. O objetivo é a descrição de um AFD D= $(Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ ➔ tal que L(D) = L(N)

- Σ é o mesmo
- O estado inicial de D é o conjunto que contem apenas o estado inicial de N

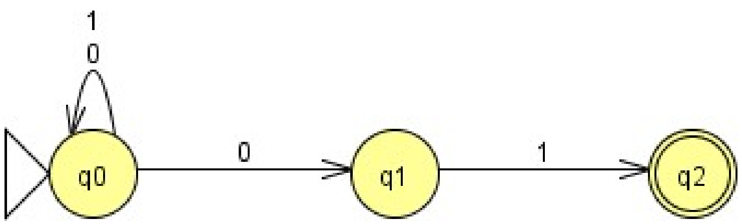
Equivalência entre AFD e AFND

- Construção dos outros elementos de D
  - Q_D é o conjunto de subconjuntos de Q_N
  - Q_D representa o conjunto de potências de Q_N. Ex:
    - Se Q_N tem n estados Q_D terá 2^n estados (no pior caso)
  - F_D é o conjunto de subconjuntos de S de Q_N → representa todos os conjuntos de estados de N que incluem pelo menos um estado de aceitação de N

$$\delta_D (S,a) = \bigcup \ \delta_N (p,a)$$

  - Para calcular $\delta_D (S,a)$, basta observar todos os estados de p em S, e ver para quais estados N vai para p sobre a entrada "a" e fazemos a união de todos esses estados.

AFND aceita cadeia de 0’s e 1’s com final 01



Estados de N → {q0,q1,q2}

- 3 estados: logo envolverá a construção de $2^3 = 8$ subconjuntos. O AFD terá no máximo 8 estados.

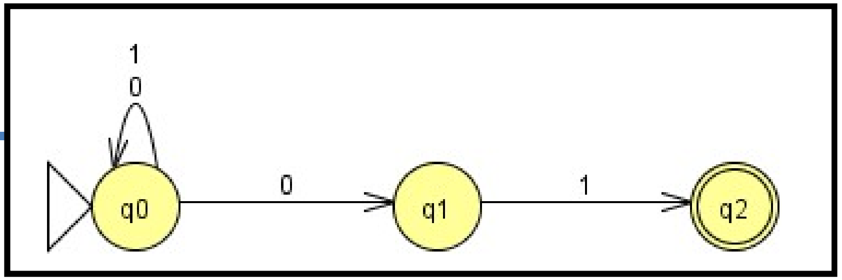
Construção dos subconjuntos

	Entrada	
Estado	0	1
Ø	Ø	Ø
→{q0}	{q0,q1}	{q0}
{q1}	Ø	{q2}
*{q2}	Ø	Ø
{q0,q1}	{q0,q1}	{q0,q2}
*{q0,q2}	{q0,q1}	{q0}
*{q1,q2}	Ø	{q2}
*{q0,q1,q2}	{q0,q1}	{q0,q2}

Renomeando os subconjuntos

	Entrada	
Estado	0	1
A	A	A
→B	E	B
C	A	D
*D	A	A
E	E	F
* F	E	B
* G	A	D
* H	E	F

Equivalência entre AFD e AFND



Eliminando estados inacessíveis

A partir de B (est.inicial) só é possível chegar em B, E e F. Os outros estados são inacessíveis e devem ser removidos.

Para cada entrada a calcula-se δ_D (S,a) → obtêm os acessíveis

Para o exemplo anterior

B δ_D ({q0}, 0) = {**q0,q1**} δ_D ({q0}, 1) = {**q0**}

E δ_D ({**q0,q1**}, 0) = {q0,q1} δ_D ({**q0,q1**}, 1) = {**q0,q2**}

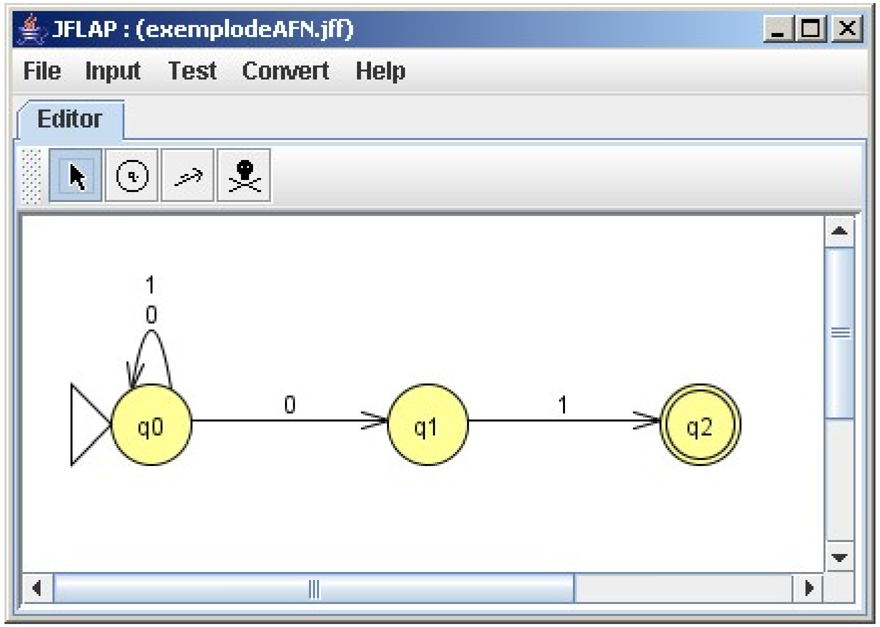
F Pois: δ_D ({**q0,q2**}, 0) = δ_N ({q0}, 0) U δ_N ({q2}, 0) → {q0,q1} U Ø = {q0,q1}

δ_D ({**q0,q2**}, 1) = δ_N ({q0},1) U δ_N ({q2},1) → {q0} U Ø = {q0}

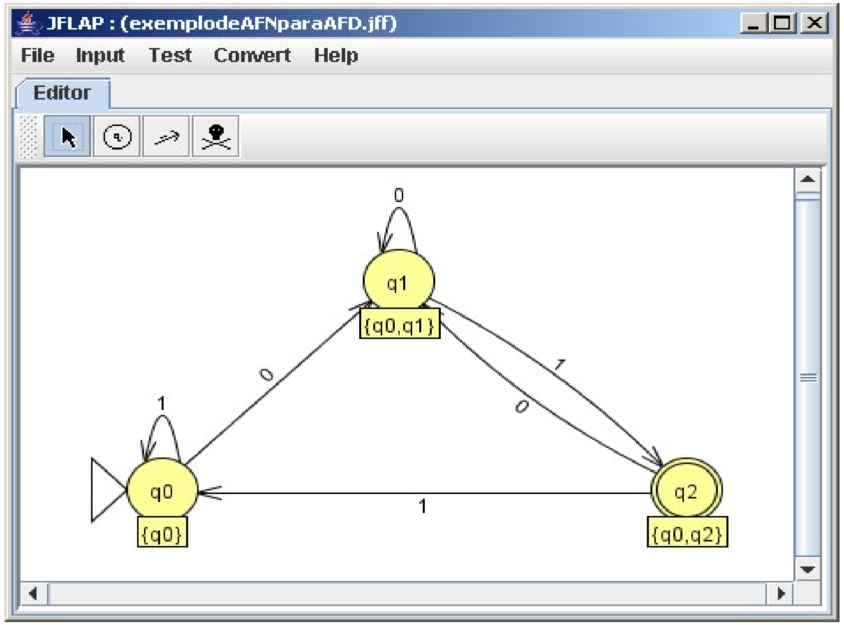
Como {q0,q1} e {q0} já foram encontrados → a simplificação pára (convergiu), conhecemos todos os estados acessíveis e suas transições

Equivalência entre AFD e AFND

AFND



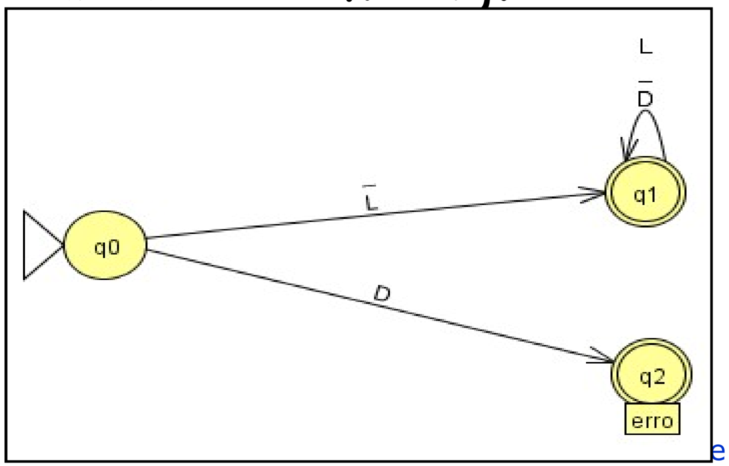
AFD



- mesmo número de estados, porém o AFD tem um número maior de transições

Prevendo entradas erradas

- A definição de um AF EXIGE que todo estado tenha uma transição para cada símbolo ($\in$ ao Σ) lido da entrada
- Podemos criar um estado de não aceitação (erro) para prever um possível dado inválido de uma determinada linguagem (AF "morre"). Ex: reconhece identificadores de variáveis em ling. C

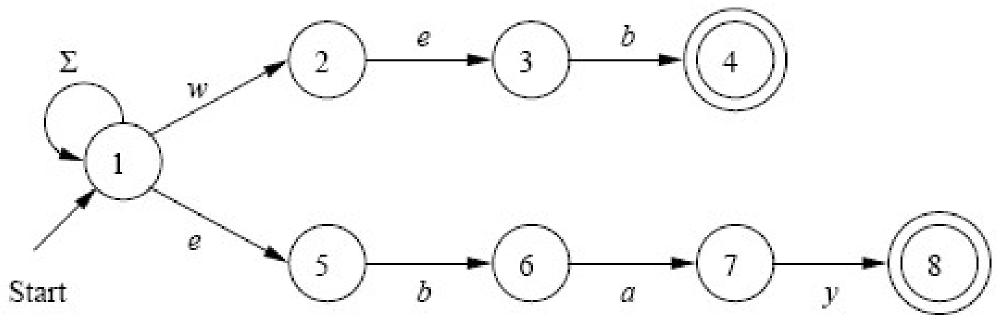


L={A..Z, a..z}

D={0..9}

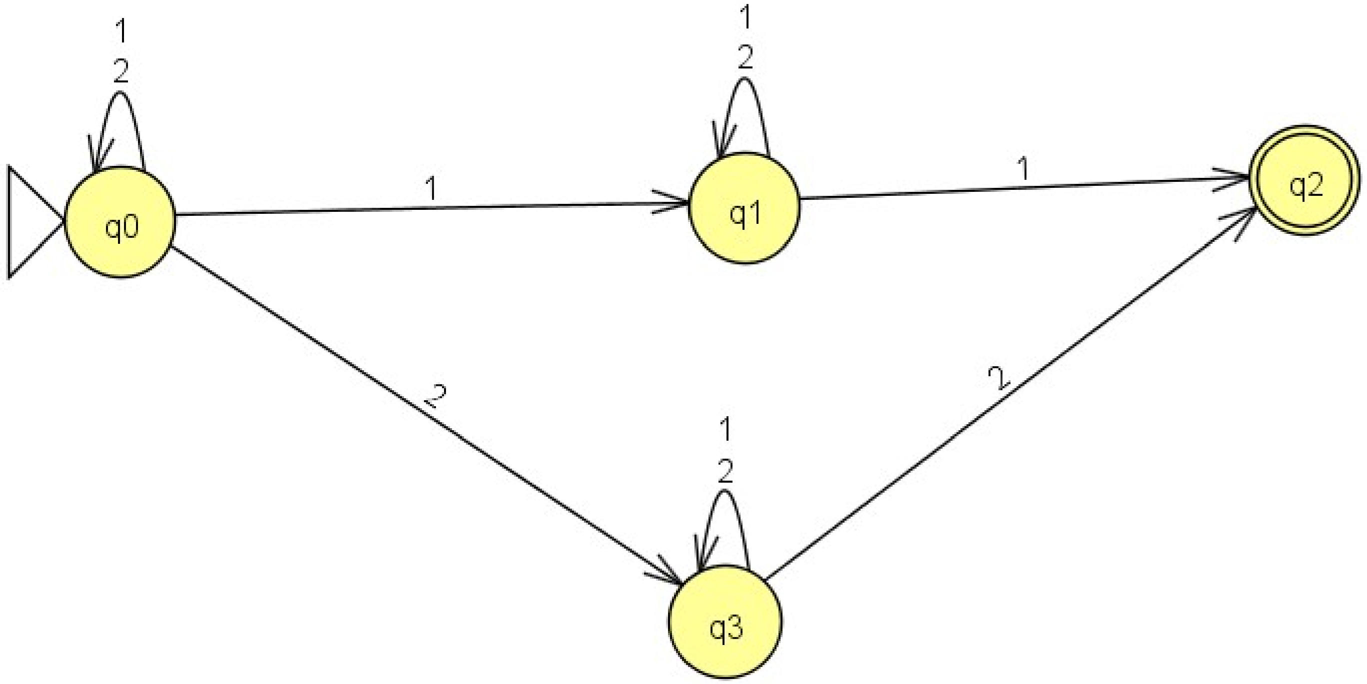
AFND para busca em textos

- Exemplo: Através de palavras-chave encontrar ocorrência de quaisquer dessas palavras em um repositório de documentos on-line.
  - Passos
    - O texto do documento é transferido um caractere de cada vez
    - O AFD deverá reconhecer as palavras-chave. Ex: web, ebay



Exercícios

1. Converta o AFND para AFD



Exercícios

2. Construa um AFND (sem mov. vazios) que seja capaz de reconhecer as seguintes palavras-chave {he, she, hers, his}. Utilize tratamento de erros, por exemplo: caso seja encontrado um caractere inválido (Σ={h, e, r, s, i}) retorne para o estado inicial, reiniciando a leitura.

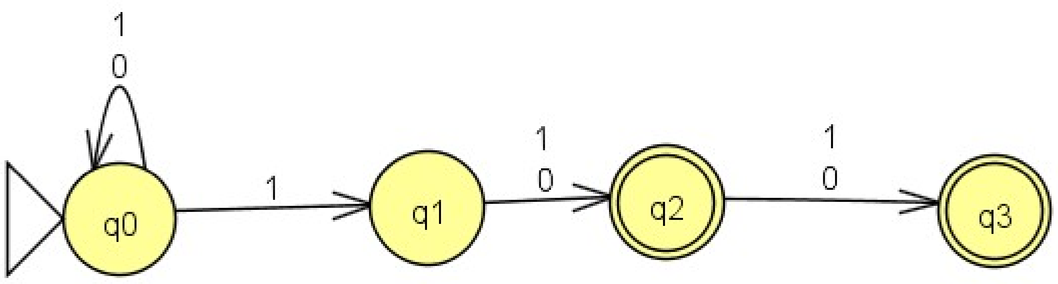
3. Converta para AFD o exercício anterior.

4. Seção 2.3 (páginas 71 a 73)

5. Seção 2.4 (página 78)

Equivalência entre AF's e ER's

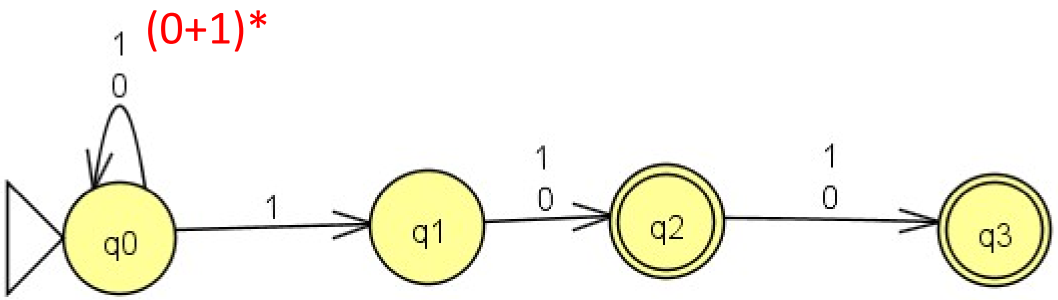
Conversão de AF para ER



ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

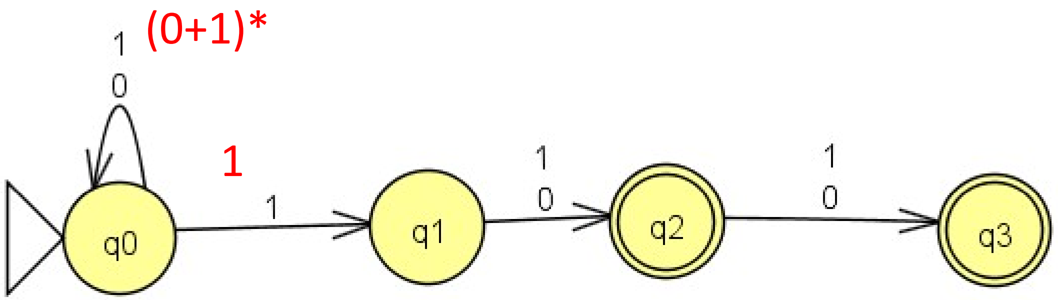
Conversão de AF para ER



ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

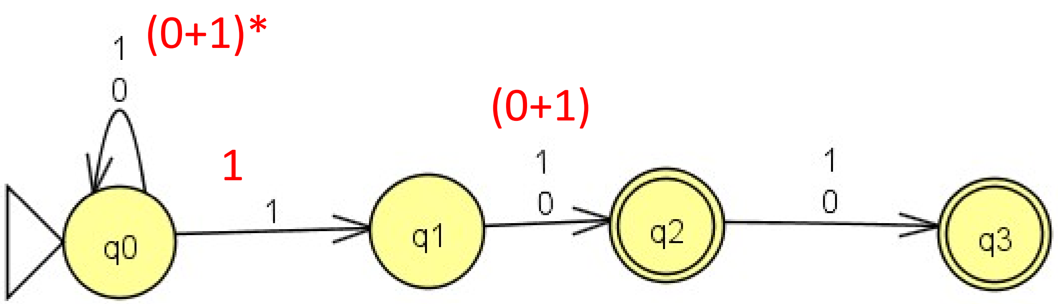
Conversão de AF para ER



ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

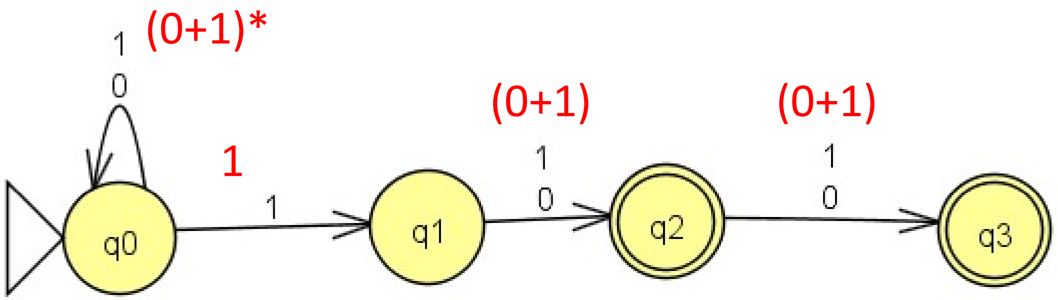
Conversão de AF para ER



ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

Conversão de AF para ER



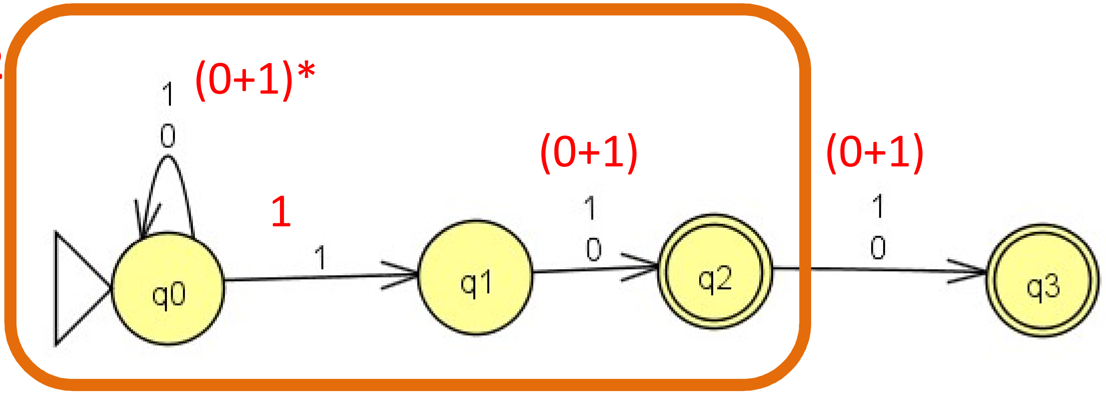
ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

Conversão de AF para ER

Passo 2
ER1: (0+1)*1(0+1)

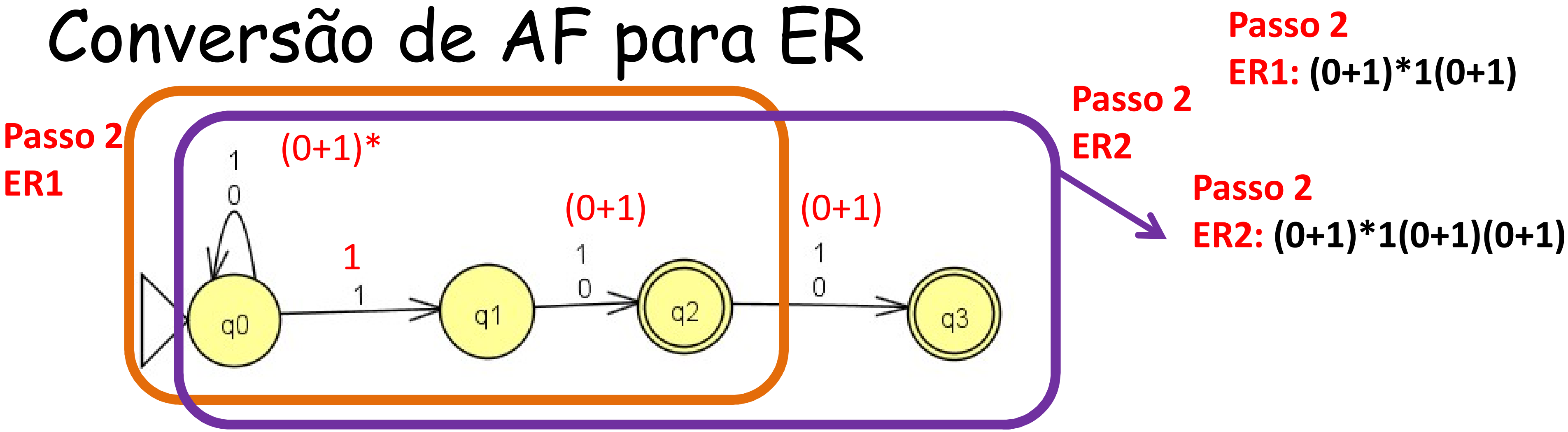
Passo 2
ER1



ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

Conversão de AF para ER

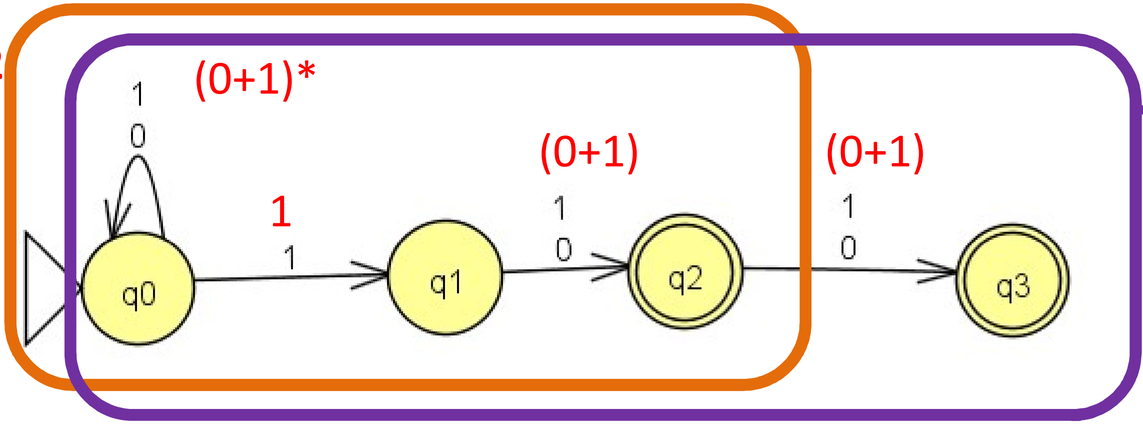


ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

Conversão de AF para ER

Passo 2
ER1



Passo 2
ER1: **(0+1)*1(0+1)**

Passo 2
ER2

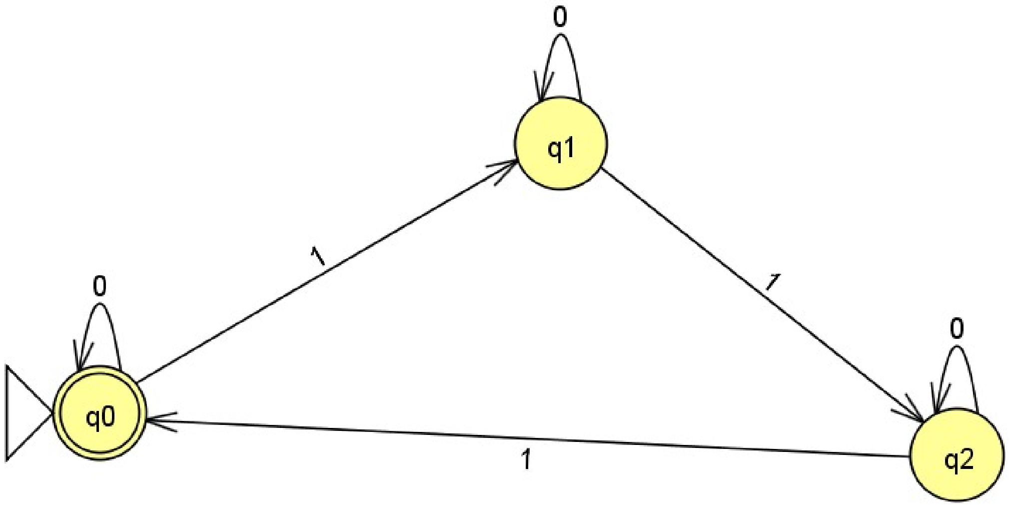
Passo 2
ER2: **(0+1)*1(0+1)(0+1)**

ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

ER final. Passo 3
ER: ER1 ∪ ER2
ER= (0+1)*1(0+1) +
(0+1)*1(0+1)(0+1)

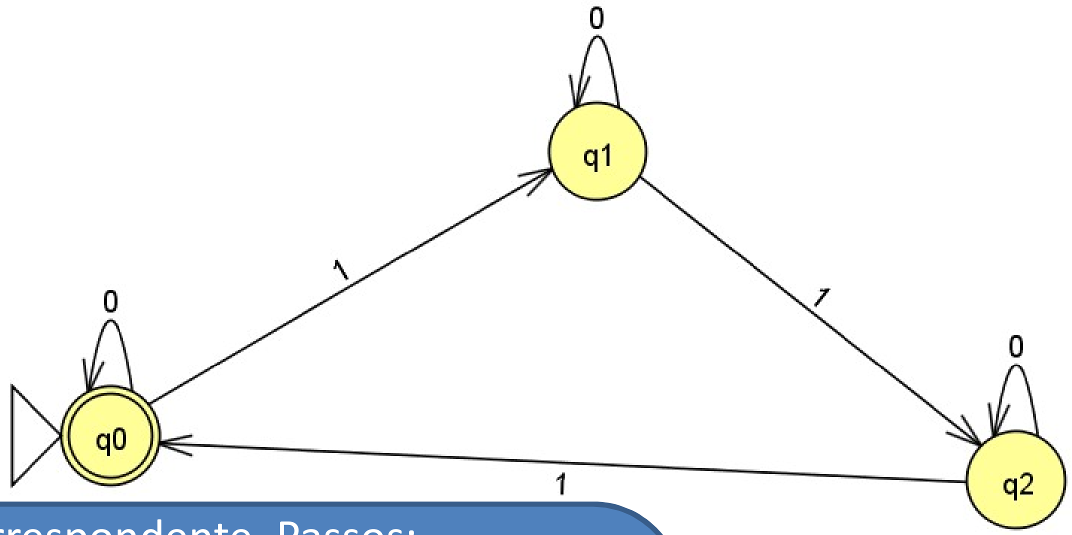
Dado o AFD que reconhece a $L=\{0^n1^m \mid n \geq 0$ e m é múltiplo de 3$\}$, encontre a ER correspondente.



ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

Dado o AFD que reconhece a $L=\{0^n1^m \mid n \geq 0 \text{ e } m \text{ é múltiplo de } 3\}$, encontre a ER correspondente.



ER
0*+(0*10*10*1)*

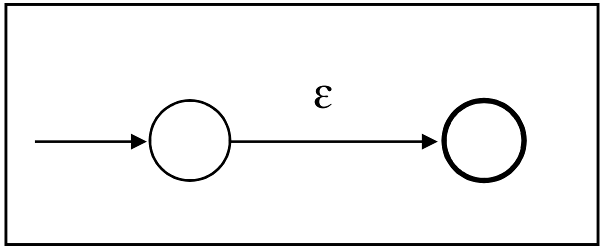
ER correspondente. Passos:

1. Encontrar a ER para cada estado
2. Encontrar, a partir da união de todas as ER's, uma ER que vai do estado inicial para o estado final
3. Caso tenha mais de um estado final, a ER resultante será a união das ER's obtidas no passo 2

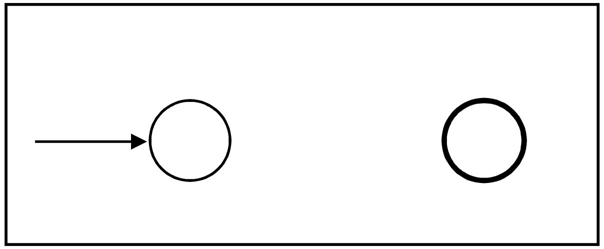
Conversão de ER's para AF's

Conversão de ER's para AF's

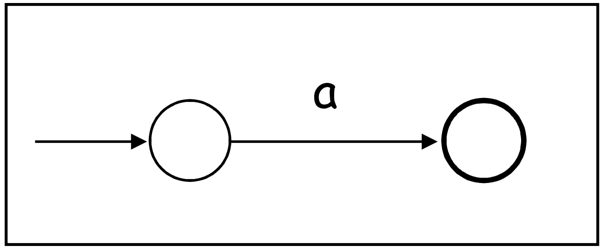
- Toda linguagem definida por um ER também é definida por um AF.
- Construção de um AF a partir de uma ER → componentes básicos:



ER = ε



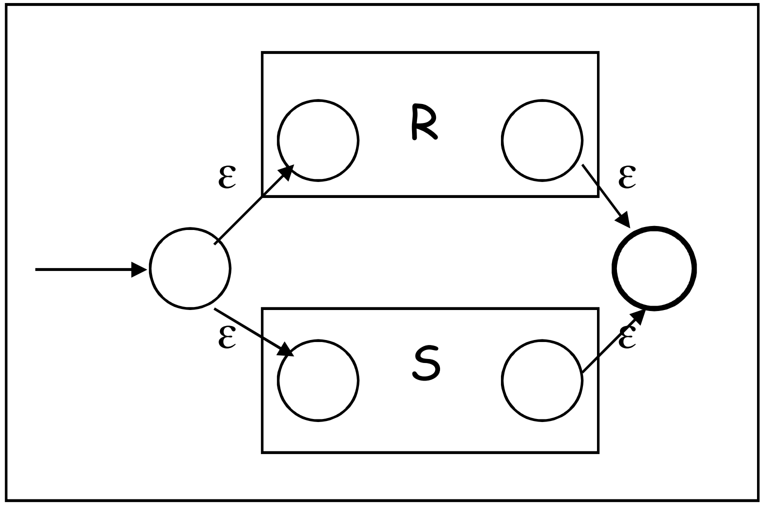
ER = θ



ER = a

Conversão de ER's para AF's

Caso a ER tenha mais de um operador (união, concatenação e fechamento).

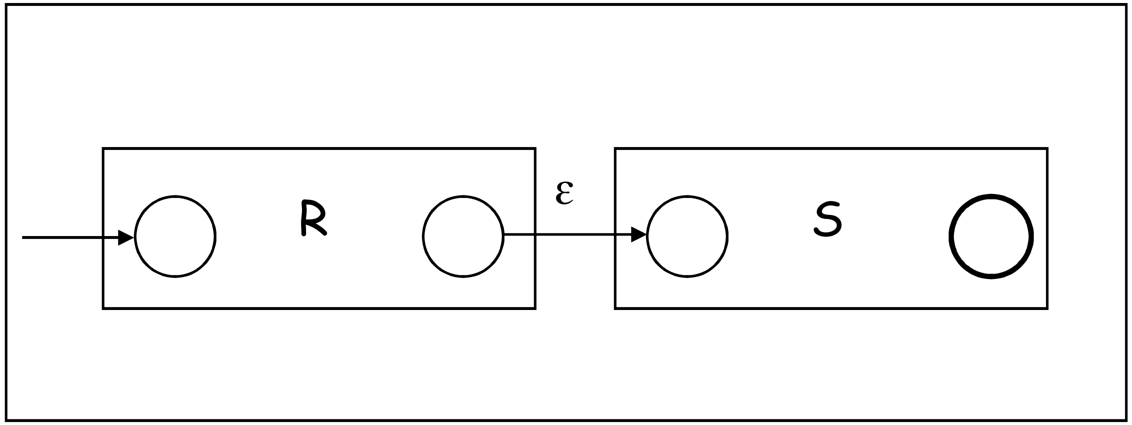


União

ER = R + S

Conversão de ER's para AF's

Caso a ER tenha mais de um operador (união, concatenação e fechamento).

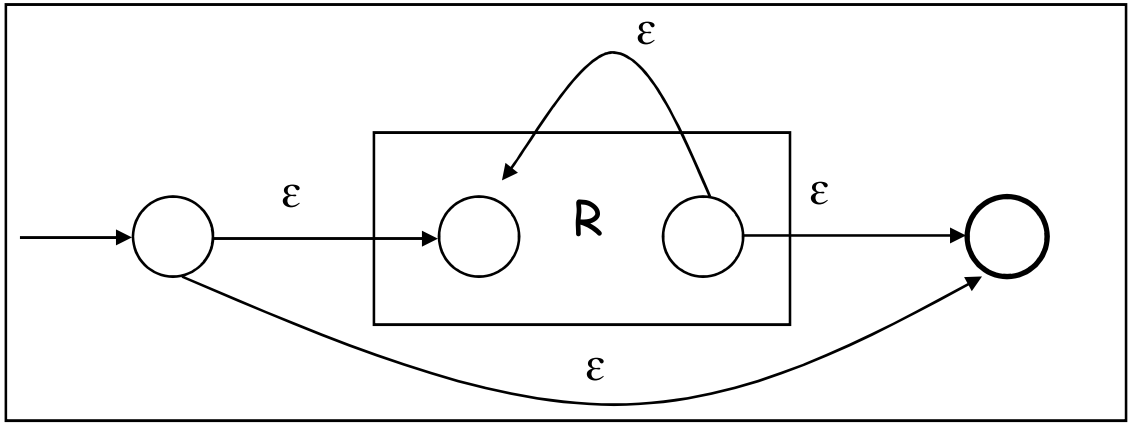


Concatenação

ER = RS

Conversão de ER's para AF's

Caso a ER tenha mais de um operador (união, concatenação e fechamento).

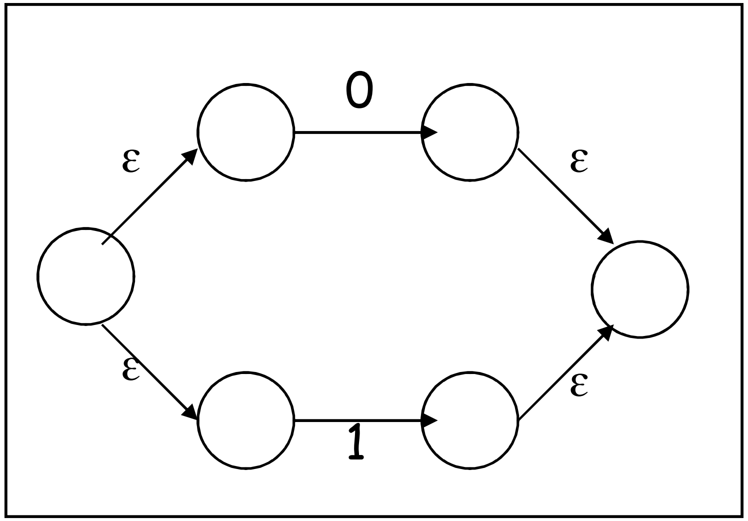


Fechamento

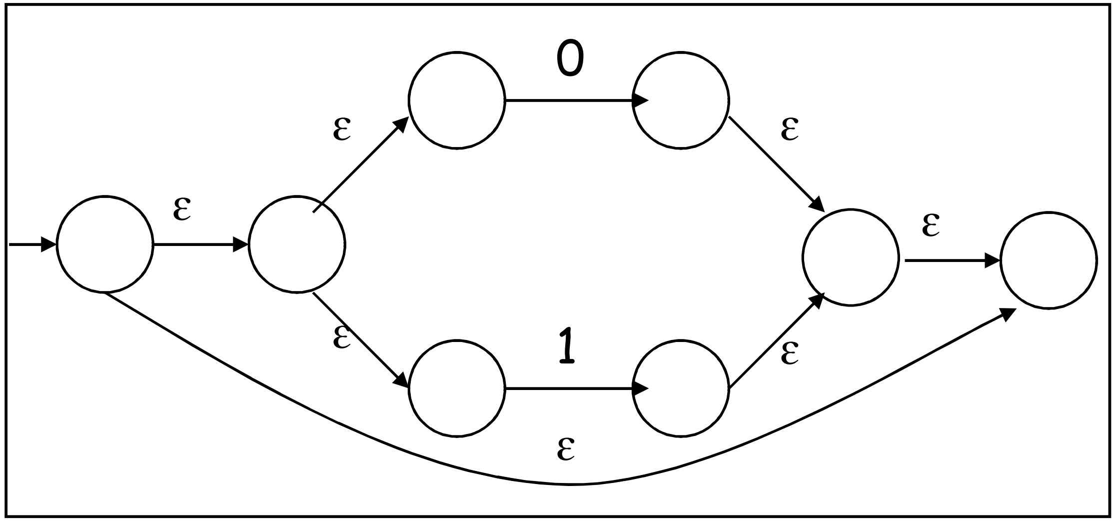
ER = R*

Conversão de ER's para AF's

Converter a ER = (0 + 1)* 1(0 + 1) em um AFND com movimentos vazios.



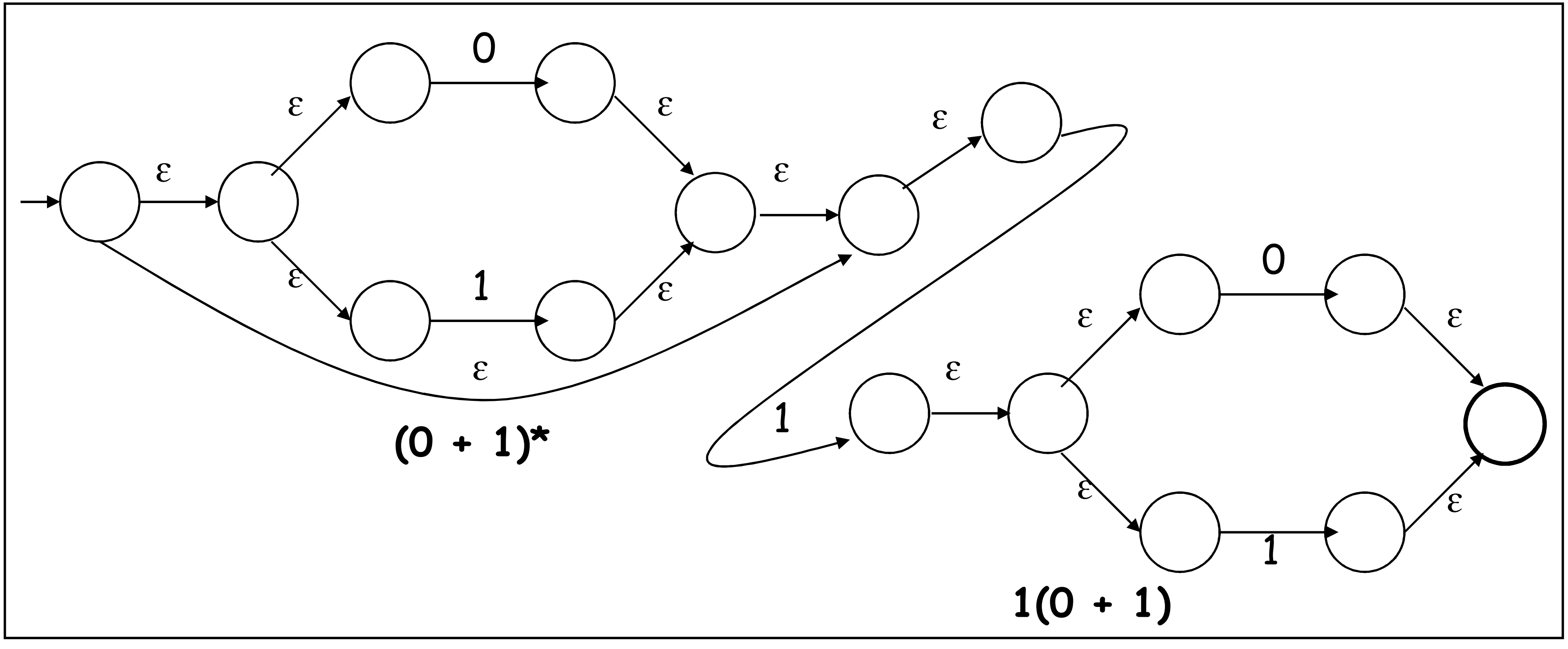
0 + 1



(0 + 1)*

Conversão de ER's para AF's

Converter a ER = (0 + 1)* 1(0 + 1) em um AFND com movimentos vazios.



Exercícios

5. Da Seção 3.2 (exercícios 3.2.1, 3.2.2, 3.2.3 e 3.2.4) – páginas 113 a 115

Conversão entre GR's e AF's

Equivalência entre GR's e AF's

- Dada uma gramática linear à direita é possível construir um autômato finito capaz de reconhecer a mesma linguagem.

> ***Seja G uma gramática linear à direita. Então é possível definir um autômato finito M de tal modo que L(G) = L(M).***

Equivalência entre GR's e AF's

- Dada a gramática G = (V,T,P,S), onde P são do tipo:

1. X → aY
2. X → Y
3. X → a
4. X → ε

- com S,Y ∈ V, a ∈ T

Algoritmo de equivalência entre GR's e AF's

1. $X \rightarrow aY$
2. $X \rightarrow Y$
3. $X \rightarrow a$
4. $X \rightarrow \varepsilon$

- com $X, Y \in V$, $a \in T$

$G = (V, T, P, S)$

- **Algoritmo de conversão GR → AF**
- **Entrada:** uma gramática linear à direita G;
- **Saída:** um autômato finito M tal que L(M) = L(G);
- **Método:**

1. Conjunto de estados:
   - Cada estado de M corresponde a um dos símbolos não-terminais de G. A esse conjunto acrescenta-se um novo símbolo (estado) $Z \notin V$, ou seja, $\{Q\} = V \cup \{Z\}$. O estado inicial de M é S, a raiz da gramática. O estado final de M é Z, o novo estado acrescentado.
2. Alfabeto de entrada:
   - O alfabeto de entrada Σ de M é o mesmo alfabeto Σ de G.

Algoritmo de equivalência entre GR's e AF's

•Algoritmo de conversão GR → AF

3. Função de transição:

•$\delta = \varnothing$;

•Para cada regra de produção em P da gramática G, e conforme seu tipo:

1. Se $X \rightarrow aY$ então $\delta = \delta \cup \{(X,a) \rightarrow Y\}$;
2. Se $X \rightarrow Y$ então $\delta = \delta \cup \{(X,\varepsilon) \rightarrow Y\}$;
3. Se $X \rightarrow a$ então $\delta = \delta \cup \{(X,a) \rightarrow Z\}$;
4. Se $X \rightarrow \varepsilon$ então $\delta = \delta \cup \{(X,\varepsilon) \rightarrow Z\}$;

$G = (V,T,P,S)$

1. $X \rightarrow aY$
2. $X \rightarrow Y$
3. $X \rightarrow a$
4. $X \rightarrow \varepsilon$

• com $X,Y \in V$, $a \in T$

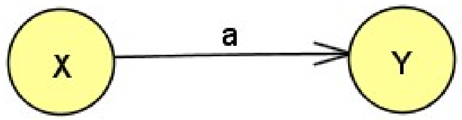
Algoritmo de equivalência entre GR's e AF's

3. Função de transição:

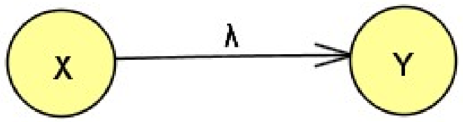
- $\delta = \emptyset$;

- Para cada regra de produção em P da gramática G, e conforme seu tipo:

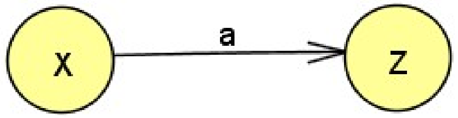
1. Se $X \rightarrow aY$ então $\delta = \delta \cup \{(X,a) \rightarrow Y\}$;



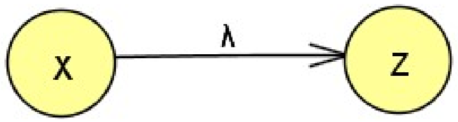
2. Se $X \rightarrow Y$ então $\delta = \delta \cup \{(X,\varepsilon) \rightarrow Y\}$;



3. Se $X \rightarrow a$ então $\delta = \delta \cup \{(X,a) \rightarrow Z\}$;



4. Se $X \rightarrow \varepsilon$ então $\delta = \delta \cup \{(X,\varepsilon) \rightarrow Z\}$;



Exemplo de equivalência entre GR's e AF's

- Seja G uma gramática linear à direita:

  $G = (V,T,P,S)$

  $V = \{S,K,L\}$

  $T = \{a,b,c\}$

  $P = \{S \rightarrow a,\ S \rightarrow aK,\ K \rightarrow bK,\ K \rightarrow L,\ L \rightarrow cL,\ L \rightarrow \varepsilon\}$

$\boxed{\delta = \delta \cup \{(S,a) \rightarrow Z\};}$

- O AF corresponde é dado por:

Exemplo de equivalência entre GR's e AF's

- Seja G uma gramática linear à direita:

$G = (V,T,P,S)$

$V = \{S,K,L\}$

$T = \{a,b,c\}$

$P = \{S \rightarrow a,\ S \rightarrow aK,\ K \rightarrow bK,\ K \rightarrow L,\ L \rightarrow cL,\ L \rightarrow \varepsilon\}$

$\boxed{\delta = \delta \cup \{(S,a) \rightarrow K\};}$

- O AF corresponde é dado por:

Exemplo de equivalência entre GR's e AF's

- Seja G uma gramática linear à direita:

$G = (V,T,P,S)$

$V = \{S,K,L\}$

$T = \{a,b,c\}$

$\delta = \delta \cup \{(K,b) \rightarrow K\};$

$P = \{S \rightarrow a,\ S \rightarrow aK,\ K \rightarrow bK,\ K \rightarrow L,\ L \rightarrow cL,\ L \rightarrow \varepsilon\}$

- O AF corresponde é dado por:

Exemplo de equivalência entre GR's e AF's

- Seja G uma gramática linear à direita:

$G = (V,T,P,S)$

$V = \{S,K,L\}$

$T = \{a,b,c\}$

$\delta = \delta \cup \{(K,\varepsilon) \rightarrow L\};$

$P = \{S \rightarrow a,\ S \rightarrow aK,\ K \rightarrow bK,\ K \rightarrow L,\ L \rightarrow cL,\ L \rightarrow \varepsilon\}$

- O AF corresponde é dado por:

Exemplo de equivalência entre GR's e AF's

- Seja G uma gramática linear à direita:

  $G = (V,T,P,S)$

  $V = \{S,K,L\}$

  $T = \{a,b,c\}$

  $P = \{S \rightarrow a,\ S \rightarrow aK,\ K \rightarrow bK,\ K \rightarrow L,\ L \rightarrow cL,\ L \rightarrow \varepsilon\}$

  $\delta = \delta \cup \{(L,c) \rightarrow L\};$

- O AF corresponde é dado por:

Exemplo de equivalência entre GR's e AF's

- Seja G uma gramática linear à direita:

  $G = (V,T,P,S)$

  $V = \{S,K,L\}$

  $T = \{a,b,c\}$

  $P = \{S \rightarrow a,\ S \rightarrow aK,\ K \rightarrow bK,\ K \rightarrow L,\ L \rightarrow cL,\ L \rightarrow \varepsilon\}$

- O AF corresponde é dado por:

$\delta = \delta \cup \{(L, \varepsilon) \rightarrow Z\};$

Exemplo de equivalência entre GR's e AF's

- Seja G uma gramática linear à direita:

$G = (V,T,P,S)$

$V = \{S,K,L\}$

$T = \{a,b,c\}$

$P = \{S \rightarrow a,\ S \rightarrow aK,\ K \rightarrow bK,\ K \rightarrow L,\ L \rightarrow cL,\ L \rightarrow \varepsilon\}$

$\delta = \delta \cup \{(S,a) \rightarrow Z\};$

$\delta = \delta \cup \{(S,a) \rightarrow K\};$

$\delta = \delta \cup \{(K,b) \rightarrow K\};$

$\delta = \delta \cup \{(K,\varepsilon) \rightarrow L\};$

$\delta = \delta \cup \{(L,c) \rightarrow L\};$

$\delta = \delta \cup \{(L,\varepsilon) \rightarrow Z\};$

- O AF corresponde é dado por:

Exemplo de equivalência entre GR's e AF's

- Seja G uma gramática linear à direita:

G = (V,T,P,S)

V = {S,K,L}

T = {a,b,c}

P = {S→a, S→aK, K→bK, K→L, L→cL, L→ε}

$\delta = \delta \cup \{(S,a) \rightarrow Z\};$

$\delta = \delta \cup \{(S,a) \rightarrow K\};$

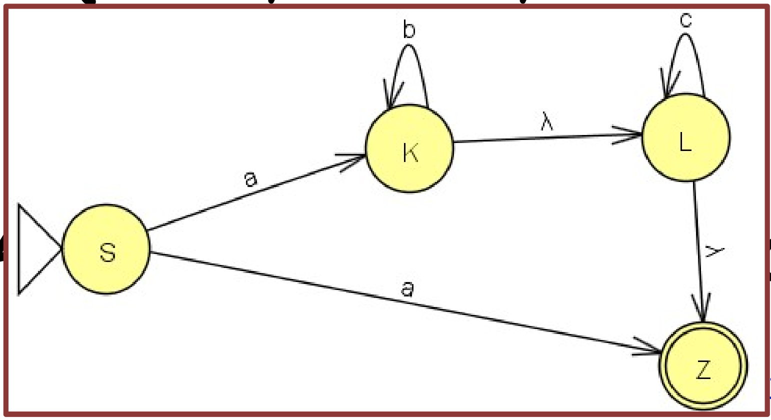
$\delta = \delta \cup \{(K,b) \rightarrow K\};$

$\delta = \delta \cup \{(K,\varepsilon) \rightarrow L\};$

$\delta = \delta \cup \{(L,c) \rightarrow L\};$

$\delta = \delta \cup \{(L, \varepsilon) \rightarrow Z\};$

- O A ado por:



- Seja G uma gramática linear à direita:

  G = (V,T,P,S)

  V = {S,K,L}

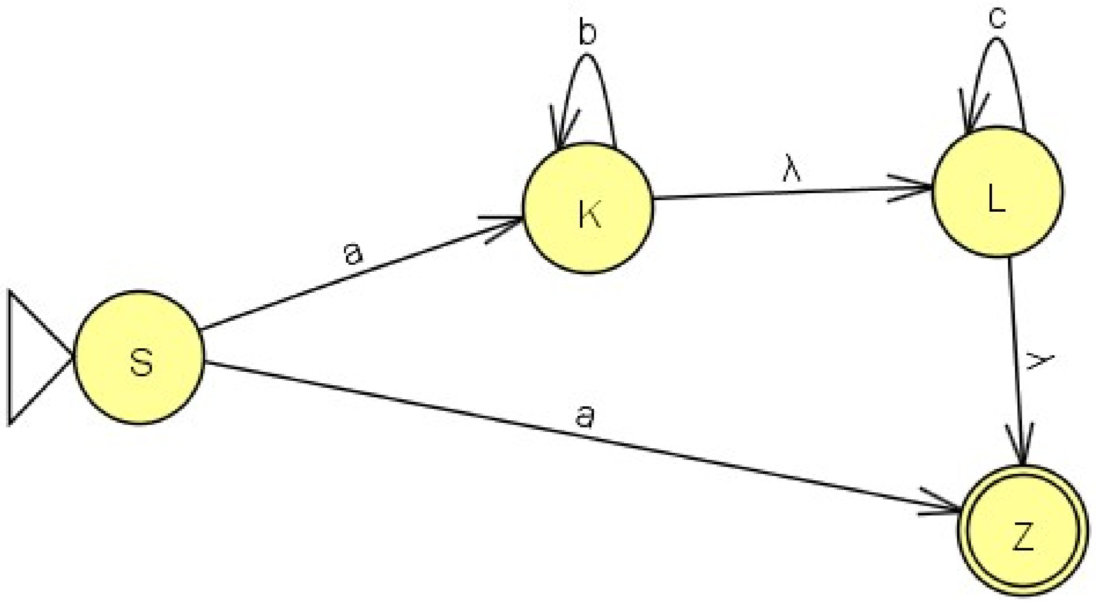
  T = {a,b,c}

  P = {S→a, S→aK, K→bK, K→L, L→cL, L→ε}

Qual a L(G) e a L(M)?

- O AF corresponde

  é dado por:



- Seja G uma gramática linear à direita:

G = (V,T,P,S)

V = {S,K,L}

T = {a,b,c}

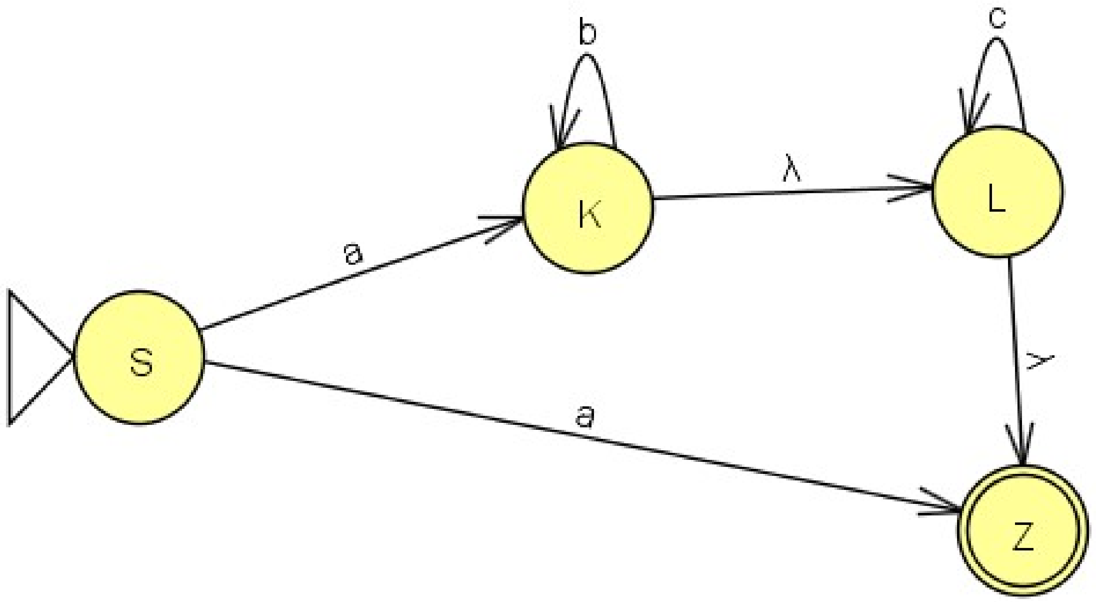
P = {S→a, S→aK, K→bK, K→L, L→cL, L→ε}

Qual a L(G) e a L(M)?

R.: ab*c*

- O AF corresponde

é dado por:



- Seja G uma gramática linear à direita:

  G = (V,T,P,S)

  V = {S,K,L}

  T = {a,b,c}

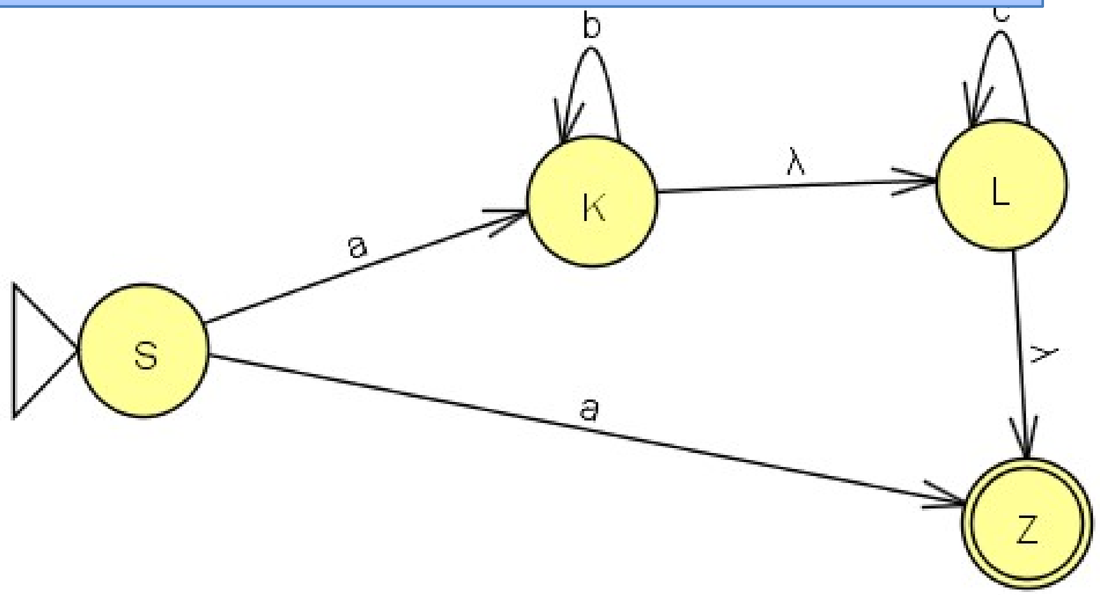
Qual a L(G) e a L(M)?

R.: ab*c*

Seja G uma gramática linear à direita. Então é possível definir um autômato finito M de tal modo que L(G) = L(M).

- O AF corresponde

  é dado por:



- Seja G uma gramática linear à direita:

  G = (V,T,P,S)

  V = {S,K,L}

  T = {a,b,c}

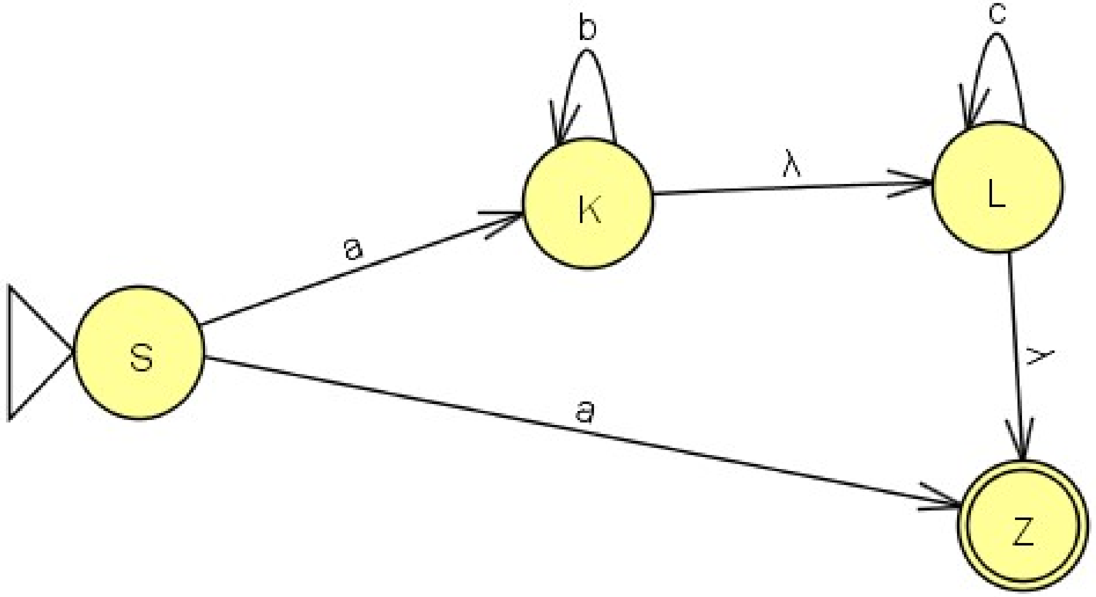
  P = {S→a, S→aK, K→bK, K→L, L→cL, L→ε}

Qual a L(G) e a L(M)?

R.: ab*c*

Exemplo de sentença aceita por L: *abbcc*

- O AF corresponde é dado por:



Conversão entre AF's e GR's

Equivalência entre AF's e GR's

Seja *M um autômato finito qualquer. Então é possível* definir uma gramática linear à direita *G, de tal modo que L(M) = L(G).*

- Dado M = ({Q}, Σ, δ, q0, {F}) um AFND-ε é possível construir uma gramática linear à direita G = (V, T, P, S) a partir de M.

Algoritmo de equivalência entre AF's e GR's

- Algoritmo de conversão AF → GR
- **Entrada:** um autômato finito M;
- **Saída:** uma gramática linear à direita G tal que L(G) = L(M);
- Método:

  1. Definição do conjunto de símbolos não-terminais:

  - Os símbolos não-terminais de G correspondem aos estados de M. A raiz da gramática é q0 (estado inicial).

  2. Alfabeto de entrada:

  - O alfabeto Σ de G é o próprio alfabeto de entrada Σ de M.

Algoritmo de equivalência entre AF's e GR's

•Algoritmo de conversão AF ➔ GR

3. Produções:

- $P \leftarrow \varnothing$;
- Para cada elemento de δ do AFND-ε M, e conforme o tipo das transições de M:
  - •1 Se $\delta(X,a) = Y$, então P {X → aY};
  - •2 Se $\delta(X,\varepsilon) = Y$, então P {X→Y}.
- Para cada elemento de Q do AFND-ε M:
  - •1 Se $X \in F$, então P {X → ε}.

Se X é um estado final

Equivalência entre AF's e GR's

- Exemplo: Dado o AFND-ε M definido e representado abaixo

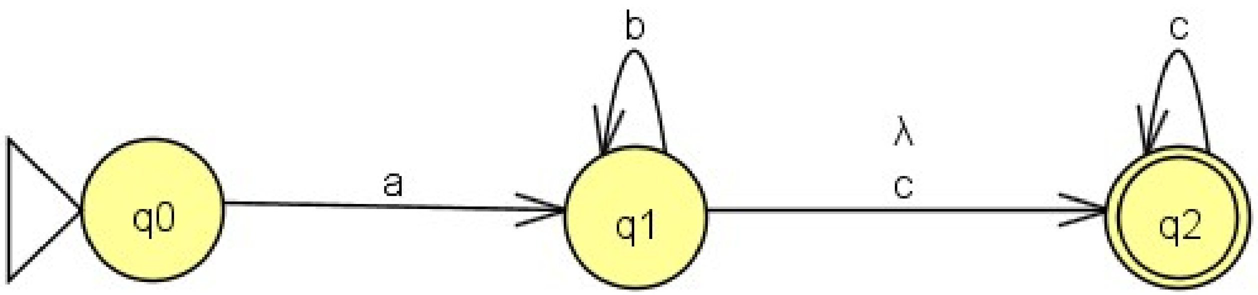
$M = (\{Q\}, \Sigma, \delta, q0, \{F\})$

$Q = \{q0, q1, q2\}$

$\Sigma = \{a,b,c\}$

$\delta = \{(q0,a)=q1, (q1,b)=q1, (q1,c)=q2, (q1,\varepsilon)=q2, (q2,c)=q2\}$

$F = \{q2\}$



Equivalência entre AF's e GR's

- Aplicando-se o algoritmo de conversão ao AFND-ε M, obtém-se a gramática linear à direita G, cujo conjunto de produções P corresponde à segunda coluna da mesma. Note que L(M) = L(G) = ab*c*.

$G = (V, \Sigma, P, q0)$
$V = \{q0,q1,q2\}$
$\Sigma = \{a,b,c\}$

•Para cada elemento de δ do AFND-ε M, e conforme o tipo das transições de M:

1. Se $\delta(X,a) = Y$, então P {$X \rightarrow aY$};
2. Se $\delta(X,\varepsilon) = Y$, então P {$X \rightarrow Y$}.

• Para cada elemento de Q do AFND-ε M:

1. Se $X \in F$, então $P \leftarrow \{X \rightarrow \varepsilon\}$.

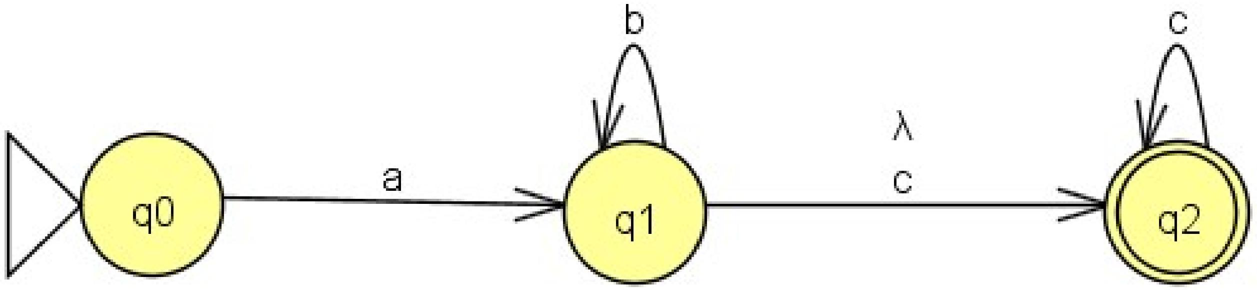
$G = (V, \Sigma, P, q0)$
$V = \{q0,q1,q2\}$
$\Sigma = \{a,b,c\}$

δ		P
$\delta(q0,a)=q1$	⟺	$q0 \rightarrow aq1$
$\delta(q1,b)=q1$	⟺	$q1 \rightarrow bq1$
$\delta(q1,c)=q2$	⟺	$q1 \rightarrow cq2$
$\delta(q1,\varepsilon)=q2$	⟺	$q1 \rightarrow q2$
$\delta(q2,c)=q2$	⟺	$q2 \rightarrow cq2$
{Q}		**P**
$q2 \in F$	⟺	$q2 \rightarrow \varepsilon$

G= ({A, B, C}, {a,b,c}, P, A)
P:{
$A \rightarrow aB$
$B \rightarrow bB$
$B \rightarrow cC$
$B \rightarrow C$
$C \rightarrow cC$
$C \rightarrow \varepsilon$
}

Renomeando os estados

$q0 \rightarrow A$
$q1 \rightarrow B$
$q2 \rightarrow C$



$G = (\{A, B, C\}, \{a,b,c\}, P, A)$

P:{

$A \rightarrow aB$

$B \rightarrow bB$

$B \rightarrow cC$

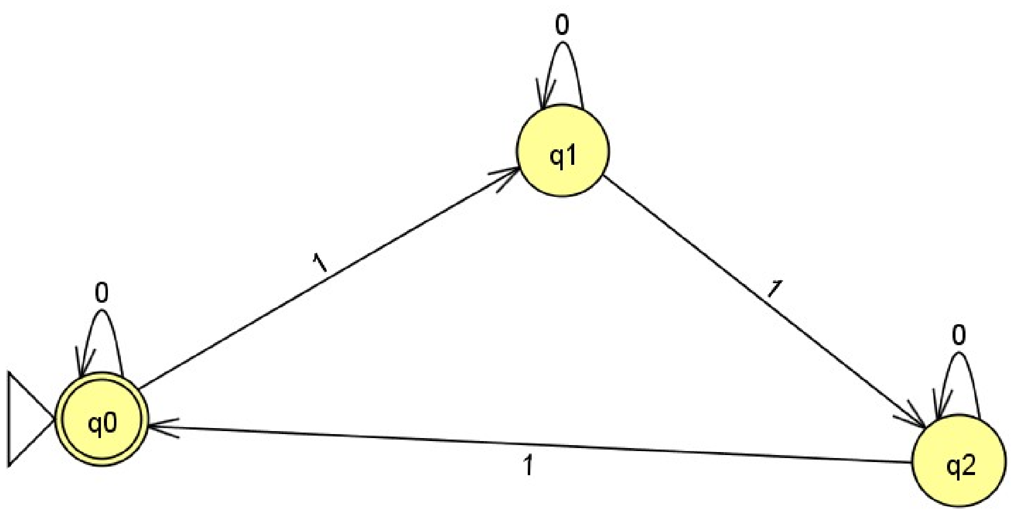
$B \rightarrow C$

$C \rightarrow cC$

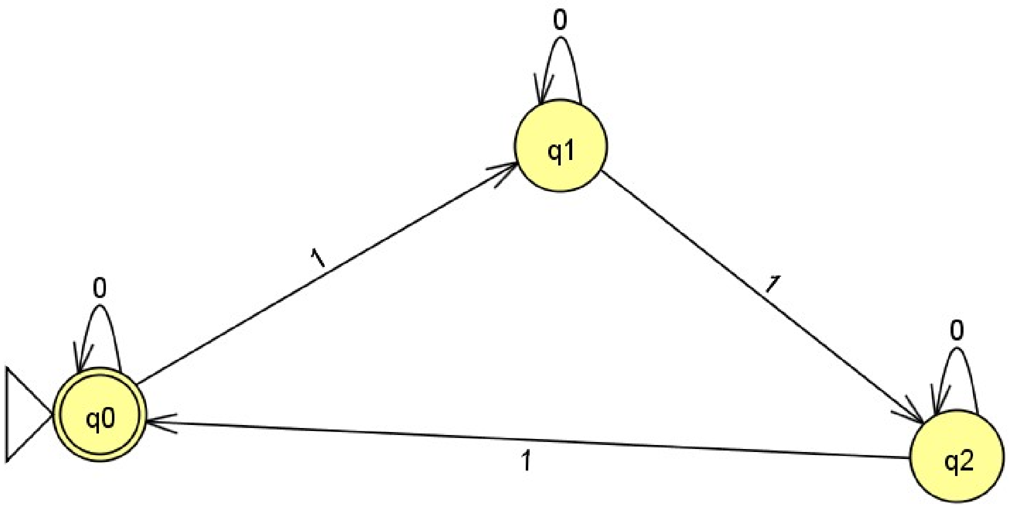
$C \rightarrow \varepsilon$

}

Dado o AFD que reconhece a $L=\{0^n1^m \mid n \geq 0$ e m é múltiplo de 3$\}$, encontre a GR correspondente.



Dado o AFD que reconhece a $L=\{0^n1^m \mid n \geq 0$ e m é múltiplo de 3$\}$, encontre a GR correspondente.



δ

$\delta(q0,0)=q0$

$\delta(q0,1)=q1$

$\delta(q1,0)=q1$

$\delta(q1,1)=q2$

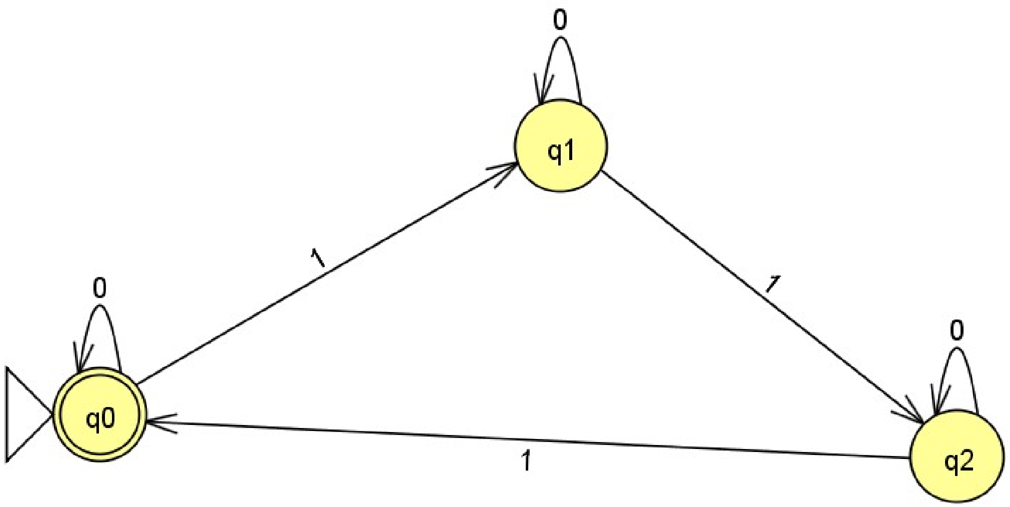
$\delta(q2,0)=q2$

$\delta(q2,1)=q0$

{Q}

$q0 \in F$

Dado o AFD que reconhece a $L=\{0^n1^m \mid n \geq 0$ e m é múltiplo de 3$\}$, encontre a GR correspondente.



δ		P
$\delta(q0,0)=q0$	$\Leftrightarrow$	$q0 \rightarrow 0q0$
$\delta(q0,1)=q1$	$\Leftrightarrow$	$q0 \rightarrow 1q1$
$\delta(q1,0)=q1$	$\Leftrightarrow$	$q1 \rightarrow 0q1$
$\delta(q1,1)=q2$	$\Leftrightarrow$	$q1 \rightarrow 1q2$
$\delta(q2,0)=q2$	$\Leftrightarrow$	$q2 \rightarrow 0q2$
$\delta(q2,1)=q0$	$\Leftrightarrow$	$q2 \rightarrow 1q0$

{Q}	P
$q0 \in F$	$q0 \rightarrow \varepsilon$

Dado o AFD que reconhece a $L=\{0^n1^m \mid n \geq 0$ e m é múltiplo de $3\}$, encontre a GR correspondente.

δ		P
$\delta(q0,0)=q0$	⟺	$q0 \rightarrow 0q0$
$\delta(q0,1)=q1$	⟺	$q0 \rightarrow 1q1$
$\delta(q1,0)=q1$	⟺	$q1 \rightarrow 0q1$
$\delta(q1,1)=q2$	⟺	$q1 \rightarrow 1q2$
$\delta(q2,0)=q2$	⟺	$q2 \rightarrow 0q2$
$\delta(q2,1)=q0$	⟺	$q2 \rightarrow 1q0$
{Q}		**P**
$q0 \in F$		$q0 \rightarrow \varepsilon$

$G = (\{A, B, C\}, \{0,1\}, P, A)$

P:{

$A \rightarrow 0A$

$A \rightarrow 1B$

$B \rightarrow 0B$

$B \rightarrow 1C$

$C \rightarrow 0C$

$C \rightarrow 1A$

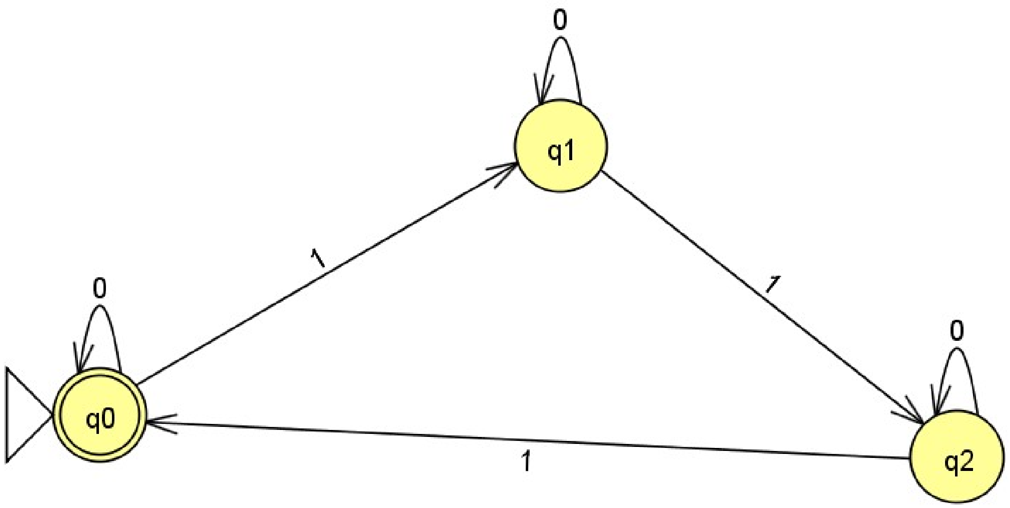
$A \rightarrow \varepsilon$

}

Renomeando os estados

$q0 \rightarrow A$
$q1 \rightarrow B$
$q2 \rightarrow C$

Dado o AFD que reconhece a L={0^n1^m | n≥ 0 e m é múltiplo de 3}, encontre a GR correspondente.



G= ({A, B, C}, {0,1}, P, A)
P:{

A→0A

A→1B

B→0B

B→1C

C→0C

C→1A

A→ε

}

Renomeando os estados

q0 → A
q1 → B
q2→ C

Exercícios

- Escolha 5 enunciados (dos 39 propostos) dos exercícios da Aula 3 e aplique os algoritmos de conversão