Autômato Finito

Stéfano S. B. V. Vita, Prof. Me.

Material adaptado: Prof. Yandre Maldonado - UEM/PR

Autômato Finito Determinístico

- AFD - modelo matemático p/ definição de linguagem
- Caráter reconhecedor
- Modelam também sistemas de estados finitos
- Exemplo clássico: problema HLCR (Hopcroft e Ullman)

Autômato Finito Determinístico

- Exemplo de sistema que pode ser representados desta forma:
  - Forno de micro-ondas
    - Entradas: porta aberta ou fechada, comandos fornecidos pelo cozinheiro através do painel, sinal do “timer” que expira.
    - Estados: aberto, esperando por comandos, cozinhando, desligado.

Definições Básicas

- Um AFD A define uma linguagem L(A) sobre um alfabeto Σ
- Caráter reconhecedor, ao contrário das gramáticas estudadas que tinham caráter gerador
  - dada uma cadeia *x*, ela pertence a L(A)?

Definições Básicas

- Uma abstração de um AFD
  - uma cabeça de leitura extrai sequencialmente o conteúdo de uma fita (string)
  - uma luz de aceitação que acende somente se a cadeia pertencer a linguagem representada pela AFD

Definições Básicas

- Definição Matemática de um AFD

Um AFD é uma quíntupla $<\Sigma, S, S_0, \delta, F>$, onde:

- Σ é o alfabeto de entrada
- S é um conjunto finito não vazio de estados
- S_0 é o estado inicial, $S_0 \in S$
- δ é a função de transição de estados, definida $\delta: S \times \Sigma \rightarrow S$
- F é o conjunto de estados finais, $F \subseteq S$

Definições Básicas

- Uma cadeia *x* para ser aceito, deve levar do estado S_0 para algum estado pertencente a F
- A função δ determina como são as transições de estados. Ela leva um par <*s*, *a*> onde *s* é um estado e *a* uma letra do alfabeto num estado *s'*
  - $\delta(s, a) = s'$

Definições Básicas

- Então, dado um AFD $A=<\Sigma,S,S_0,\delta,F>$ e a cadeia $x=a_1a_2...a_n \in \Sigma^*$, o autômato parte de S_0. Ao processar a_1 passa para o estado $\delta(S_0,a_1)$. Ao processar a_2 passa para $\delta(\delta(S_0,a_1),a_2)$, e assim por diante até processar a_n. Nesse ponto o autômato estará num estado R qualquer. Se $R \in F$ então a cadeia $x \in L(A)$.

Definições Básicas

- Finito: numero de estados envolvidos no sistema é finito
- Determinístico: estabelece que para uma cadeia $x \in L(A)$, só existe uma única seqüência de estados no AFD A para processá-la.

- Exemplo de Autômato:

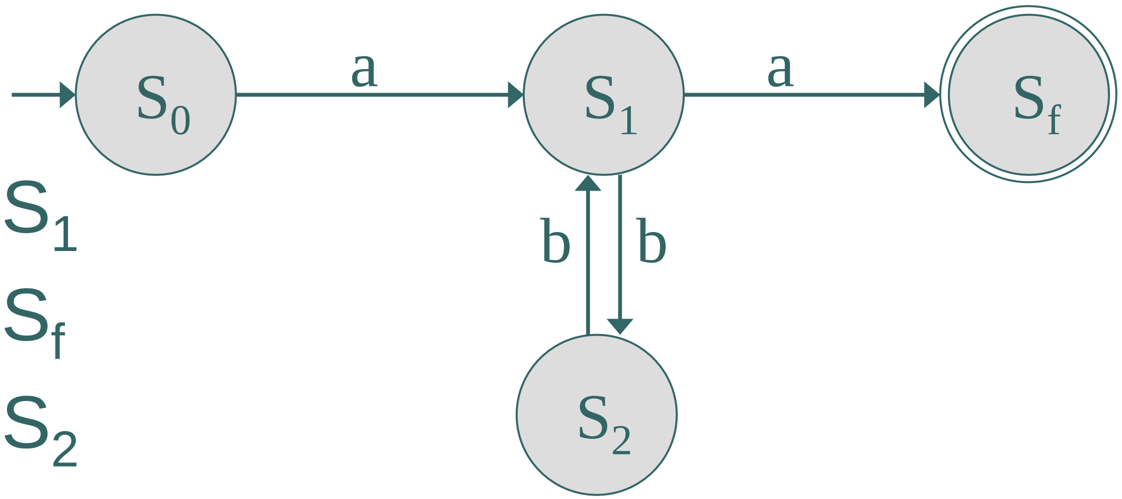
  $V = \langle \Sigma, S, S_0, \delta, F \rangle$ onde:

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

  $S = \{S_0, S_1, S_2, S_f\}$

  $S_0 = S_0$ - estado inicial

  $F = \{S_f\}$



  $\delta(S_0, a) = S_1$

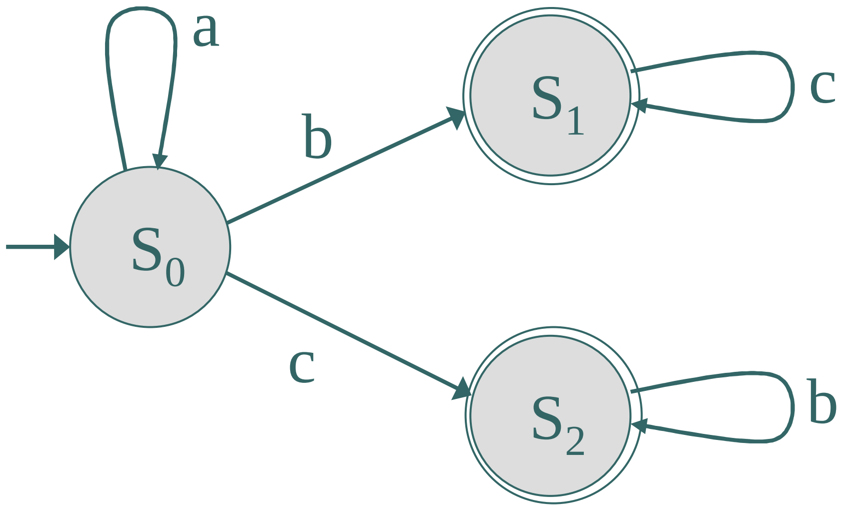
  $\delta(S_1, a) = S_f$

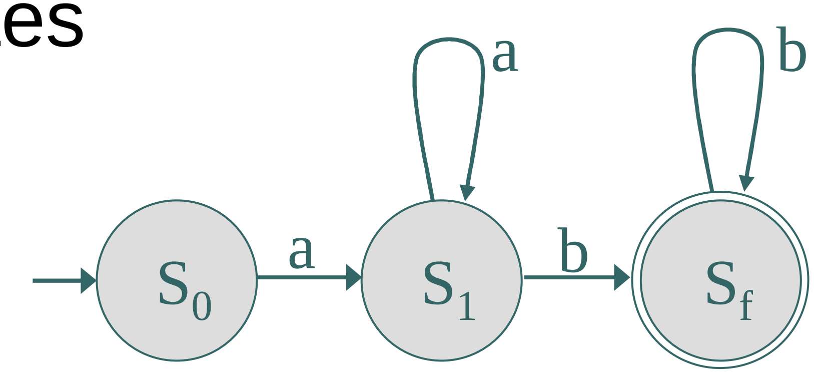
  $\delta(S_1, b) = S_2$

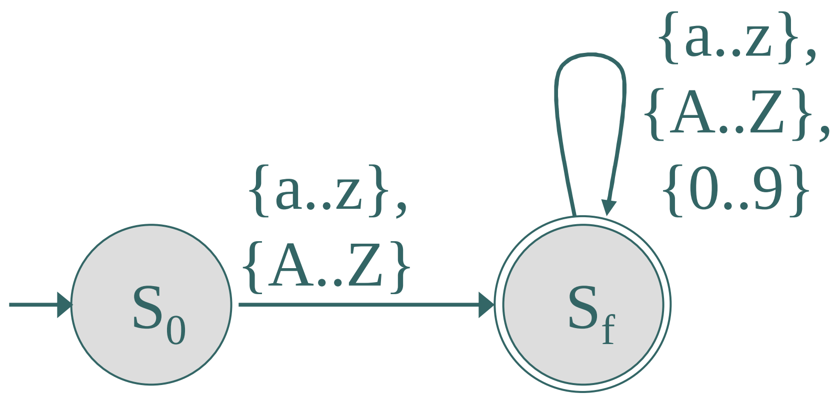
  $\delta(S_2, b) = S_1$

Exercícios

- Dados os seguintes autômatos:







  - Identifique a linguagem associada a cada um;
  - Descreva as funções de transição de cada um (exceto para o terceiro diagrama);

Definições Básicas

- Exemplo:
  - Modelagem de uma "*vending machine*" que aceita moedas de 5, 10 e 25 centavos. O preço do produto que ela entrega é 30 centavos.

Definições Básicas

- Partindo do estado inicial (0 centavos) deveremos reconhecer seqüências que nos levem a estados finais (valor inserido >= 30 centavos)
- Podemos chamar o autômato de V

- Assim:

  $V=<\Sigma, S, S_0, \delta, F>$ onde:

  Σ = {5, 10, 25} - cada um destes símbolos (ou letras) representa uma ação

  S = {<0c>, <5c>, <10c>, <15c>, <20c>, <25c>, <30c>, <35c>, <40c>, <45c>, <50c>} - cada estado indica quanto foi depositado

  S_0 = <0c> - estado inicial

  F = {<30c>, <35c>, <40c>, <45c>, <50c>} - estado onde a entrada é válida e o produto pode ser liberado

Delta é definida como:

$\delta(<0c>, 5) = <5c>$

$\delta(<0c>, 10) = <10c>$

$\delta(<0c>, 25) = <25c>$

$\delta(<5c>, 5) = <10c>$

$\delta(<5c>, 10) = <15c>$

$\delta(<5c>, 25) = <30c>$

$\delta(<10c>, 5) = <15c>$

$\delta(<10c>, 10) = <20c>$

$\delta(<10c>, 25) = <35c>$

...

Tabela de transição de estados

δ	5	10	25
<0c>	<5c>	<10c>	<25c>
<5c>	<10c>	<15c>	<30c>
<10c>	<15c>	<20c>	<35c>
<15c>	<20c>	<25c>	<40c>
<20c>	<25c>	<30c>	<45c>
<25c>	<30c>	<35c>	<50c>
<30c>	-	-	-
<35c>	-	-	-
<40c>	-	-	-
<45c>	-	-	-
<50c>	-	-	-

- Teste de cadeias:
  - 5 5 25
  - 5 5 10

- Diagrama de transições

Algoritmo do AFD

Início

Estado Atual $\leftarrow$ *Estado Inicial;*

Para I variar do Símbolo inicial da fita até o símbolo final

Faça *Se Existe* δ*(Estado Atual, I)*

Então Estado Atual $\leftarrow$ δ*(Estado Atual, I);*

Senão REJEITA;

Se Estado Atual é estado final

Então ACEITA;

Senão REJEITA;

Fim.

Autômato Finito Não Determinístico - AFND

- Modelo matemático semelhante ao AFD;
- Condições mais flexíveis;
- Podem haver múltiplos caminhos para processar uma cadeia;

Definição formal

- Definição Matemática de um AFND

Um AFND é uma quíntupla $<\Sigma, S, S_0, \delta, F>$, onde:

- Σ é o alfabeto de entrada
- S é um conjunto finito não vazio de estados
- S_0 é um conjunto não vazio de estados iniciais, $S_0 \subseteq S$
- δ é a função de transição de estados, definida δ: $S \times \Sigma \rightarrow \rho(S)$
- F é o conjunto de estados finais, $F \subseteq S$

AFND

- *Σ, S e F* são os mesmos do AFD;
- S_0 era um único estado em AFD. Em AFND é um conjunto com pelo menos um estado inicial;
- Então em AFD $S_0 \in S$, e em AFND $S_0 \subseteq S$;
- Um AFND pode ter mais de um estado “ativo” (corrente) num instante;
- Inicialmente, todo estado de S_0 são ativados;

AFND

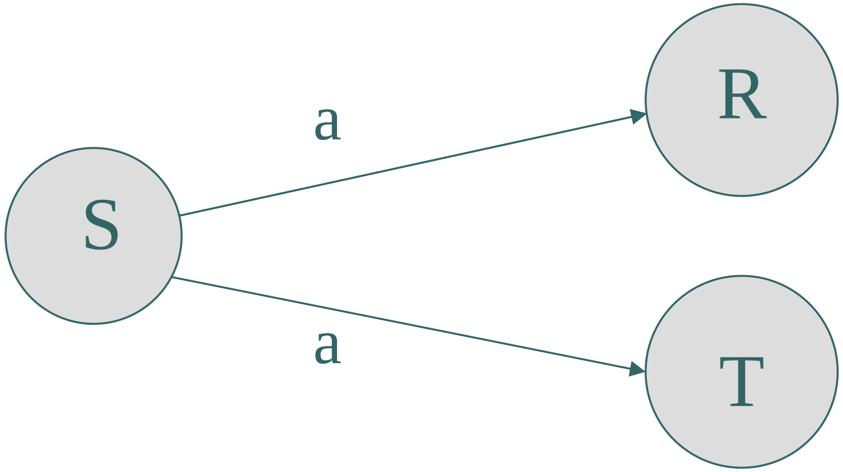
- “Alternativas” para processar uma única cadeia;
- Se o processamento de uma cadeia não levar ao estado final por um caminho, deve-se tentar por outros caminhos (se houverem);
- Se algum dos caminhos possíveis levar a cadeia *x* a um estado final, então ela faz parte da linguagem definida pelo autômato.

AFND

- A cadeia é rejeitada se a partir de nenhum estado inicial for possível atingir um estado final ao término do processamento;
- A função δ agora é definida $S \times \Sigma \rightarrow \rho(S)$, onde cada elemento do contra-domínio ($\rho(S)$) é um conjunto de estados pertencentes a S;

AFND

- Exemplo da função: $\delta(S,a) = \{R, T\}$



- Se o estado S estiver ativo e a letra *a* for processada, então tanto R quanto T passaram a estar ativos;

AFND

- Se seguirmos por R e não chegarmos a um estado final, podemos tentar por T;
- Se alguma das alternativas levar a um estado final, a cadeia é reconhecida;
- Se nenhuma alternativa levar a um estado final, a cadeia é rejeitada;

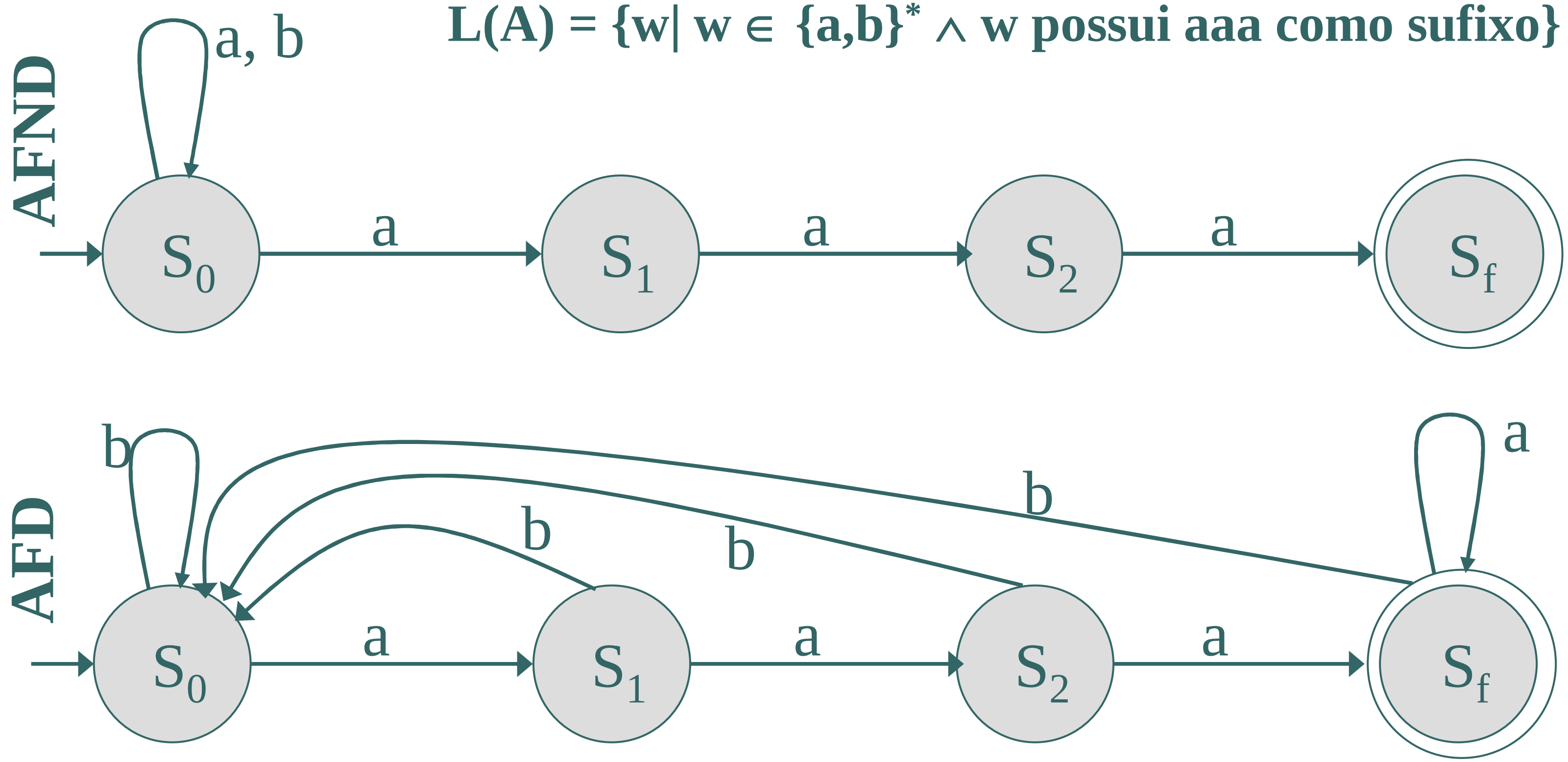
AFND

- Ao processar uma letra, todas as transições rotuladas com aquela letra, a partir de todos os estados ativos serão efetuadas;
- Então, podemos novamente observar que podemos ter vários estados ativos ao mesmo tempo, ao contrário dos AFD's.

Equivalência entre AFD e AFND

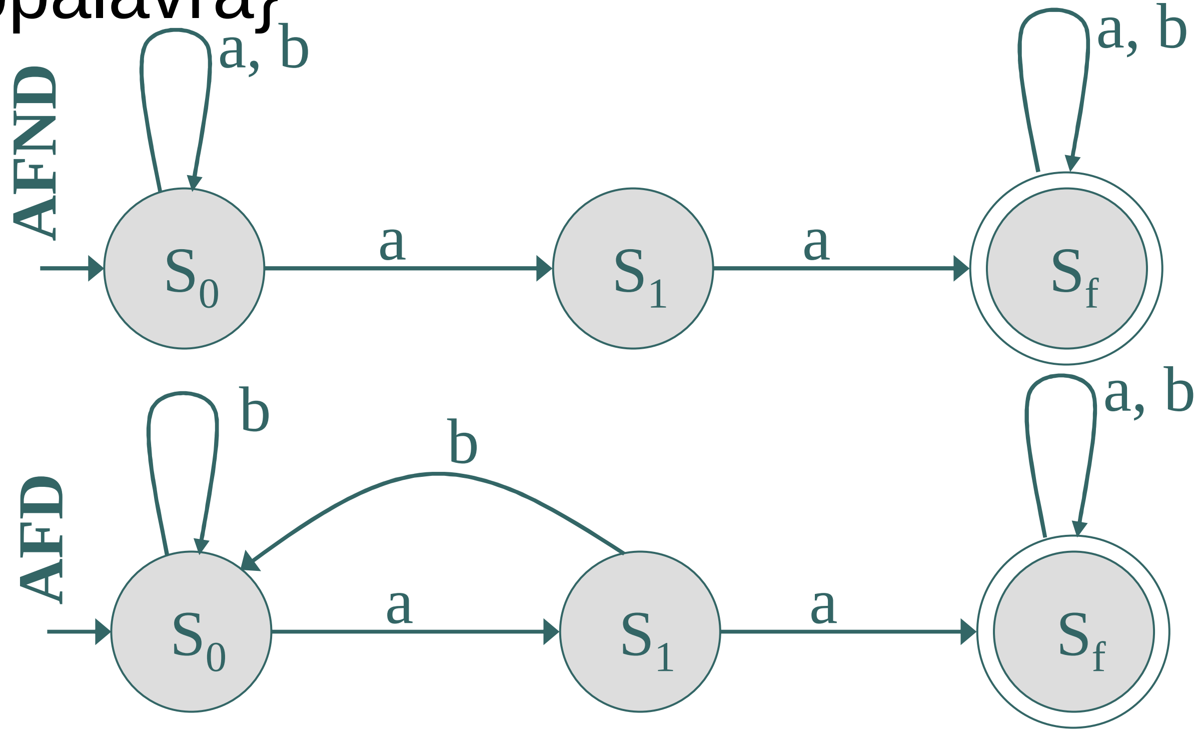
- A classe dos AFD's é equivalente à classe dos AFND's. Assim, para todo AFD existe um AFND equivalente e vice-versa.
- Exemplo:

$L(A) = \{w \mid w \in \{a,b\}^* \wedge w$ possui aaa como sufixo$\}$



Exercício

- Construa um AFD e um AFND para a seguinte linguagem:
  - L(A) = {w|w ∈ {a,b}* ∧ w possui aa como subpalavra}



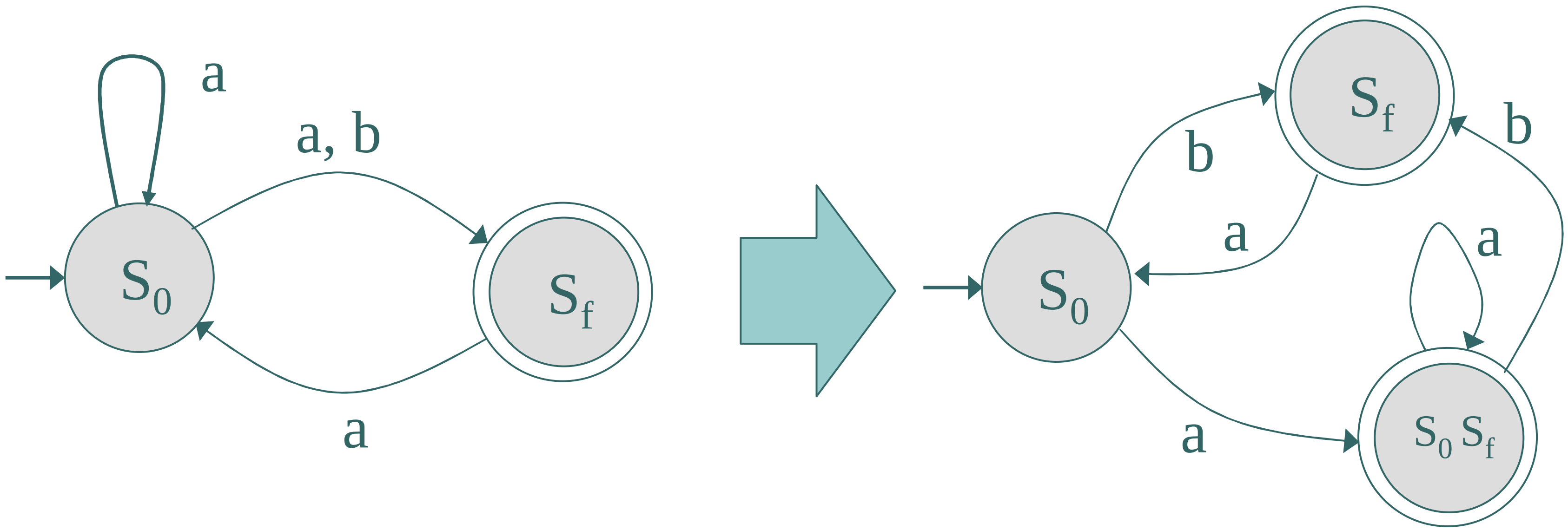
Transformação de AFND em AFD

- Para todo AFND existe um AFD equivalente;
- Dado um AFND $A=<\Sigma, S, S_0, \delta, F>$, define-se o AFD $A^d=<\Sigma, S^d, S^d_0, \delta^d, F^d>$ equivalente da seguinte forma:
  - $S^d = \rho(S)$
  - $S^d_0 = S_0$
  - F^d : todas as combinações de estados de S^d que possui como componente algum estado de F;
  - Seja $Q \in S^d$, $\delta^d(Q,a)$ vai levar à um estado que corresponde à combinação de todos os estados que podem ser alcançados ao processar o símbolo ‘a’ a partir de qualquer componente de Q.

Conjunto de todas as combinações de estados de S

Transformação de AFND em AFD

- Exemplo: Criação das transições
  - $\delta^d(Q,a)$ será igual ao estado que corresponda à combinação de todos os estados que formam o conjunto $\delta(Q,a)$;

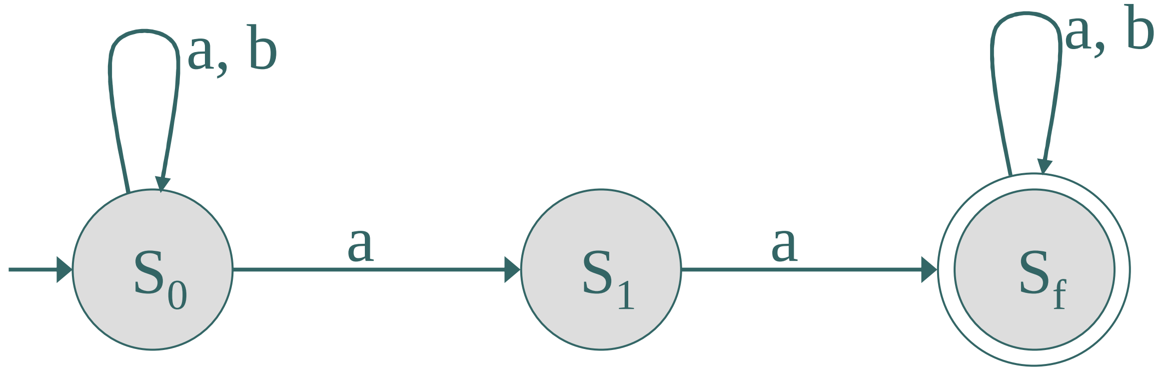


Transformação de AFND em AFD

- Exercício:
  - É possível prever quantos estados serão formados em um AFD obtido a partir de um AFND?
    - 2^Q para um AFD com função de transição total;
    - 2^Q-1 para um AFD com função de transição parcial;
      - Onde Q é o número de estados do AFND.
  - Mostre como ficaria o AFD do exemplo 1 com função de transição total.

Transformação de AFND em AFD

- Durante a transformação de um AFND em um AFD podem ser criados estados que fiquem “isolados” do(s) estado(s) inicial(is);
- Estes podem simplesmente ser eliminados;
- Exercício: transforme o AFND descrito a seguir em AFD.



Minimização de AFD

- Dois AFD's A e B são equivalentes, denotado por $A \equiv B$, se $L(A)=L(B)$;
- Um autômato mínimo para uma linguagem regular L é um autômato com o menor número de estados possível que aceita L;

Um AFD $A=<\Sigma, S_A, S_{0A}, \delta_A, F_A>$ é dito mínimo se para qualquer AFD $B=<\Sigma, S_B, S_{0B}, \delta_B, F_B>$ tal que $A \equiv B$, temos que $|S_A| \leq |S_B|$.

Minimização de AFD

- Pré-requisitos para a minimização:
  - O Autômato deve ser determinístico;
  - Não pode ter estados inacessíveis a partir do estado inicial;
  - A função de transição deve ser total (todas as saídas devem ser previstas para todos os estados).
    - Para este último requisito, muitas vezes é necessário introduzir um estado S_{trash} para lançar as transições não previstas originalmente.

Minimização de AFD

- A estratégia consiste fundir estados equivalentes* num mesmo estado;
- Para isto, são identificados estados não-equivalentes e, por exclusão, encontra-se os equivalentes;
- Utiliza-se uma tabela triangular que possui um cruzamento para cada par de estados distintos do autômato (incluindo S_{trash} quando este é acrescentado).

***** Dois estados são equivalentes se, ao processarem uma mesma cadeia, ambos chegam a estados finais, ou ambos chegam a estados não finais.

Minimização de AFD

- Supondo um autômato com $S = \{S_0, S_1, ..., S_{n-1}, S_n, S_{trash}\}$

S_1					
S_2					
...					
S_n					
S_{trash}					
	S_0	S_1	...	S_{n-1}	S_n

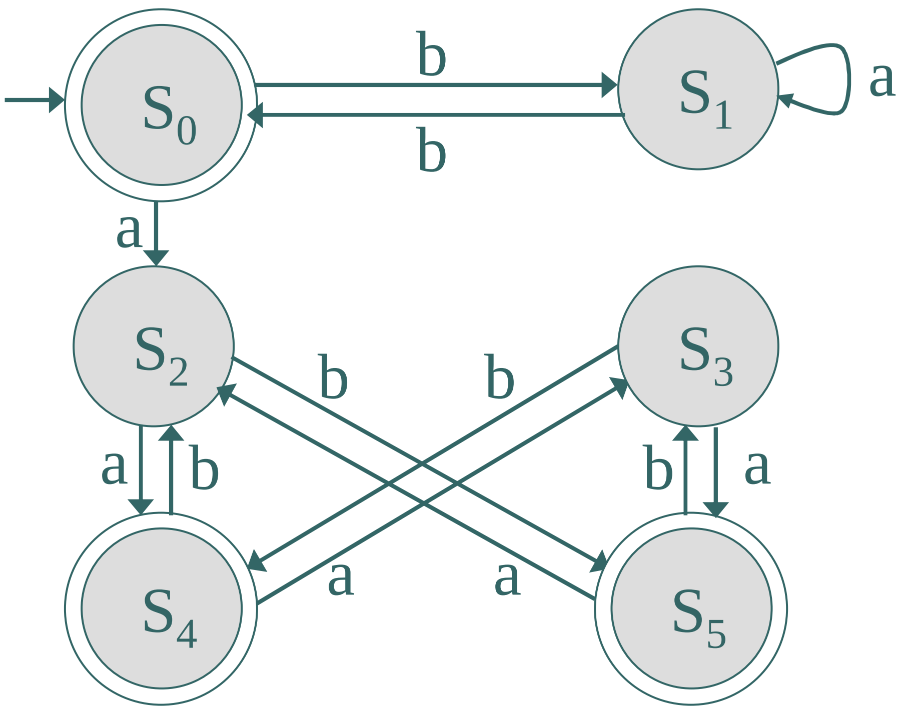
Estados não equivalentes serão sinalizados, ao final do processo, os cruzamentos em branco deverão indicar estados equivalentes.

Minimização de AFD

- Inicia-se a marcação pelos estados trivialmente não-equivalentes;
  - Dois estados são trivialmente não-equivalentes se um é final e o outro não;

Minimização de AFD

- Exemplo
  - Considere o seguinte AFD e sua tabela de cruzamentos de estados:

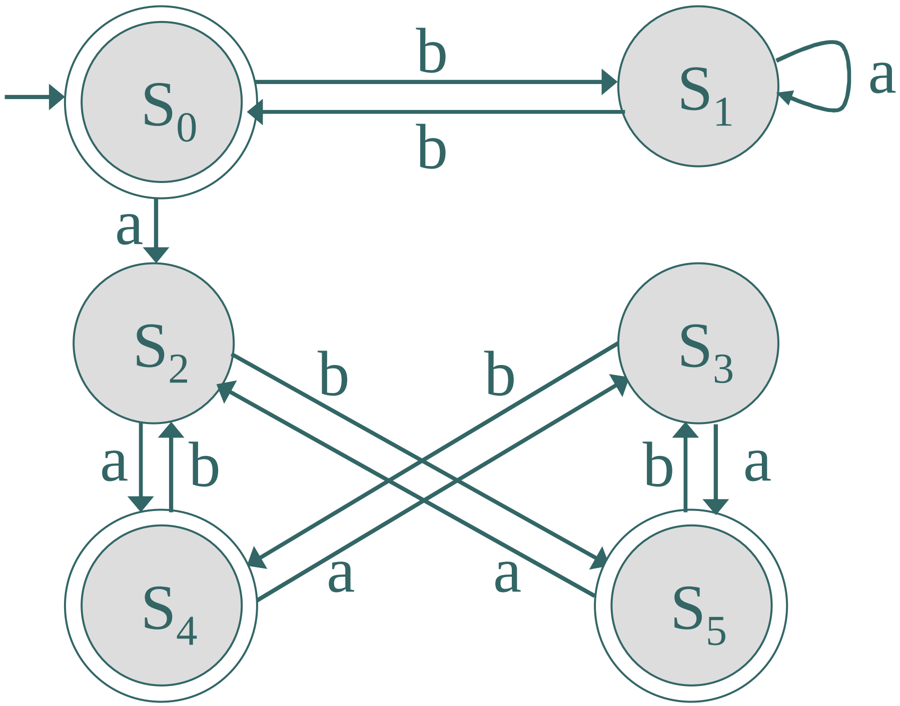


Estados trivialmente não-equivalentes

S_1	X				
S_2	X				
S_3	X				
S_4		X	X	X	
S_5		X	X	X	
	S_0	S_1	S_2	S_3	S_4

Minimização de AFD

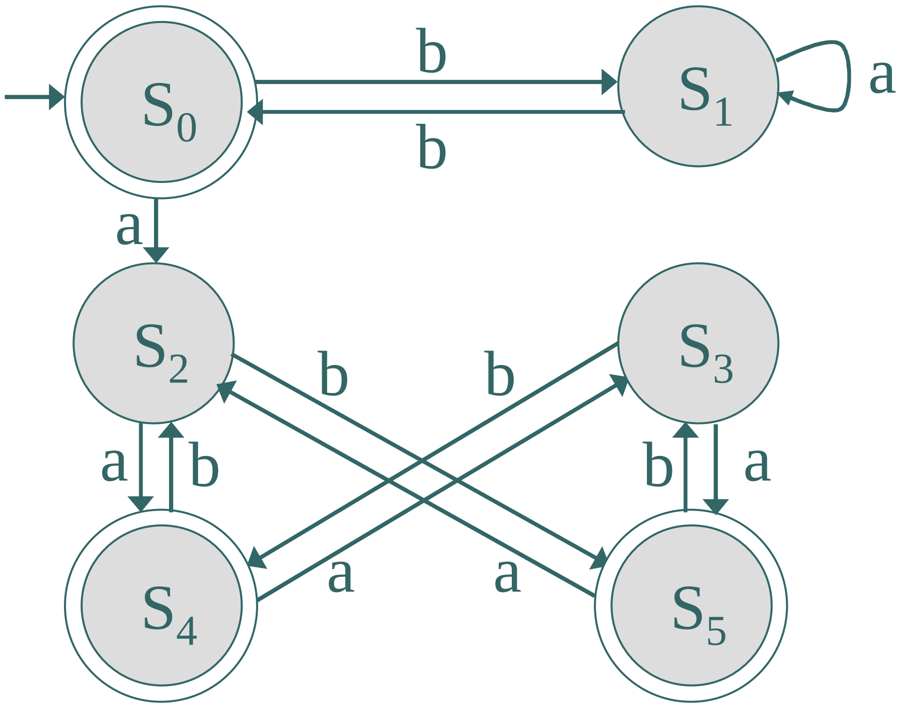
- Análise dos pares ainda não marcados:
  - Verificar se cada par de estados não marcado leva, ao processar um mesmo símbolo, à estados não-equivalentes;
  - Nesta etapa os pares serão marcados com o símbolo ⊗.



S_1	X				
S_2	X				
S_3	X				
S_4		X	X	X	
S_5		X	X	X	
	S_0	S_1	S_2	S_3	S_4

Minimização de AFD

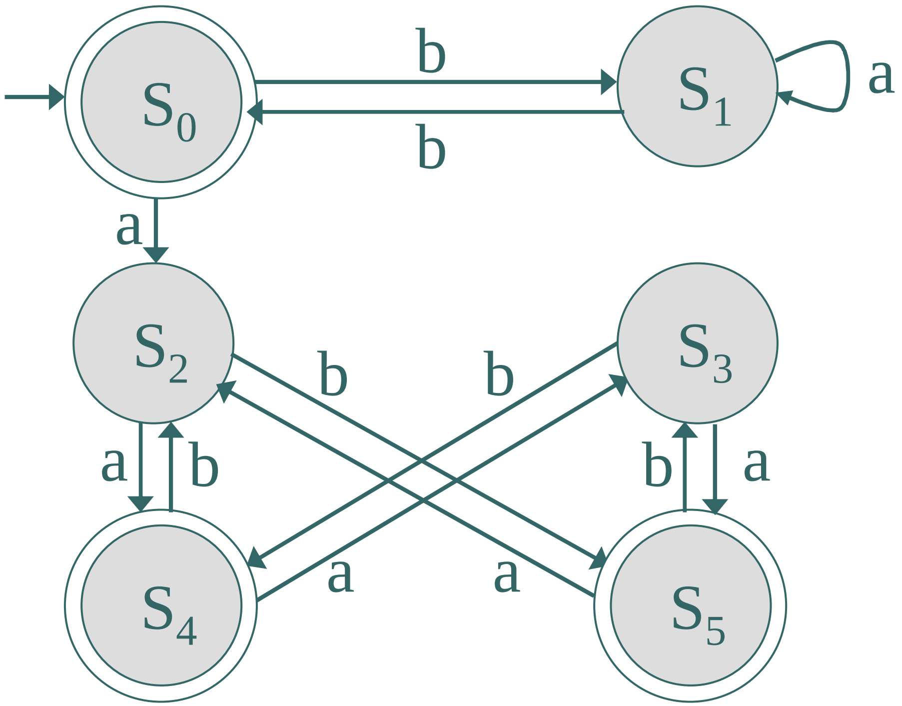
- Análise dos pares ainda não marcados:
  - $\{S_0, S_4\}$?
    - $\delta(S_0, a)=S_2$ $\delta(S_4, a)=S_3$ $\Rightarrow$ $\{S_2, S_3\}$?
    - $\delta(S_0, b)=S_1$ $\delta(S_4, b)=S_2$ $\Rightarrow$ $\{S_1, S_2\}$?



S_1	X				
S_2	X	$\rightarrow \{S_0, S_4\}$			
S_3	X		$\rightarrow \{S_0, S_4\}$		
S_4		X	X	X	
S_5		X	X	X	
	S_0	S_1	S_2	S_3	S_4

Minimização de AFD

- Análise dos pares ainda não marcados:
  - $\{S_0, S_5\}$?
    - $\delta(S_0, a)=S_2$ $\quad$ $\delta(S_5, a)=S_2$
    - $\delta(S_0, b)=S_1$ $\quad$ $\delta(S_5, b)=S_3$ $\quad \Rightarrow \quad$ $\{S_1, S_3\}$?



S_1	X				
S_2	X				
S_3	X				
S_4		X	X	X	
S_5		X	X	X	
	S_0	S_1	S_2	S_3	S_4

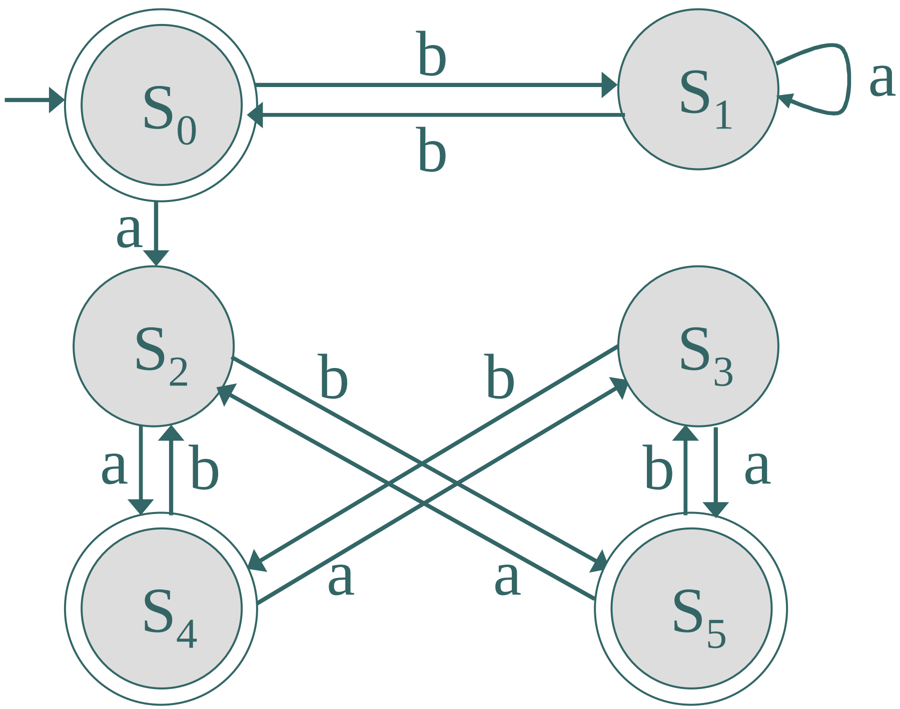
$\rightarrow \{S_0, S_5\}$

$\rightarrow \{S_0, S_4\}$

$\rightarrow \{S_0, S_4\}$

Minimização de AFD

- Análise dos pares ainda não marcados:
  - $\{S_1, S_2\}$?
    - $\delta(S_1, a)=S_1$ $\delta(S_2, a)=S_4$ $\Rightarrow$ $\{S_1, S_4\}$ = X
    - $\delta(S_1, b)=S_0$ $\delta(S_2, b)=S_5$ $\Rightarrow$ $\{S_0, S_5\}$?



S_1	X				
S_2	X				
S_3	X				
S_4		X	X	X	
S_5		X	X	X	
	S_0	S_1	S_2	S_3	S_4

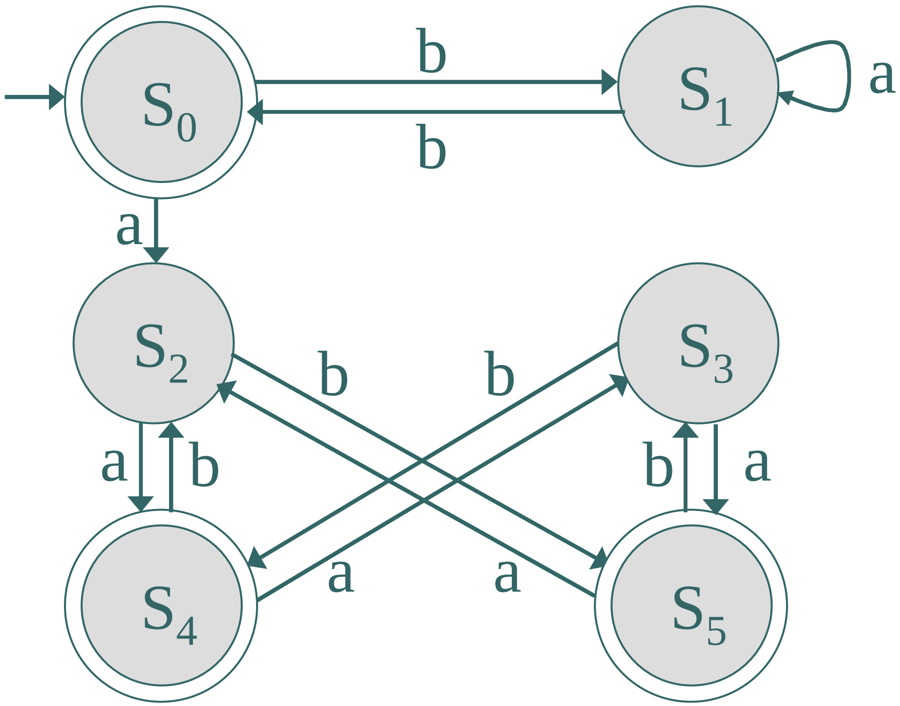
$\{S_0, S_5\}$

$\{S_0, S_4\}$

$\{S_0, S_4\}$

Minimização de AFD

- Análise dos pares ainda não marcados:
  - $\{S_1, S_3\}$?
    - $\delta(S_1, a)=S_1 \quad \delta(S_3, a)=S_5 \quad \Rightarrow \quad \{S_1, S_5\} = X$
    - $\delta(S_1, b)=S_0 \quad \delta(S_3, b)=S_4 \quad \Rightarrow \quad \{S_0, S_4\} = \otimes$

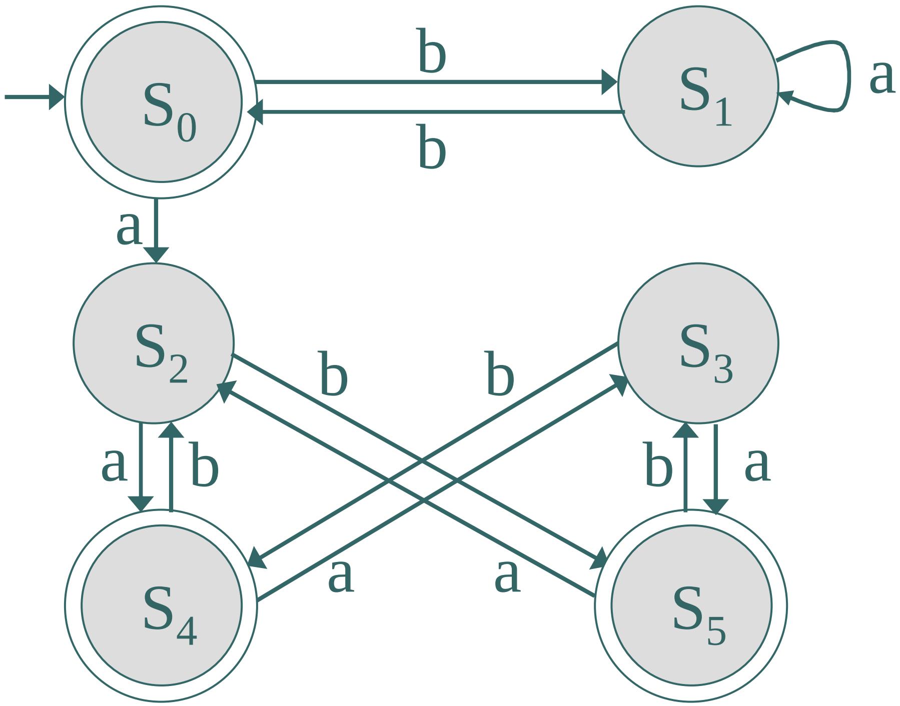


$\{S_0, S_5\}$

S_1	X				
S_2	X	$\otimes$			
S_3	X				
S_4	$\otimes$	X	X	X	
S_5		X	X	X	
	S_0	S_1	S_2	S_3	S_4

Minimização de AFD

- Análise dos pares ainda não marcados:
  - $\{S_2, S_3\}$?
    - $\delta(S_2, a)=S_4$ $\quad$ $\delta(S_3, a)=S_5$ $\quad \Rightarrow \quad$ $\{S_4, S_5\}$?
    - $\delta(S_2, b)=S_5$ $\quad$ $\delta(S_3, b)=S_4$ $\quad \Rightarrow \quad$ $\{S_4, S_5\}$?

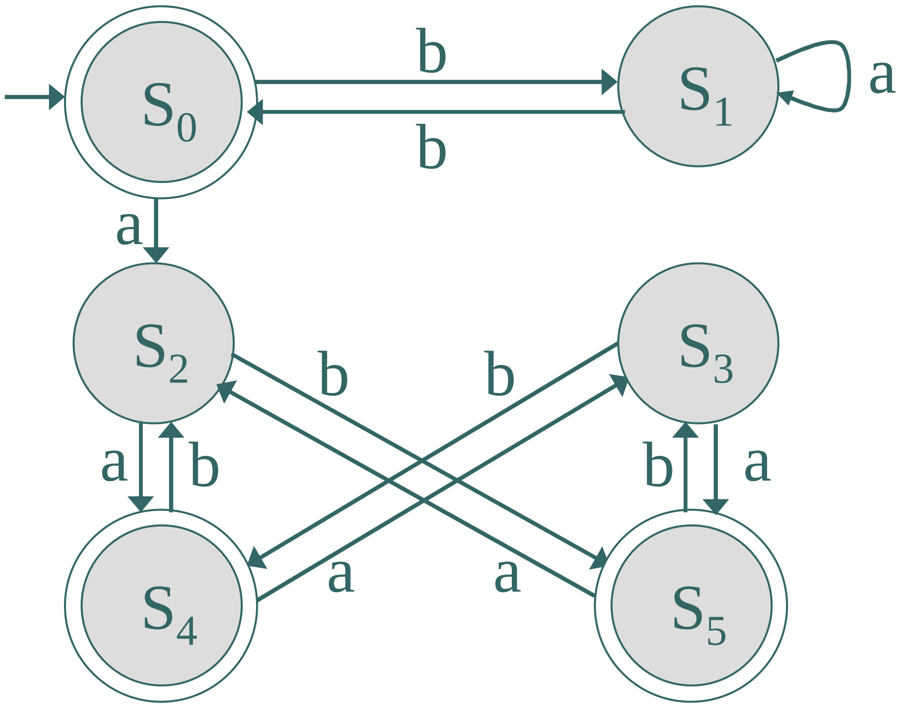


S_1	X				
S_2	X	⊗			
S_3	X	⊗			
S_4	⊗	X	X	X	
S_5	⊗	X	X	X	
	S_0	S_1	S_2	S_3	S_4

$\{S_2, S_3\}$

Minimização de AFD

- Análise dos pares ainda não marcados:
  - $\{S_4, S_5\}$?
    - $\delta(S_4, a)=S_3$ $\delta(S_5, a)=S_2$ $\Rightarrow$ $\{S_2, S_3\}$?
    - $\delta(S_4, b)=S_2$ $\delta(S_5, b)=S_3$ $\Rightarrow$ $\{S_2, S_3\}$?



S_1	X				
S_2	X	⊗			
S_3	X	⊗			
S_4	⊗	X	X	X	
S_5	⊗	X	X	X	
	S_0	S_1	S_2	S_3	S_4

$\{S_4, S_5\}$

$\{S_2, S_3\}$

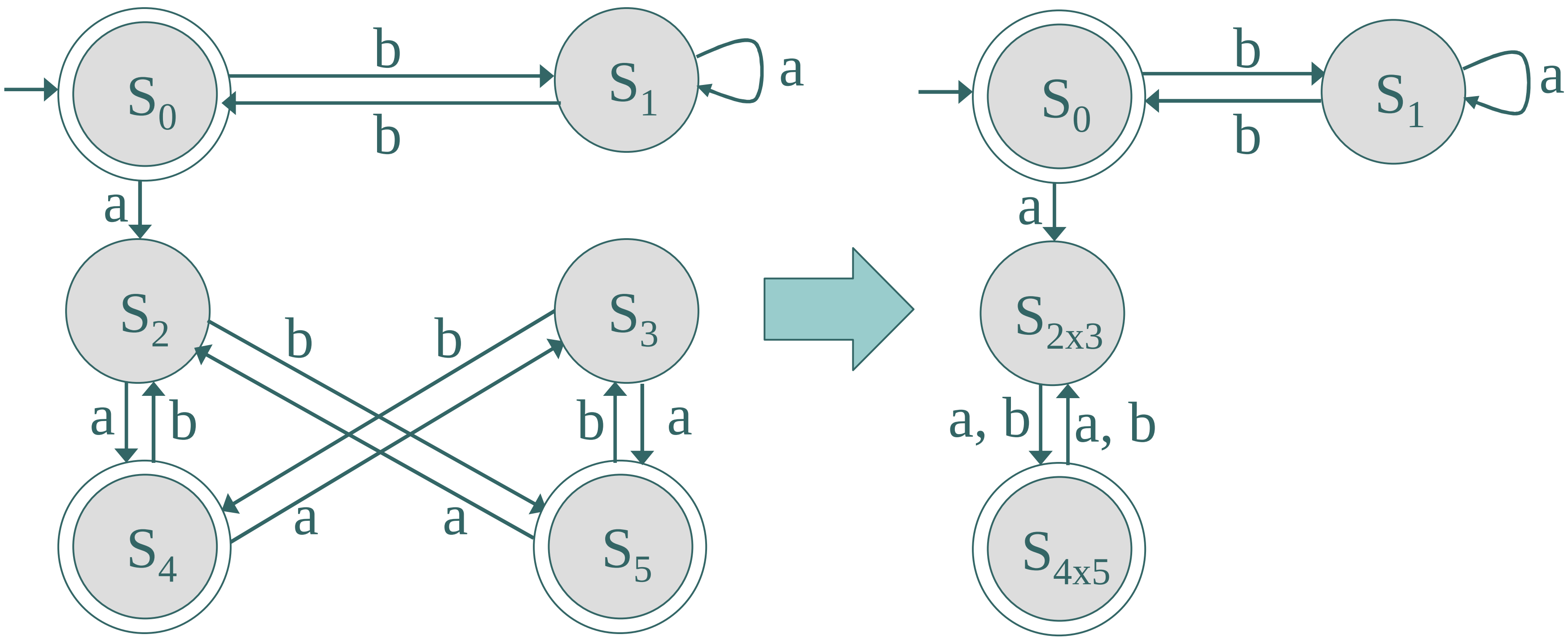
Minimização de AFD

- Conclusão
  - Não foi identificada a não-equivalência entre os pares: S_2xS_3 e S_4xS_5;
  - Assim, estes pares podem ser “fundidos” em um único estado;

S_1	X				
S_2	X	⊗			
S_3	X	⊗			
S_4	⊗	X	X	X	
S_5	⊗	X	X	X	
	S_0	S_1	S_2	S_3	S_4

Minimização de AFD

- Fusão dos pares S_2xS_3 e $S_4xS_{5:}$



Minimização de AFD

- Critério usado para a marcação dos estados não-equivalentes:
  - Para cada par $\{S_u, S_v\}$ não marcado e para cada símbolo $a \in \Sigma$, suponha que $\delta(S_u, a)=R_u$ e $\delta(S_v, a)=R_v$:
    - Se $R_u = R_v$, então S_u e S_v são equivalentes e, para o símbolo a não deve ser marcado;
    - Se $R_u \neq R_v$ e o par $\{R_u, R_v\}$ não está marcado, então $\{S_u, S_v\}$ é incluído em uma lista a partir de $\{R_u, R_v\}$ para posterior análise;
    - Se $R_u \neq R_v$ e o par $\{R_u, R_v\}$ está marcado, então:
      - $\{S_u, S_v\}$ é não equivalente e deve ser marcado;
      - Se $\{S_u, S_v\}$ encabeça uma lista de pares, então todos os pares da lista devem ser marcados.

Minimização de AFD

- Na unificação de estados equivalentes:
  - Conjuntos de estados finais equivalentes podem ser unificados como um único estado final;
  - Conjuntos de estados não-finais equivalentes podem ser unificados como um único estado não-final;
  - Se algum dos estados equivalentes é inicial, então o correspondente estado unificado é inicial.

Bibliografia

- MENEZES, Paulo Blauth. Linguagens Formais e Autômatos. Porto Alegre: Editora Sagra-Luzzatto, 1998;
- DELAMARO, Márcio Eduardo. Linguagens Formais e Autômatos. UEM, 1998;
- HOPCROFT, J. E., ULLMAN, J. D. e MOTWANI, R. Introdução à Teoria de Autômatos, Linguagens e Computação. Rio de Janeiro: Editora Campus, 2003.