TEORIA DA COMPUTAÇÃO

AULA 4: AUTÔMATOS E LINGUAGENS (CAP 1)

NÃO-DETERMINISMO

PROFESSOR: LUCAS CAMBUIM

Determinismo × Não-Determinismo

- Os AFs que vimos até agora funcionam assim: quando a máquina está em um dado estado e lê o próximo símbolo de entrada, **sabemos qual será o próximo estado**, está determinado. Chamamos isso de computação *determinística*.

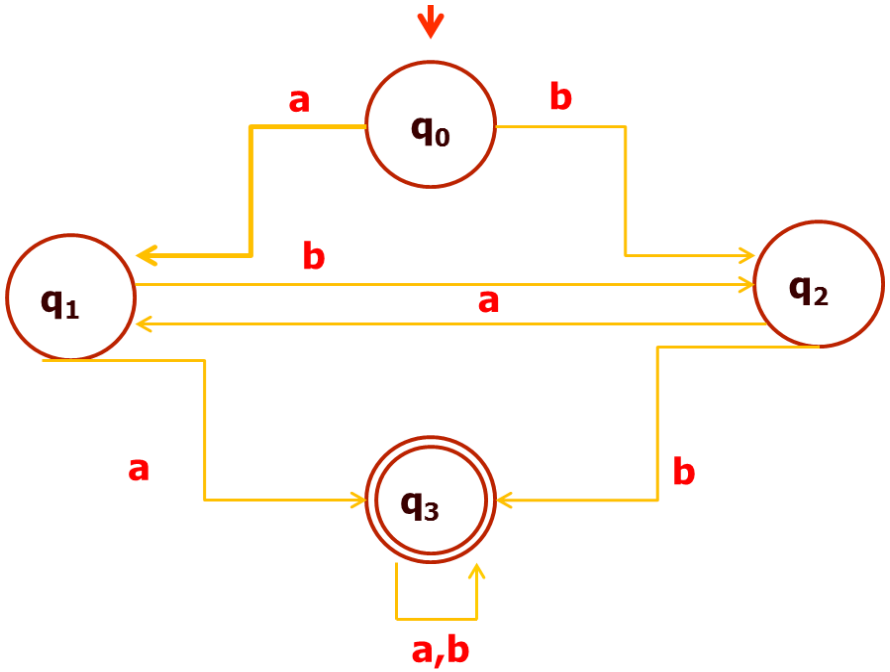
- Em um autômato finito *não-determinístico (AFN)*, **várias escolhas podem existir** para o próximo estado em qualquer ponto.

Determinismo × Não-Determinismo

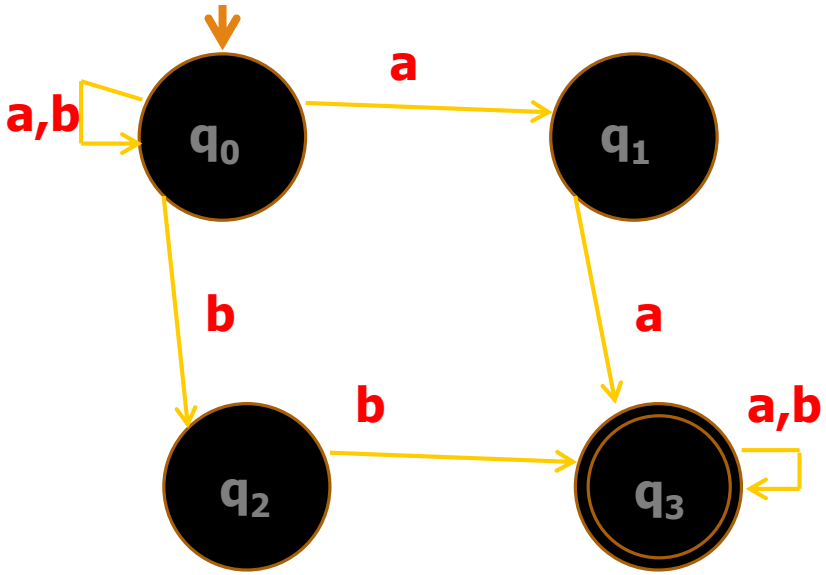
- Algumas diferenças práticas
  - Em um AFN um estado pode ter zero, uma ou várias setas saindo para cada símbolo do alfabeto.

Autômatos finitos não-determinísticos

Considere um AFN N_1 que aceita as cadeias sobre o alfabeto {a,b} que possuem aa ou bb como subcadeia.



Todo estado de um AFD sempre tem exatamente uma seta de transição saindo para cada símbolo no alfabeto



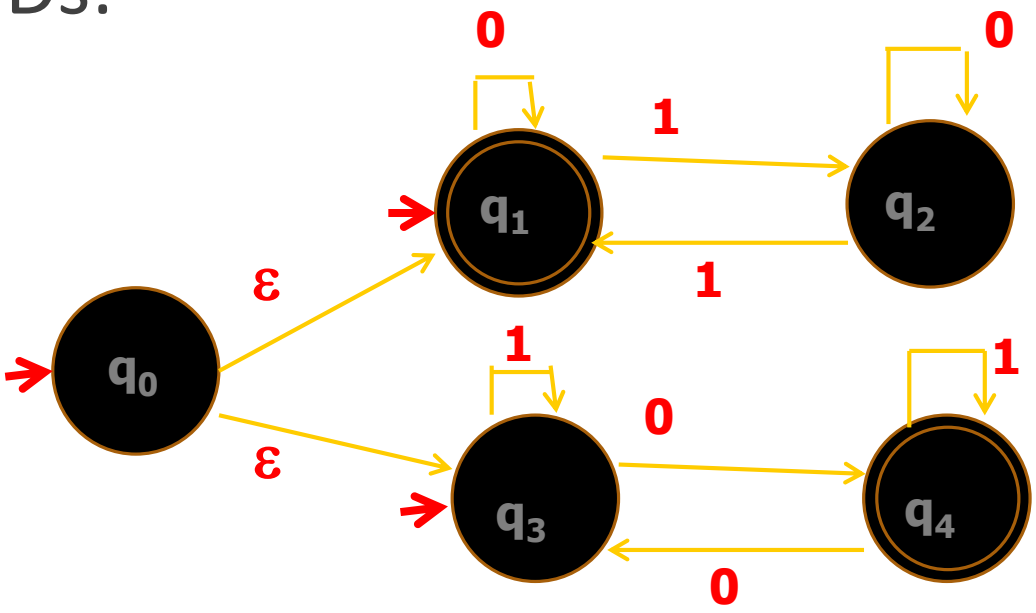
Em um AFN um estado pode ter zero, uma ou muitas setas saindo para cada símbolo do alfabeto.

Determinismo × Não-Determinismo

- Algumas diferenças práticas
  - Em geral, um AFN pode ter setas rotuladas com membros do alfabeto ou com ϵ
    - Zero uma ou mais setas podem sair de cada estado com o rótulo ϵ

Autômatos finitos não-determinísticos

Vamos construir um AFN que aceita cadeias com um número par de 1s ou ímpar de 0s a partir desses AFDs:

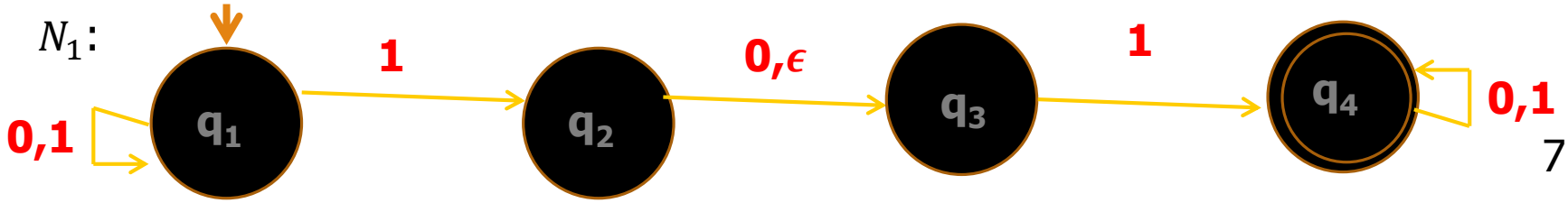


Em geral, um AFN pode ter
ter setas rotuladas com membros do alfabeto
ou com ε.

Autômatos finitos não-determinísticos

Como devemos pensar no processamento de um AFN?

- No estado q_1: Se receber '1', N_1 se divide em múltiplas cópias de si mesma e segue *todas* as possibilidades em **paralelo**
- Se existirem escolhas subsequentes, a máquina divide-se novamente
- Se o próximo símbolo de entrada não aparece em qualquer das setas, a cópia da máquina morre.
- Um estado com uma seta **ϵ:** sem ler qualquer entrada, a máquina divide-se em múltiplas cópias, uma seguindo cada uma das setas que tem **ϵ** e uma permanecendo no estado corrente.
- Finalmente, se *qualquer uma* dessas cópias da máquina está em um estado de aceitação no final da entrada, o AFN aceita a cadeia de entrada



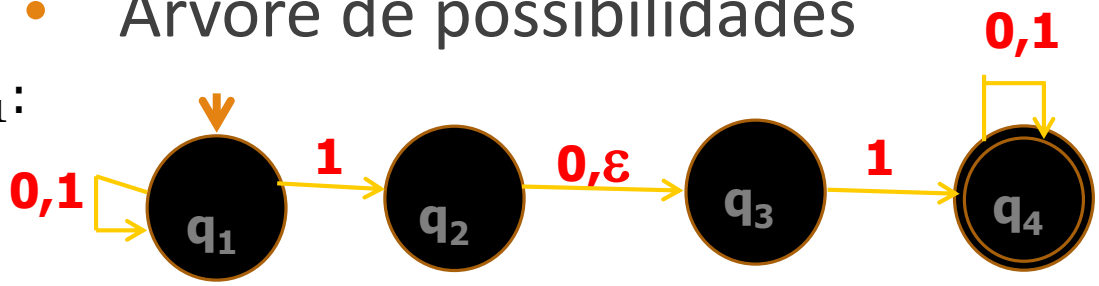
A computação em um AFN

- Não-determinismo pode ser visto como uma espécie de **computação paralela** na qual múltiplos e independentes “processos” ou “threads” podem estar rodando concorrentemente.
- Se pelo menos um desses processos aceita então a computação inteira aceita

A computação em um AFN

- Outra forma de pensar:
  - Árvore de possibilidades

N_1:

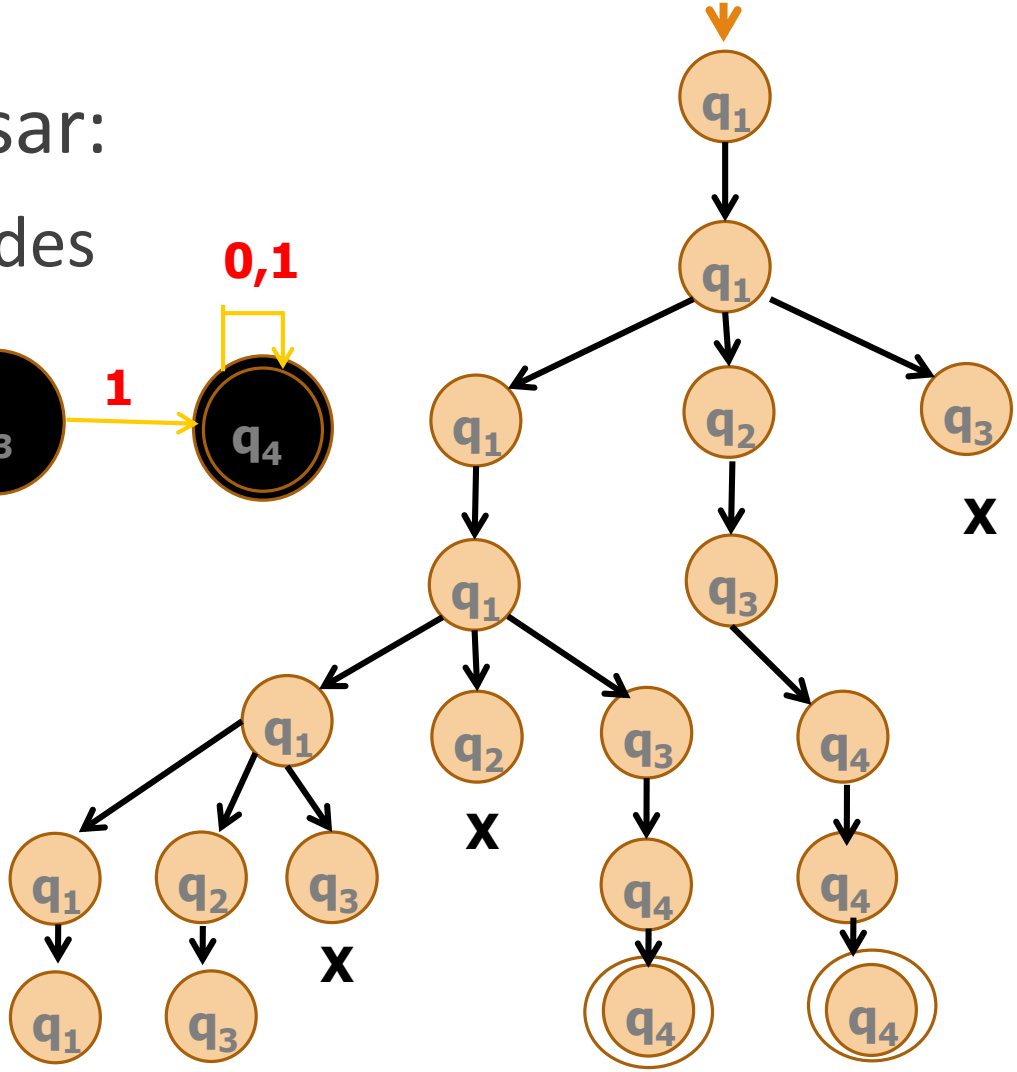


Entrada **010110**

Leitura: **0 1 0 1 1 0**

Qual linguagem N_1 reconhece?

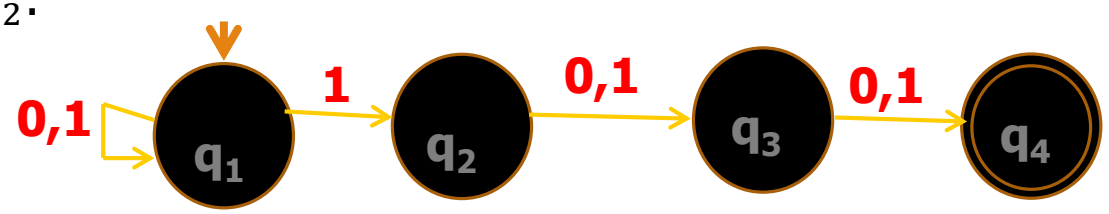
N_1 aceita todas as cadeias que contêm 101 ou 11 como uma subcadeia



Exemplo 1.30

Seja A a linguagem consistindo de todas as cadeias sobre {0,1} contendo um 1 na terceira posição a partir do final (por exemplo, 000100 está em A, mas 0011 não).

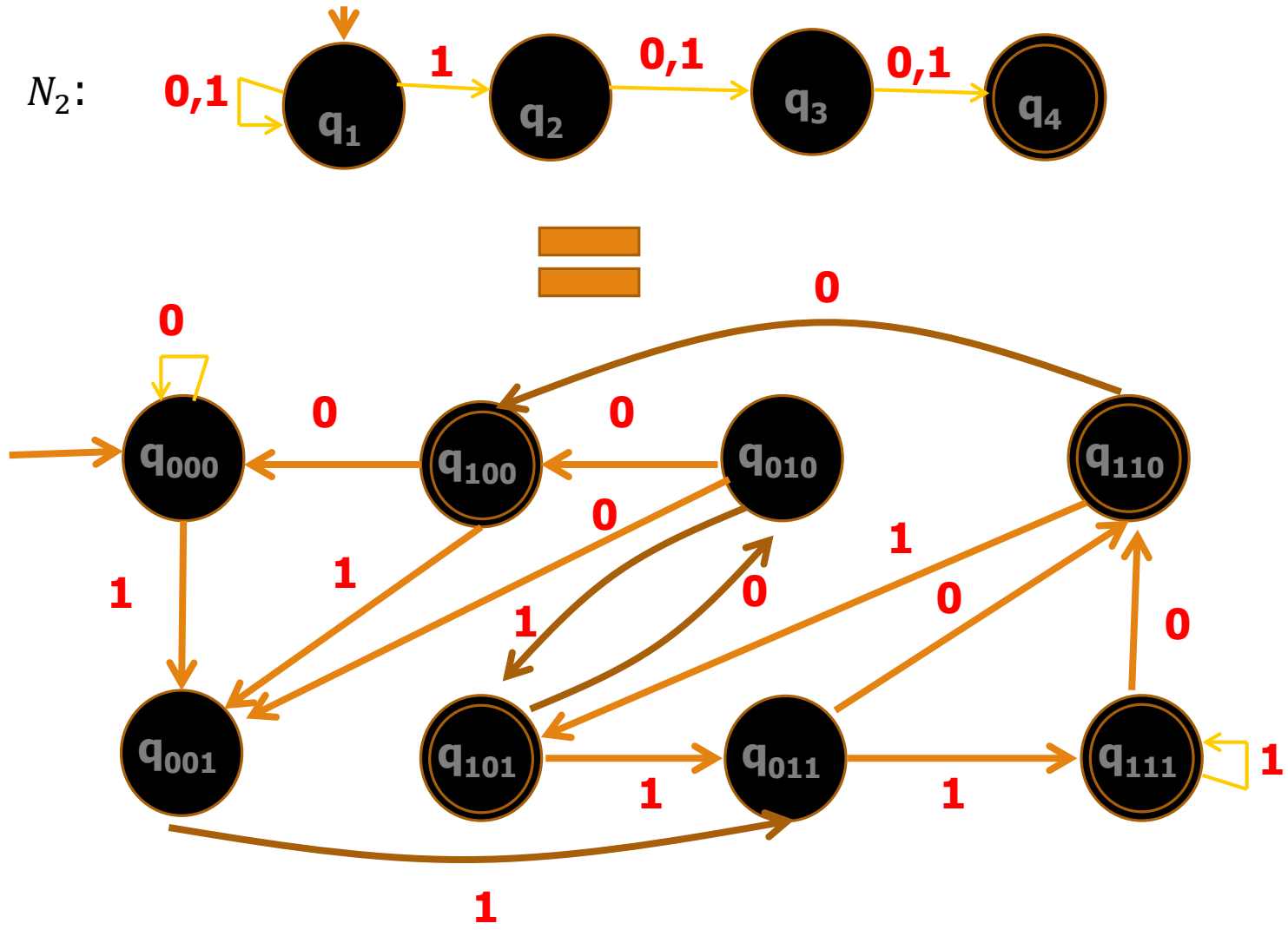
N_2:



Utilidade dos AFNs

- Construir AFNs é muitas vezes mais fácil que construir diretamente AFDs
  - Um AFN pode ser muito menor que sua contrapartida AFD
  - Ou se funcionamento pode ser mais fácil de entender
- Todo AFN pode ser convertido em um AFD equivalente (veremos adiante).

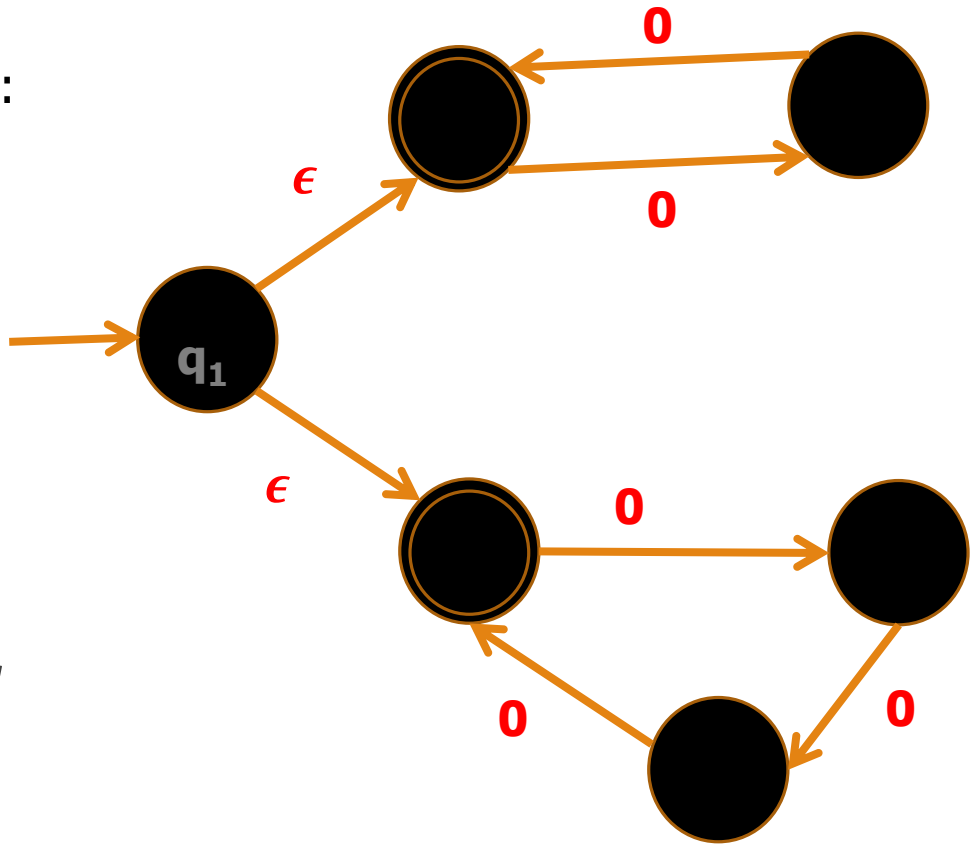
Utilidade dos AFNs



Exemplo 1.33

Considere o seguinte AFN N_3 que tem o alfabeto de entrada {0} consistindo de um único símbolo.

N_3:

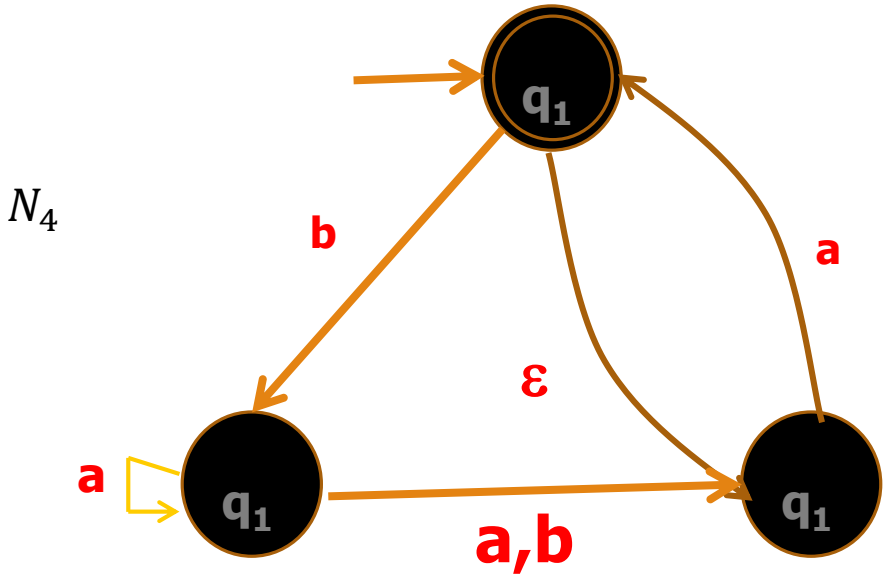


Verifica-se a conveniência de ser ter seta ϵ

Exemplo 1.35

Esse autômato aceita as cadeias ε, a, baba e baa?

E as cadeias b, bb, babba?



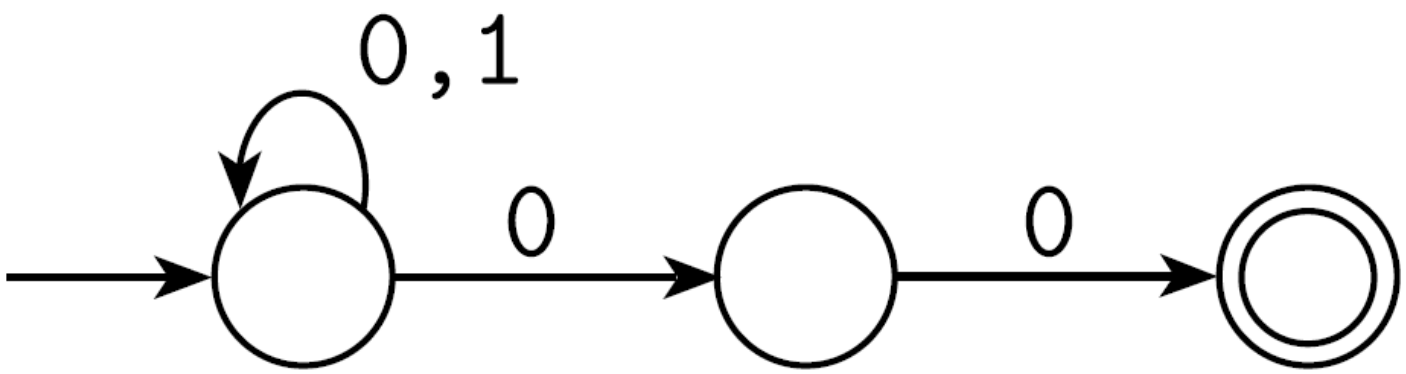
Exercícios em sala

Dê o diagrama de estados de AFNs que reconheça a linguagem {w|w termina em 00} com três estados

Exercícios em sala

Dê o diagrama de estados de AFNs que reconheça a linguagem {w|w termina em 00} com três estados

Resposta:



Exercícios em sala

Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$

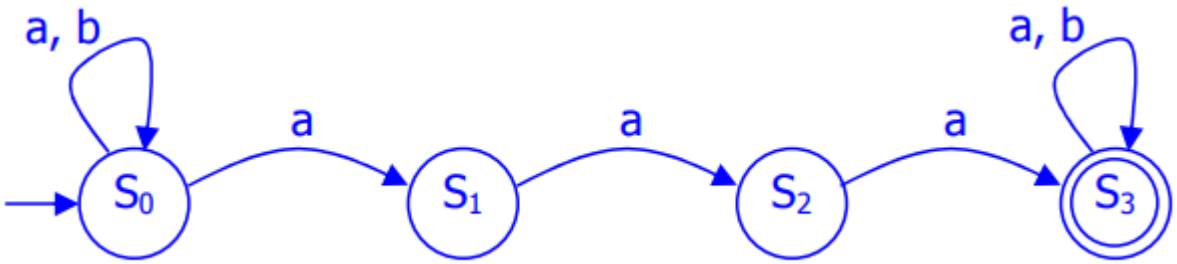
{w | aaa é subpalavra de w}

Exercícios em sala

Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$

{w | aaa é subpalavra de w}

Resposta:



Exercícios em sala

Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$

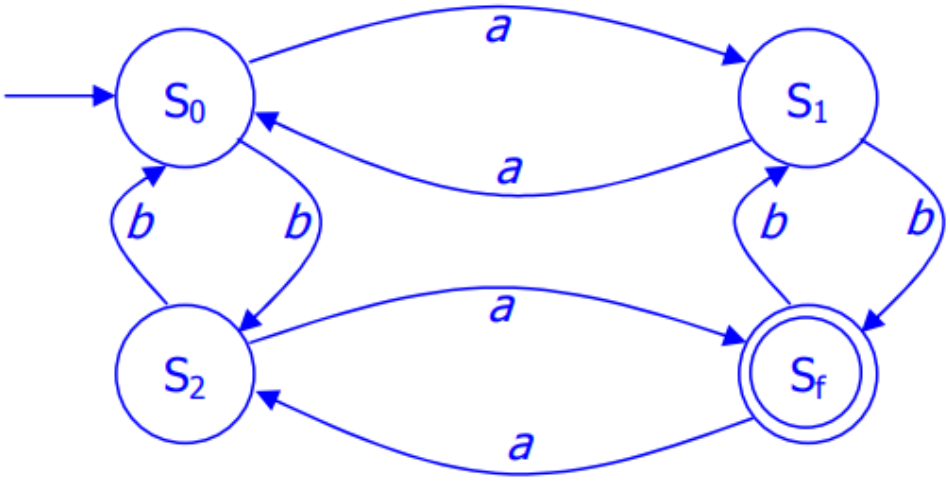
{w | w possui uma quantidade ímpar de a e de b}

Exercícios em sala

Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$

{w | w possui uma quantidade ímpar de a e de b}

Resposta:



Exercícios em sala

Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$

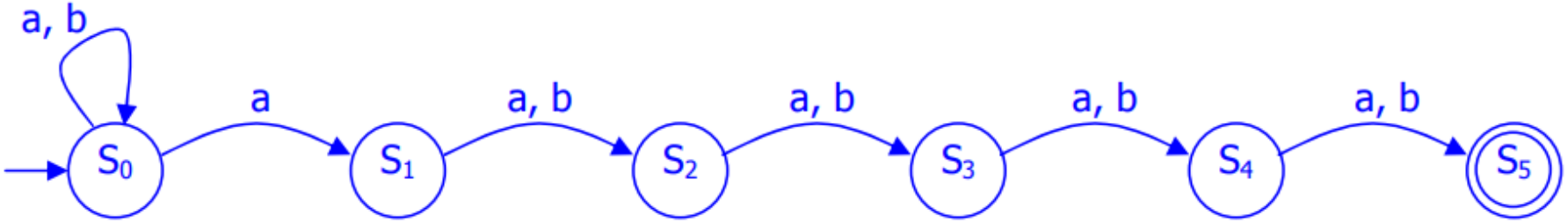
{w | o quinto símbolo da direita para a esquerda de w é a}

Exercícios em sala

Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$

{w | o quinto símbolo da direita para a esquerda de w é a}

Resposta:



AFN: Descrição Formal

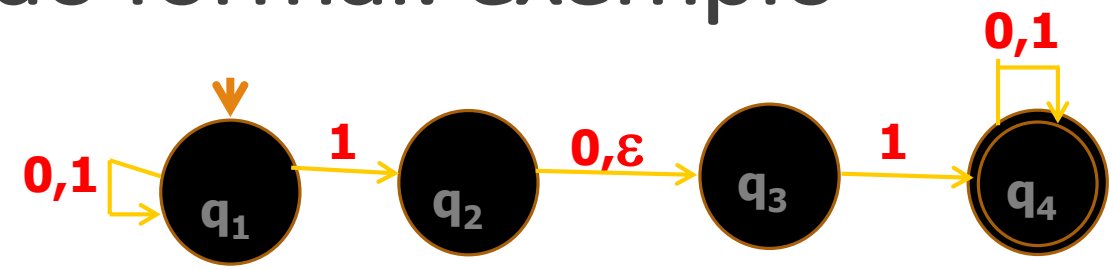
- Em um AFD a função de transição toma um estado e um símbolo de entrada e produz o próximo estado.
- Em um AFN a função de transição toma um estado e um símbolo de entrada *ou a cadeia vazia* e produz *o* ***conjunto de próximos estados possíveis***.
- Desse modo, vamos usar a notação $\Sigma_\varepsilon = \Sigma \cup \varepsilon$.
- E definir a função de transição da seguinte forma: $\delta: Q \times \Sigma_\varepsilon \rightarrow \mathcal{P}(Q)$.

AFN: Definição Formal

Um *autômato finito não-determinístico* é uma 5-upla (Q, Σ, δ, q_0, F), onde

1. Q é um conjunto finito denominado os *estados*,
2. Σ é um conjunto finito denominado *alfabeto*,
3. δ: $Q \times \Sigma_\varepsilon \rightarrow \mathcal{P}(Q)$ é a *função de transição,* $\mathcal{P}(Q)$ conjunto das partes de Q.
4. $q_0 \in Q$ é o *estado inicial*, e
5. $F \subseteq Q$ é o *conjunto de estados de aceitação (ou finais).*

Definição formal: exemplo



A definição formal desse AFN é $(Q, \Sigma, \delta, q_1, F)$, **onde:**

$Q=\{q_1,q_2,q_3,q_4\}$

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

q_1 é o estado inicial

$F=\{q_4\}$

δ	0	1	ε
q_1	$\{q_1\}$	$\{q_1,q_2\}$	$\varnothing$
q_2	$\{q_3\}$	$\varnothing$	$\{q_3\}$
q_3	$\varnothing$	$\{q_4\}$	$\varnothing$
q_4	$\{q_4\}$	$\{q_4\}$	$\varnothing$

Definição Formal de computação de um AFN

Seja N =(Q, Σ, δ, q_0, F) um AFN e w uma cadeia de Σ.

Dizemos que N *aceita* w se podemos escrever w como w = $y_1y_2...y_n$ onde cada $y_i \in \Sigma_\varepsilon$ e uma sequência de estados r_0, $r_1...r_n$ em Q existe com três condições:

1. $r_0 = q_0$,
2. $r_{i+1} \in \delta(r_i,y_{i+1})$ = , para i = 0,...,n-1; e
3. $r_n \in$ F.

Observe que $\delta(r_i,y_{i+1})$ ***é o conjunto de próximos estados permissíveis***.

Exercício do livro

1.7) Dê diagramas de estado de AFNs com o número especificado de estados reconhecendo cada uma das linguagens a seguir. Em todos os casos o alfabeto é {0,1}.

b. {w|w contém a a subcadeia 0101, isto é, w = x0101y para algum x e algum y}

c. {w | w contém um número par de 0s, ou contém exatamente dois 1s}

d. A linguagem {0} com dois estados