Autômatos Finitos

Eduardo Furlan Miranda

Baseado em: GARCIA, A. de V.; HAEUSLER, E. H. Linguagens Formais e Autômatos. Londrina: EDA, 2017.

Autômatos Finitos (AF)

- Seja a gramática G :
  - $S \rightarrow 0A \mid 1S$
  - $A \rightarrow 0S \mid 1A \mid \varepsilon$

- Ela gera as cadeias sobre o alfabeto $\Sigma = \{ 0 , 1 \}$, que têm quantidade ímpar de caracteres '0'

- P. ex., a cadeia 01010 é gerada :
  - $S \Rightarrow 0A \Rightarrow 01A \Rightarrow 010S \Rightarrow 0101S \Rightarrow 01010A \Rightarrow 01010$

Análise sintática

- Verifica se uma cadeia dada, pertence à linguagem definida pela gramática
- Dada a cadeia 01010, podemos determinar se ela está na linguagem gerada pela gramática?
- Para estruturas simples, podemos guardar sempre qual é a única variável que está no final da cadeia sendo gerada
  - Ex.: temos uma cadeia em construção: 0B, onde B é uma variável. Quando aplicamos uma regra como B → 1A, a nova cadeia seria 01A. Neste ponto, a atenção do programa se volta para o A no final, pois essa será a próxima variável a ser substituída ou avaliada

Análise sintática

- Se ao final a variável for A, como existe a regra $A \rightarrow \varepsilon$, então significa que A pode ser substituído por nada, ou seja, termina a derivação e então a cadeia gerada é validada

- A validação sintática geralmente simula esse processo, partindo da cadeia e tentando "retroceder" para o símbolo inicial (S) ou construindo a cadeia passo a passo conforme as regras da gramática

Autômato Finito Determinístico (AFD)

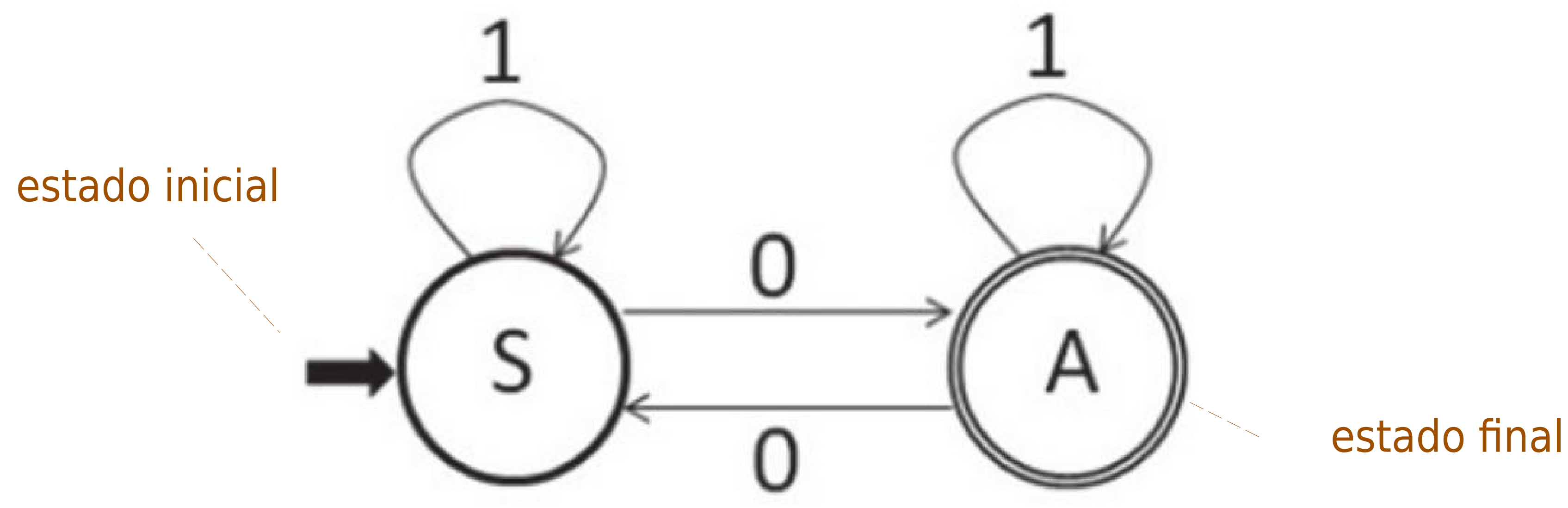


Figura 2.1

- Os círculos representam “estados”
- Inicia pelo círculo marcado com uma seta
- No final da leitura, se o estado for o do círculo duplo, a cadeia foi “reconhecida”

AFD

- O AFD está sempre em exatamente um estado, por isso é chamado de determinístico

- Representação tabular do AFD:

	0	1
→S	A	S
*A	S	A

indica estado inicial

indica estado final

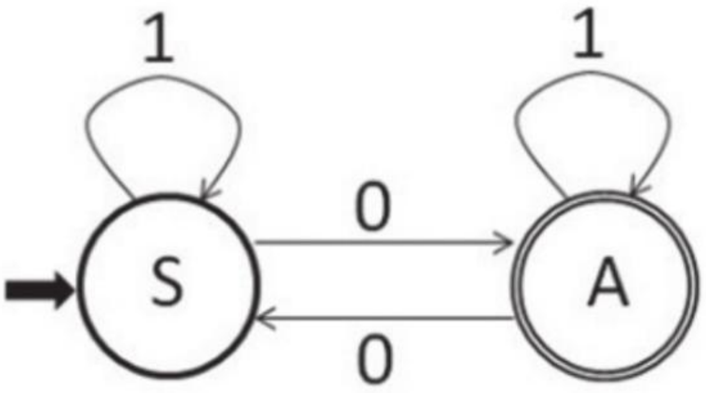
AFD, formalmente

δ = delta

- Um AFD M é uma tupla $(Q, \Sigma, \delta, q0, F)$, onde:
  - Q conjunto finito e não vazio de estados
  - Σ alfabeto de entrada
  - $\delta : Q \times \Sigma \rightarrow Q$ função de transição de estados. “Delta (δ) é uma função que mapeia o produto cartesiano de Q e Sigma (Σ) em Q”
  - $q0 \in Q$ estado inicial
  - $F \subseteq Q$ conjunto de estados finais

- Representa todos os pares possíveis (estado, símbolo).
- O produto cartesiano de dois conjuntos A e B é o conjunto de todos os pares ordenados possíveis (a, b), onde 'a' pertence a A e 'b' pertence a B.
- Ex.: Seja A = {1, 2} e B = {x, y, z}. O produto cartesiano A × B seria:
  - A × B = {(1, x), (1, y), (1, z), (2, x), (2, y), (2, z)}

Aplicando no AFD da Fig. 2.1 (slide 5)



- $Q = \{ S, A \}$ conjunto finito e não vazio de estados
- $\Sigma = \{ 0, 1 \}$ alfabeto de entrada
- $\delta : Q \times \Sigma \rightarrow Q$ é a função de transição de estados na qual:

$\delta(S, 0) = A$, $\delta(S, 1) = S$, $\delta(A, 0) = S$ e $\delta(A, 1) = A$

- S estado inicial
- $F = \{ A \}$ conjunto de estados finais

Função estendida de transição ($\hat{\delta}$)

- É quando o autômato lê uma cadeia composta de símbolos

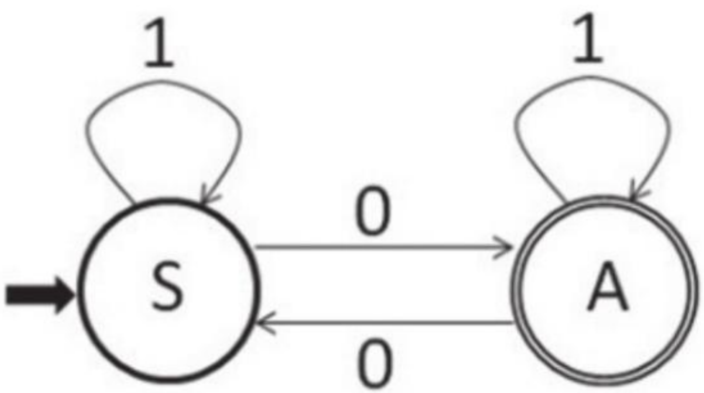
- A função $\hat{\delta}$ representa a ação do AFD ao ler uma cadeia
  - Se ao lermos um 0 temos que δ (S, 0) = A , e o AFD fica no estado A
  - Ao lermos um segundo 0, temos δ (A, 0) = S e o AFD retorna ao estado S
  - Isso significa que $\hat{\delta}$(S, 00) = S

- Formalmente podemos definir $\hat{\delta}$ em função de δ :
  - $\hat{\delta}(q, \varepsilon) = q$, para todo $q \in Q$ conjunto finito e não vazio de estados
  - $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$, para todo $q \in Q, a \in \Sigma, w \in \Sigma^*$

É a função estendida de transição, que calcula o estado final do autômato após ler toda a cadeia aw começando no estado q

Exemplo

- Aplicando esta definição ao AFD (da Fig. 2.1), com entrada 00, temos:
  - $\hat{\delta}(S, 00) = \hat{\delta}(\delta(S, 0), 0) = \hat{\delta}(A, 0) = \hat{\delta}(\delta(A, 0), \varepsilon) = \hat{\delta}(S, \varepsilon) = S$



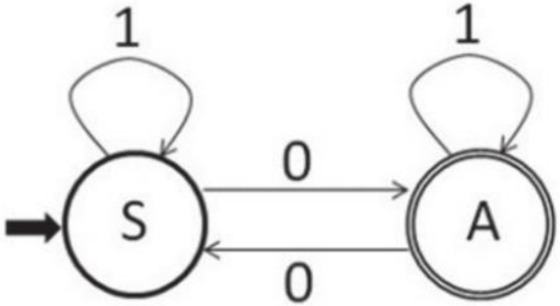
Representa a ausência de símbolos restantes na entrada. Isso ocorre no último passo do cálculo, após todos os símbolos da cadeia terem sido processados

- $\emptyset$ = conjunto vazio = {}

- ε = string vazia = “”

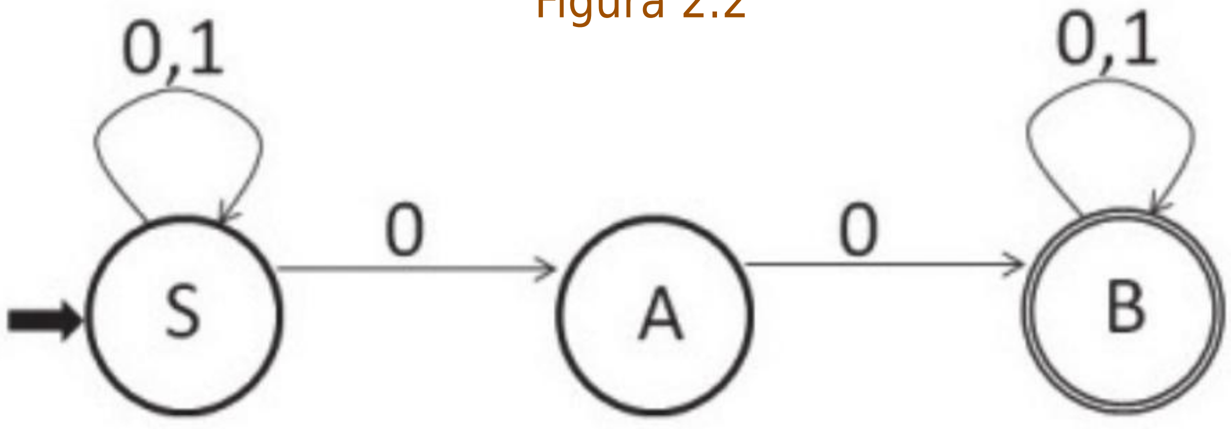
Linguagem gerada por um AFD

- Definição de uma linguagem T gerada por um AFD M :
  - Linguagem formada por todas as cadeias reconhecidas por M
    - Isto é, todas as cadeias que levam M a um estado final
  - Dado um AFD M = (Q, Σ, δ, q0 , F) , definimos a linguagem reconhecida pelo autômato M como
    - $T(M) = \{ w \in \Sigma^* \mid \hat{\delta}(q0 , w) \in F \}$
      - No caso do autômato da Fig. 2.1 temos que T(M)={0, 01, 10, 011, 101, 110, 000, 011, ... }, o conjunto de todas as cadeias com número ímpar de caracteres 0



Autômato Finito Não Determinístico (AFND)

Figura 2.2



- A gramática G2 gera as cadeias sobre o alfabeto $\Sigma = \{ 0, 1 \}$, que possuem a subcadeia “00”, e possui as regras:
  - $S \rightarrow 0S \mid 1S \mid 0A$
  - $A \rightarrow 0B$
  - $B \rightarrow 0B \mid 1B \mid \varepsilon$

A função δ retorna um conjunto de estados, ex.:

$\delta(S, 0) = \{S, A\}$

$\delta(A, 1) = \varnothing$

Ambos os casos violam a condição AFD de que para cada estado e símbolo de entrada as setas nos levem a exatamente um estado

(continua)

Representação tabular do AFND

indica estado inicial

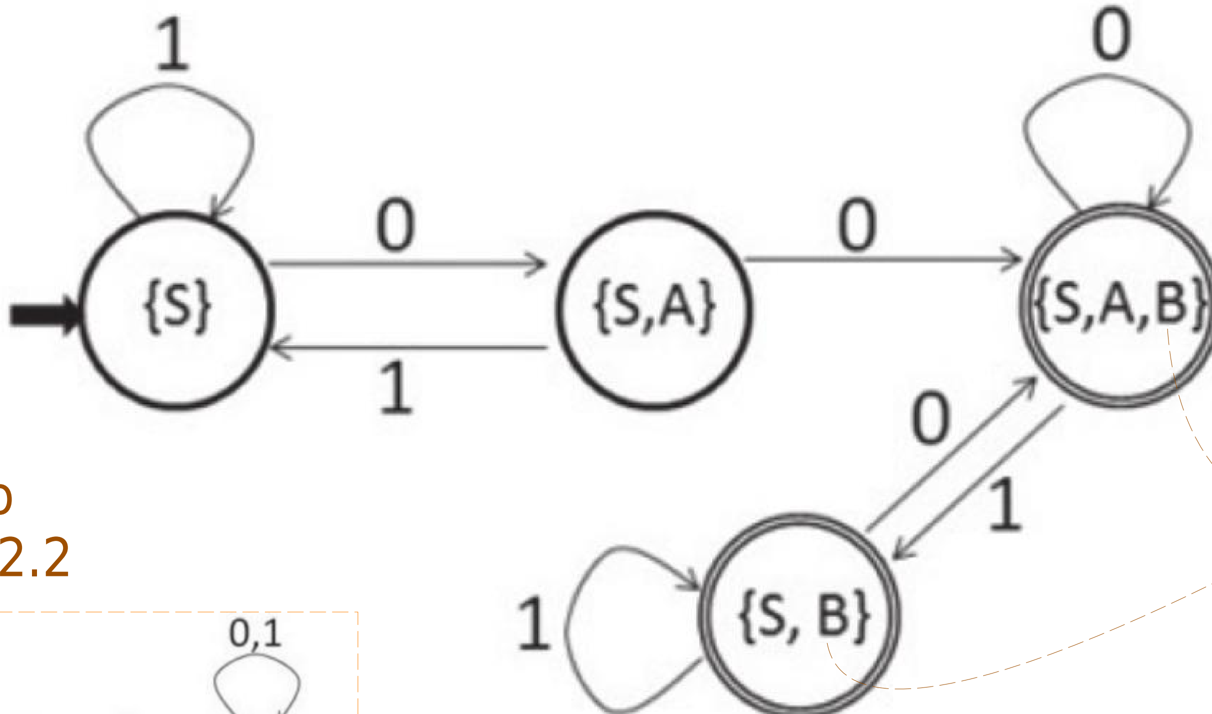
indica estado final

	0	1
$\rightarrow S$	$\{S, A\}$	$\{S\}$
A	B	$\varnothing$
$*B$	B	B

- Podemos interpretar um AFND como podendo seguir diversos caminhos ao ler uma entrada

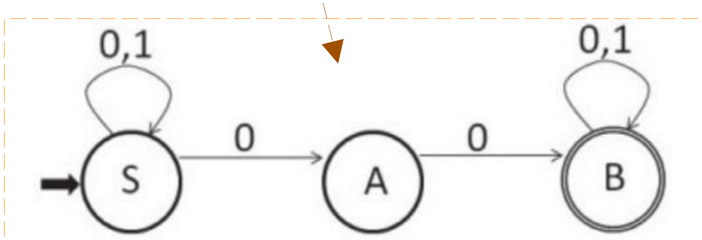
AFD obtido a partir de AFND

Usando o algoritmo de construção de subconjuntos



Os estados finais do AFD correspondem aos conjuntos de estados do AFND que contém ao menos um estado final (neste caso, B)

AFD obtido do AFND da Fig. 2.2



- Cada estado do AFD corresponde a todos os possíveis estados que o AFND estaria após ler determinada entrada

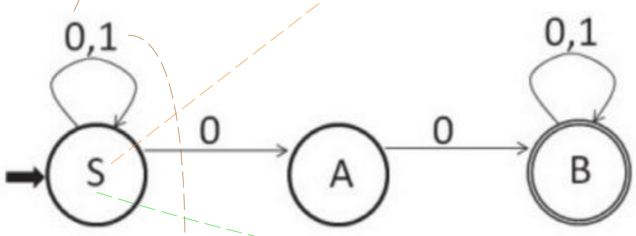
(continua)

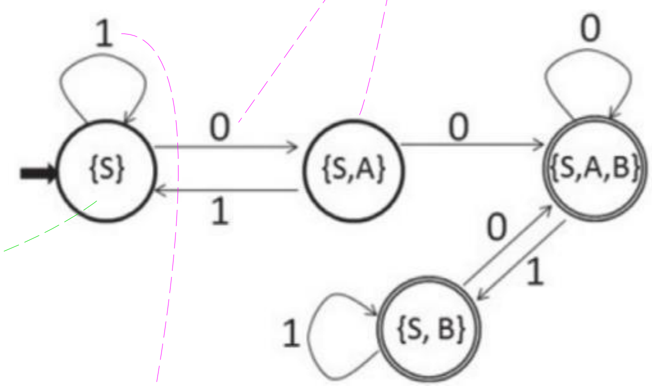
AFD obtido a partir de AFND

Passos 1 e 2

- Transições a partir de {S}:
  - Com 0: S vai para S ou A, então a transição é para {S, A}.

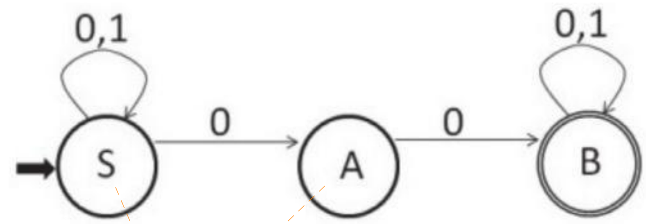
cria um novo
estado no AFD





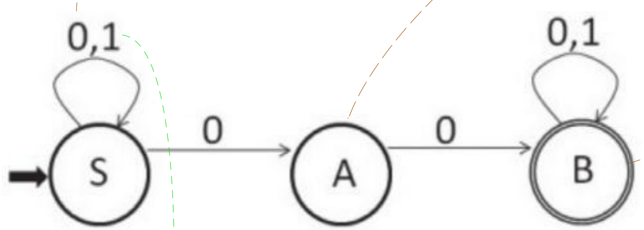
Passo 1
Usamos o mesmo
estado inicial

  - Com 1: S vai para S, então a transição é para {S}.

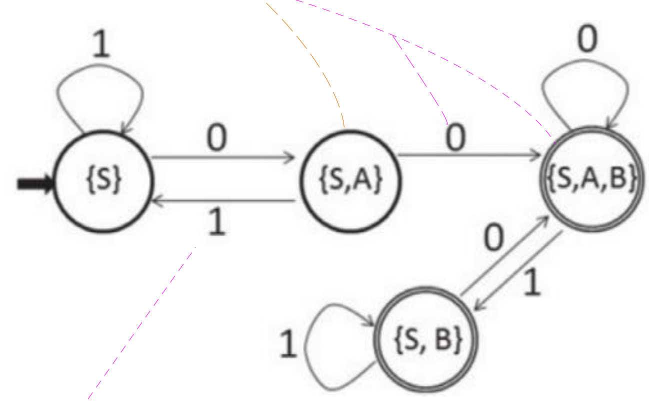


Passo 3

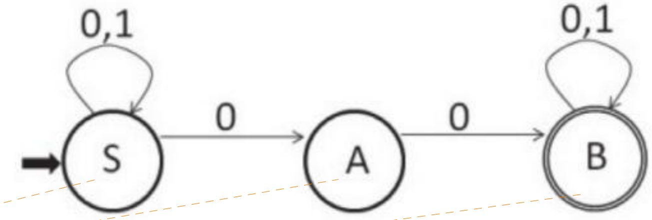
- Transições a partir no novo estado {S, A}:
  - Com 0: S vai para S ou A, e A vai para B, então a transição é para {S, A, B}.



cria um novo
estado no AFD

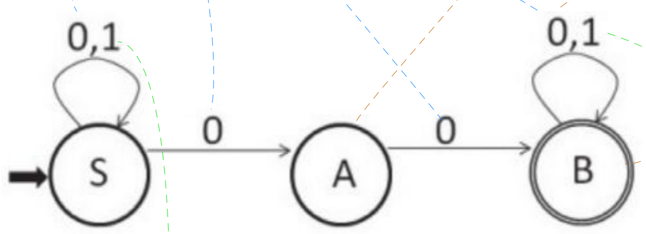


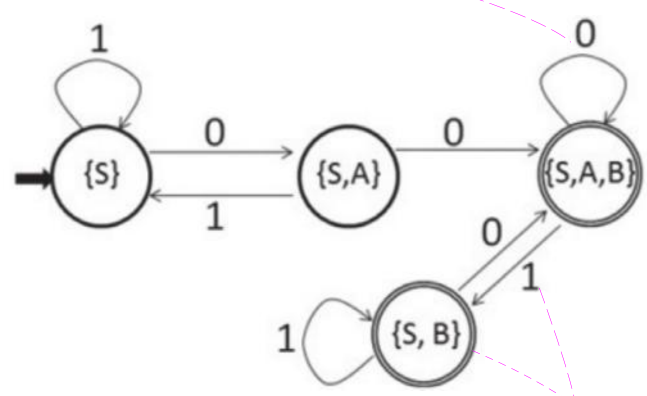
  - Com 1: S vai para S, então a transição é para {S}.



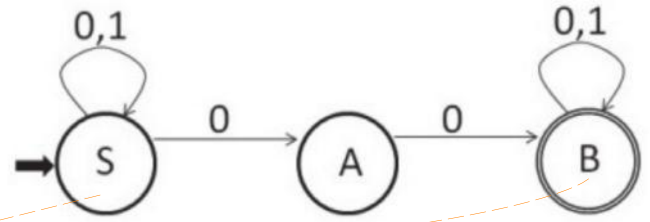
Passo 4

- Transições a partir de {S, A, B}:
  - Com 0: S vai para S ou A, A vai para B, e B vai para B, então a transição é para {S, A, B}.



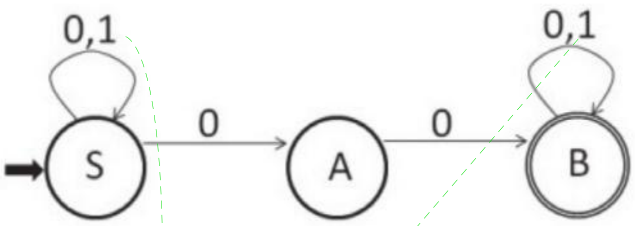


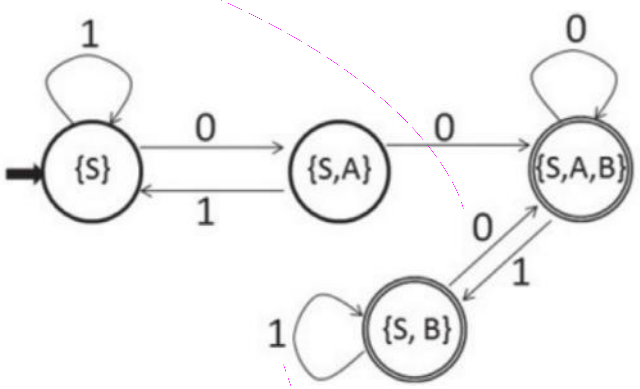
  - Com 1: S vai para S, e B vai para B, então a transição é para {S, B}.



Passo 5

- Transições a partir de {S, B}:
  - Com 0: S vai para S ou A, e B vai para B, então a transição é para {S, A, B}.

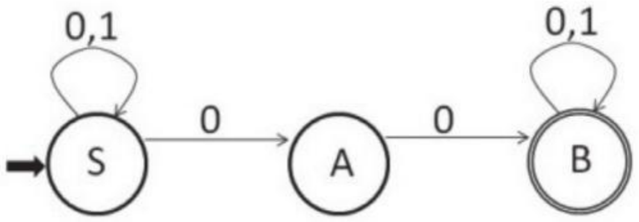


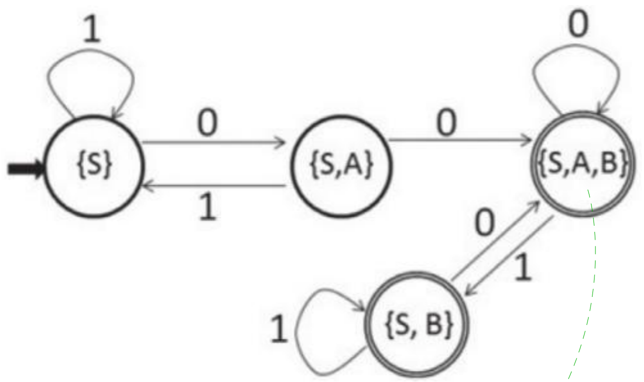


  - Com 1: S vai para S, e B vai para B, então a transição é para {S, B}.

Combinação de transições - exemplo

(já visto nos slides anteriores)





- Transições a partir de S com 0
  - Pela regra S→0S, temos S
  - Pela regra S→0A, temos A
  - Assim, $\delta(S, 0) = \{S, A\}$
- Transições a partir de A com 0
  - Pela regra A→0B, temos B
  - Assim, $\delta(A, 0) = \{B\}$
- Combinação das transições
  - $\delta_2(\{S, A\}, 0) = \delta(S, 0) \cup \delta(A, 0) = \{S, A\} \cup \{B\} = \{S, A, B\}$

(continua)

Combinação de transições - exemplo

- Se $\delta 2$ é a função de transição do AFD que desejamos obter,

  $\delta 2(\{S, A\}, 0) = \bigcup_{q \in \{S,A\}} \delta(q, 0) =$

  $= \delta(S, 0) \cup \delta(A, 0) = \{S, A\} \cup \{B\} =$

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

  cálculo de todas as transições possíveis do conjunto de estados {S,A} ao receber o símbolo 0

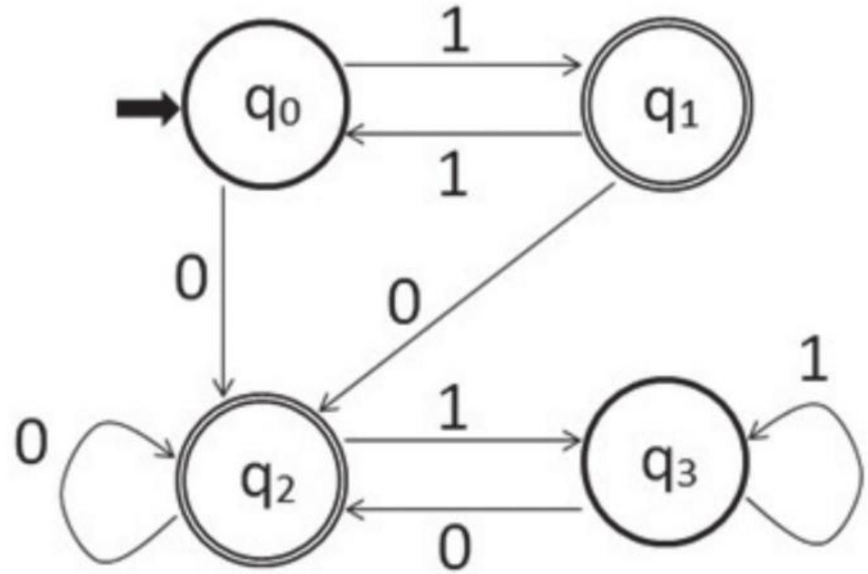
  - $\delta 2(\{S, A\}, 0)$: representa o estado resultante da transição do estado composto {S, A} em um AFD, dado a entrada 0
  - $\cup_{q \in \{S,A\}} \delta(q, 0)$: indica a união de todos os conjuntos $\delta(q, 0)$ onde q pertence ao conjunto {S, A}
    - Une os resultados das transições para cada estado q em {S, A}
  - $\delta(S, 0) \cup \delta(A, 0)$: indica que estamos calculando as transições individuais dos estados S e A no AFN original com a entrada 0 e,
    - em seguida, unindo os conjuntos de estados resultantes

Equivalência de estados

- Dado um AFD $M = (Q, \Sigma, \delta, q0, F)$, e estados $q, r \in Q$, definimos que q e r são equivalentes $(q \equiv r)$ quando para todo w, tanto $\delta(q, w)$ quanto $\delta(r, w)$ levam a estados finais ou ambos a não-finais
  - Eles podem ser unidos em um único estado no AFD minimizado
  - Matematicamente: $\forall\ w \in \Sigma^*, \delta(q, w) \in F \Leftrightarrow \delta(r, w) \in F$
    - **$\forall$** : "para todo"
    - **$w \in \Sigma^*$** : a string w pertence ao conjunto de todas as strings finitas (incluindo a string vazia) que podem ser formadas com os símbolos do alfabeto Σ ("sigma")
    - **δ** : função de transição ("delta")
      - **$\delta(q, w)$** : significa "o estado alcançado a partir do estado 'q' após ler a string 'w' "
    - **F** : conjunto de estados finais
    - **$\Leftrightarrow$** : se e somente se

AFD - minimização - exemplo 1

Algoritmo de preenchimento de tabela, ou
algoritmo de minimização de Moore



1º passo

- Iniciamos na coluna q0
- Analisamos os estados finais (F) e não-finais (N)
- q0 e q3 são iguais, deixa (q0, q3) em branco
- q0 e q2 são ≠, coloca um “X”
- Idem q0 e q1, e para as demais colunas

3º passo

- faz a mesma coisa, também é ≠
- como não tem mais nenhum → não dá para minimizar

q_1	X		
q_2	X		
q_3	X	X	X
	q_0	q_1	q_2

2º passo

δ(q0, 0) = q2
δ(q3, 0) = q2

são = , então não marca

δ(q0, 1) = q1
δ(q3, 1) = q3

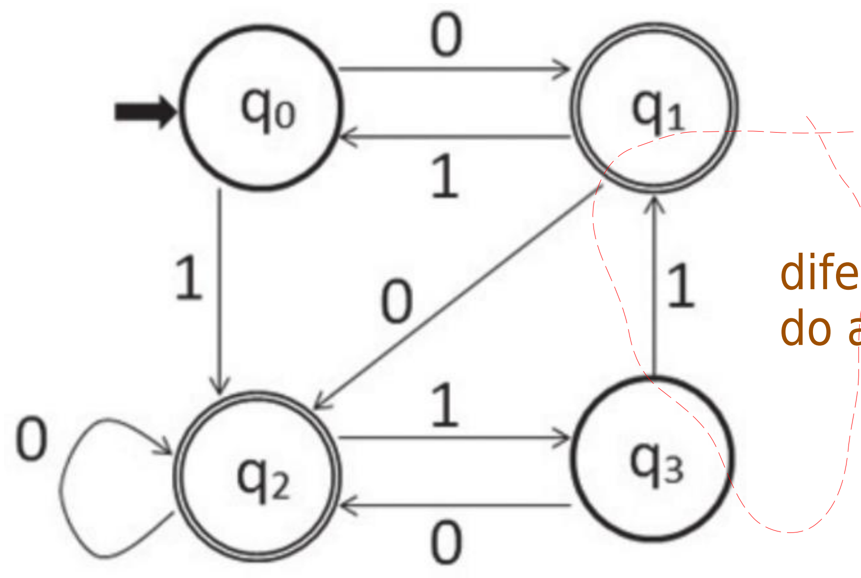
são ≠ , então:

como em (q1, q3) tem um X, então marcamos (q0, q3) com um X

AFD - minimização - exemplo 2

1º passo

- Iniciamos na coluna q0
- q0 é N
- q3 também é N, deixa (q0, q3) em branco
- q2 é F , então coloca “X”
- Repete para os demais elementos
- Os pares não marcados são potencialmente equivalentes



diferente
do anterior

q_1	X		
q_2	X		
q_3		X	X
	q_0	q_1	q_2

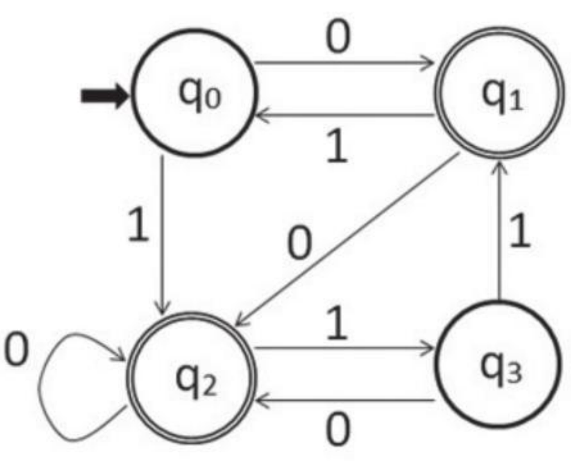
(continua)

AFD - minimização - exemplo 2 (continuação)

2º passo

- Começando com (q0, q3)
  - δ(q0, 0)=q1 e δ(q3, 0)=q2
    - Verificar se (q1, q2) são equivalentes
      - Na tabela não estão marcados e isso sugere que podem ser equivalentes
  - δ(q0, 1)=q2 e δ(q3, 1)=q1
    - Novamente, (q1, q2) não está marcado
- Como q1 e q2 não são distinguíveis na tabela: q0 e q3 são equivalentes (q0≡q3)
- Ainda dependemos de (q1, q2), que não foi analisado
- Colocamos um ponteiro "(q0, q3)" em (q1, q2)

	q_0	q_1	q_2
q_1	X		
q_2	X	(q0, q3)	
q_3		X	X



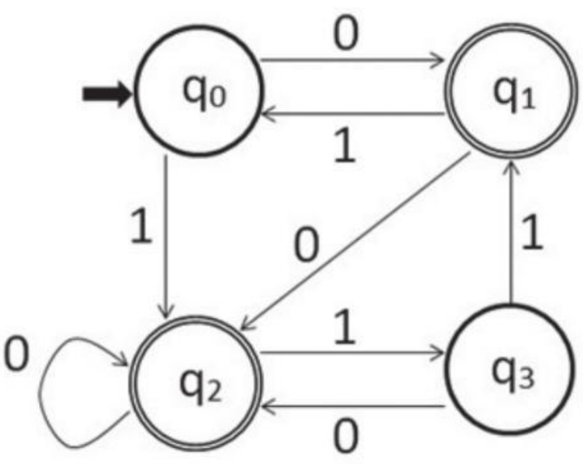
(continua)

AFD - minimização - exemplo 2 (continuação)

3º passo

- Verificamos a equivalência de $(q1, q2)$
- $\delta(q1, 0) = q2$ e $\delta(q2, 0) = q2$
  - q1 e q2 são equivalentes pois se comportam da mesma forma para todas as entradas
- $\delta(q1, 1) = q0$ e $\delta(q2, 1) = q3$
  - Verificar se q_0 e q_3 são equivalentes
    - Já determinamos que $q0 \equiv q3$
- Como todas as transições levam a estados equivalentes, q1 e q2 são equivalentes

q_1	X		
q_2	X	(q_0, q_3)	
q_3		X	x
	q_0	q_1	q_2

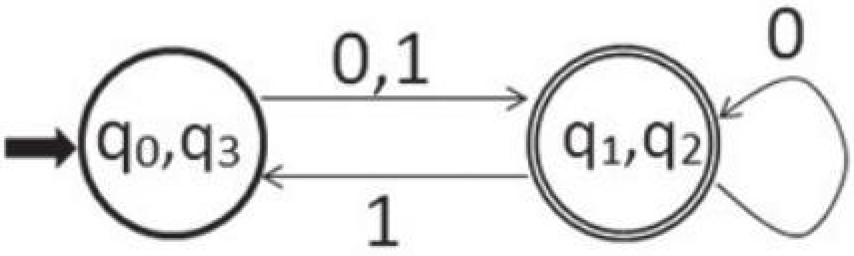


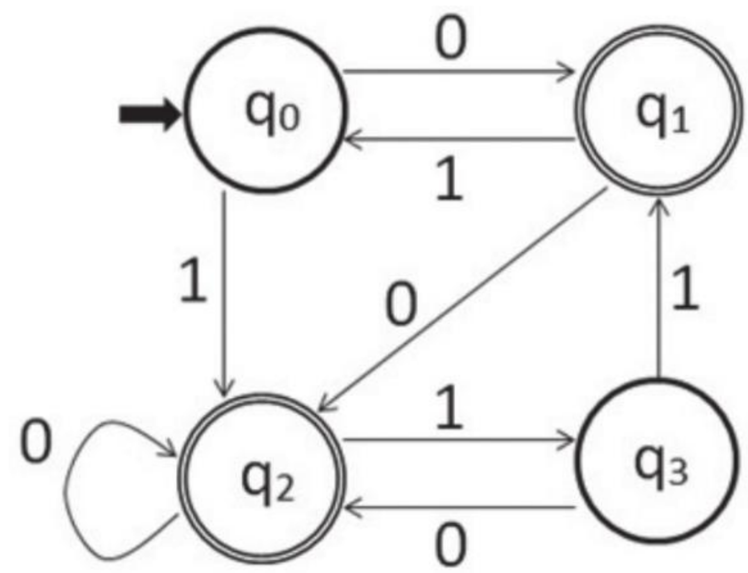
AFD - minimização - exemplo 2 (continuação)

4º passo

- Os estados equivalentes são:
  - {q0,q3} (q0 inicial, e q3 não final)
  - {q1,q2} (q1 e q2 finais)

Desenhamos o AFD minimizado:





E conferimos se todas as transições estão representadas