TEORIA DA COMPUTAÇÃO

AUTÔMATOS FORMAIS NÃO-DETERMINISTICOS

Determinismo X 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.

Determinismo X Não-Determinismo

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

Determinismo X Não-Determinismo

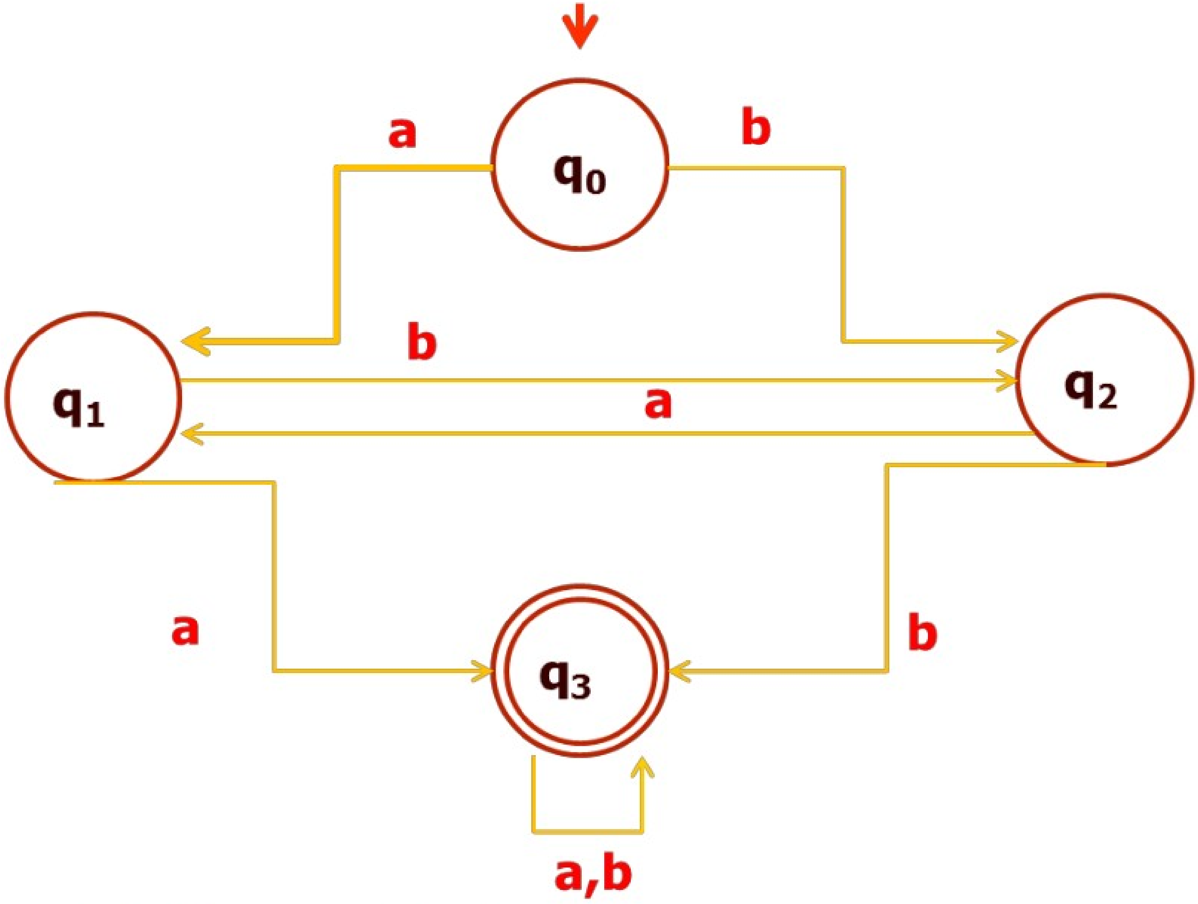
Algumas diferenças práticas:

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

Determinismo X Não-Determinismo

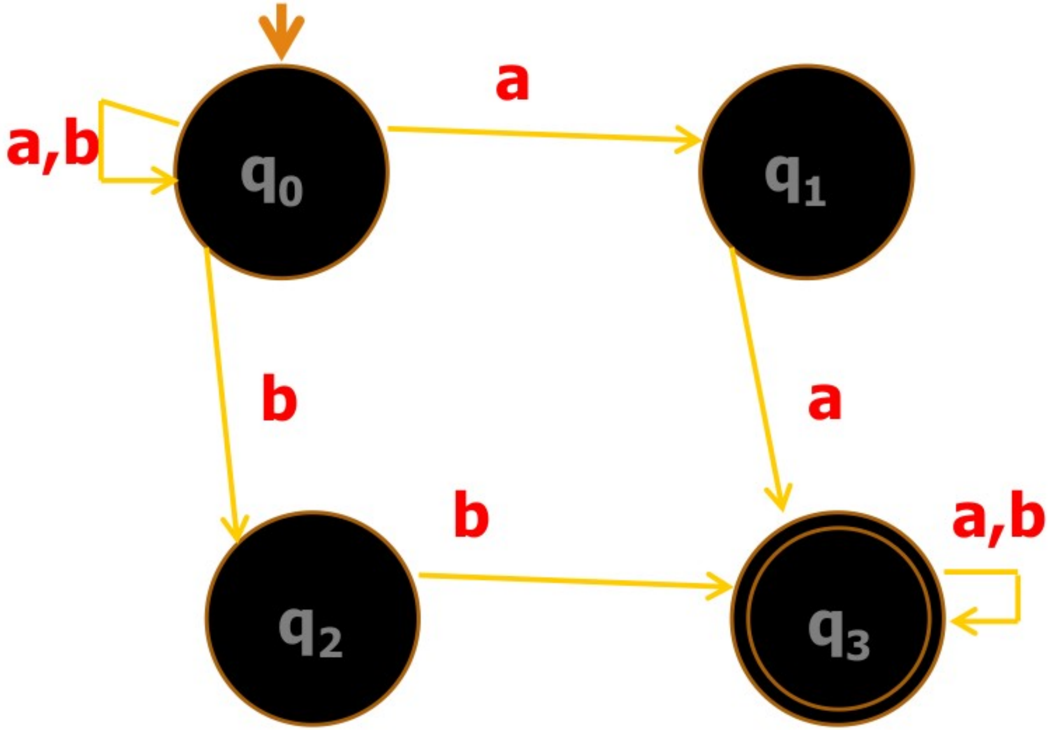
Considere um AFN N 1 que aceita as cadeias sobre o alfabeto {a,b} que possuem aa ou bb como subcadeia.

Determinismo X Não-Determinismo



Todo estado de um AFD sempre tem exatamente uma seta de transição saindo para cada símbolo do alfabeto

Determinismo X Não-Determinismo



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

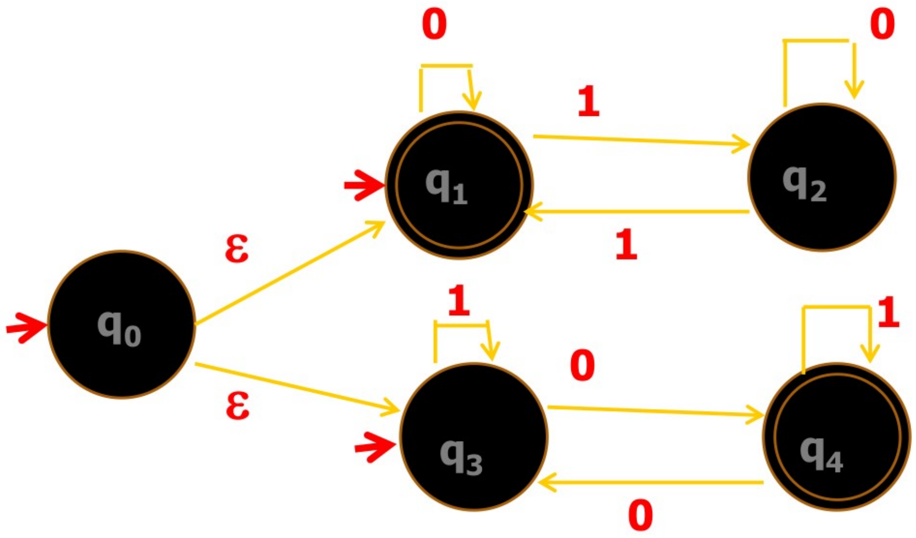
Determinismo X Não-Determinismo

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.

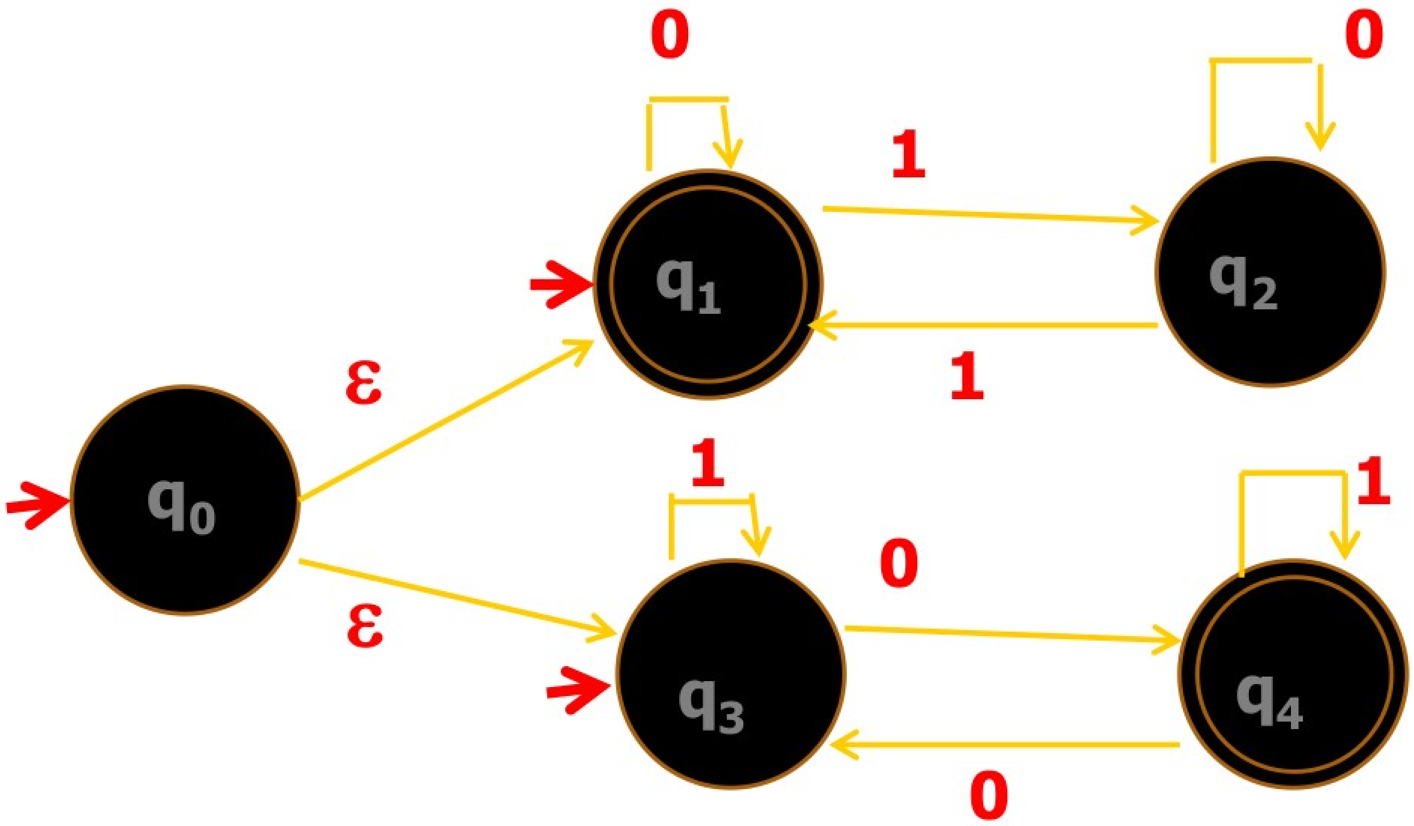
Autômatos Finitos Não-Determinísticos

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



Autômatos Finitos Não-Determinísticos

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

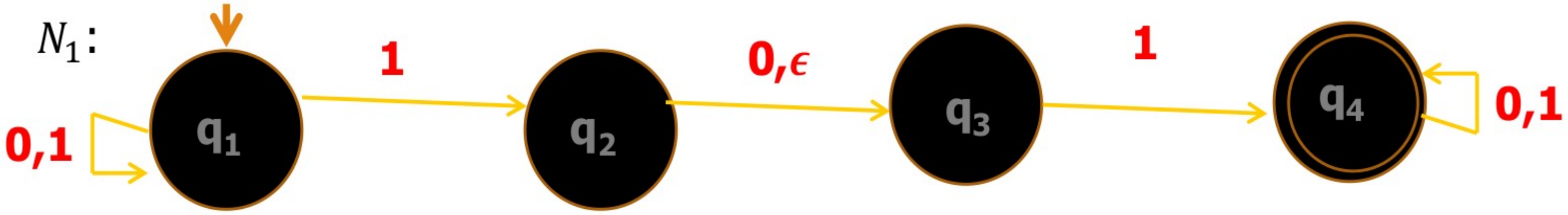


Autômatos Finitos Não-Determinísticos

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 N 1:.

Autômatos Finitos Não-Determinísticos



Autômatos Finitos Não-Determinísticos

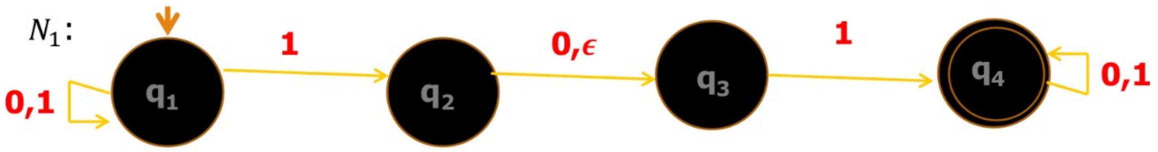
A computação em um AFN:

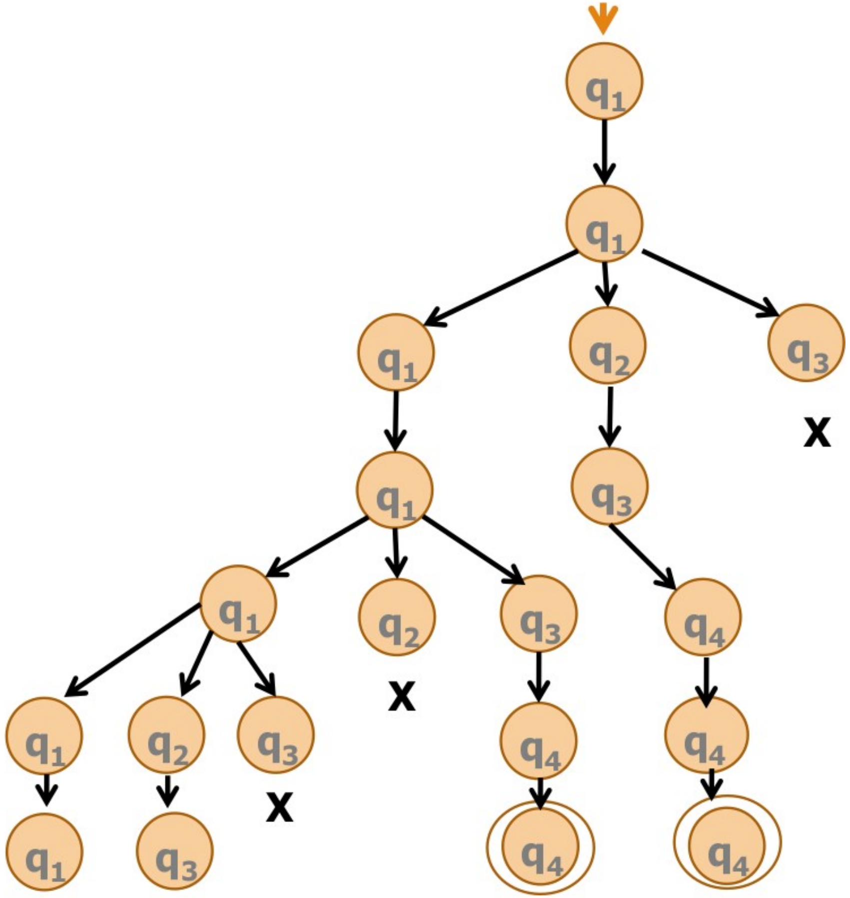
- Não-determinismo pode ser visto como uma espécie de computação paralela na qual múltiplos e independentes “processos” ou “threads” podem estar rodando concorrentemente.
- Se pelo menos um desses processos aceita então a computação inteira aceita

A computação em um AFN

Outra forma de pensar:
Árvore de possibilidades
Entrada: 010110

Qual linguagem N1 reconhece? N1 aceita todas as cadeias que contêm 101 ou 11 como uma subcadeia

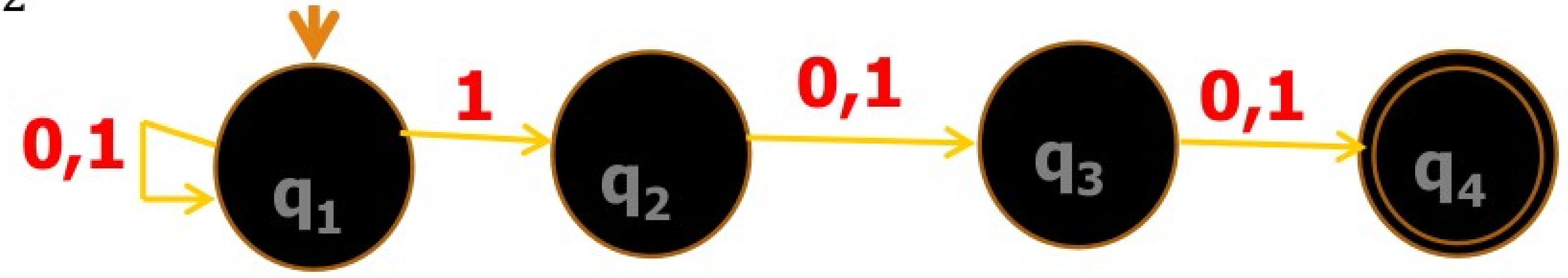




Exemplo

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).

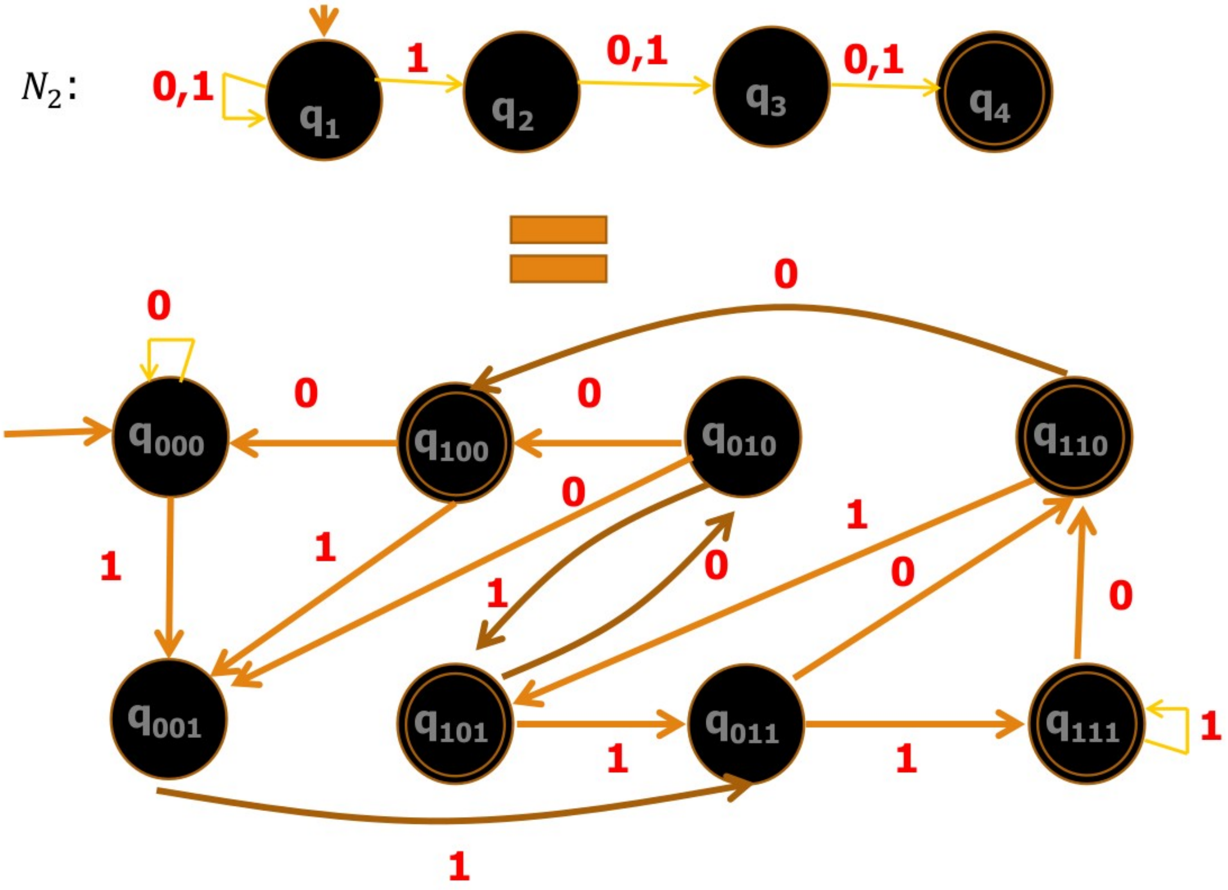
N_2:



Utilidade dos AFNs

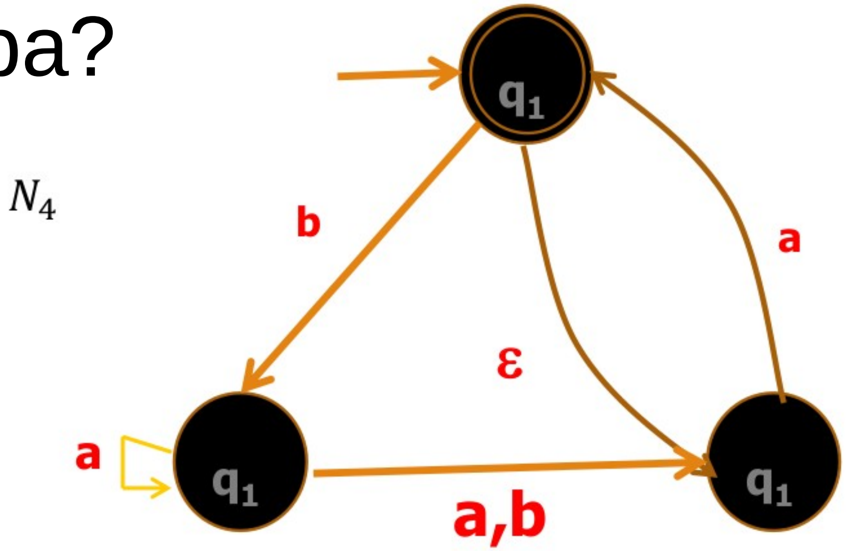
- Construir AFNs é muitas vezes mais fácil que construir diretamente AFDs.
  - Um AFN pode ser muito menor que sua contrapartida AFD;
  - Ou seu funcionamento pode ser mais fácil de entender.
- Todo AFN pode ser convertido em um AFD equivalente

Utilidade dos AFNs



Exemplo

- Esse autômato aceita as cadeias ε, a, baba e baa?
- E as cadeias b, bb, babba?

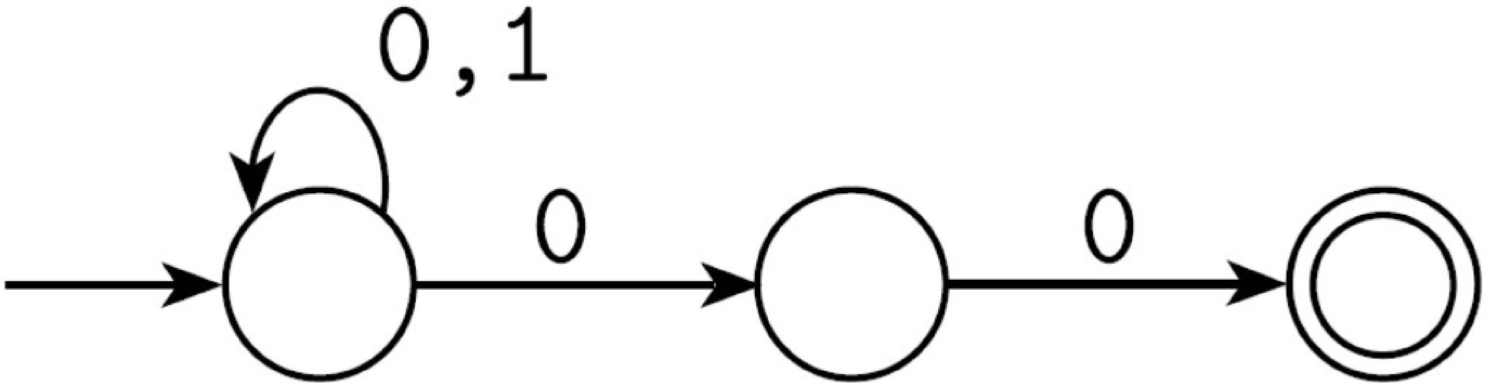


Exercícios em sala

- 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

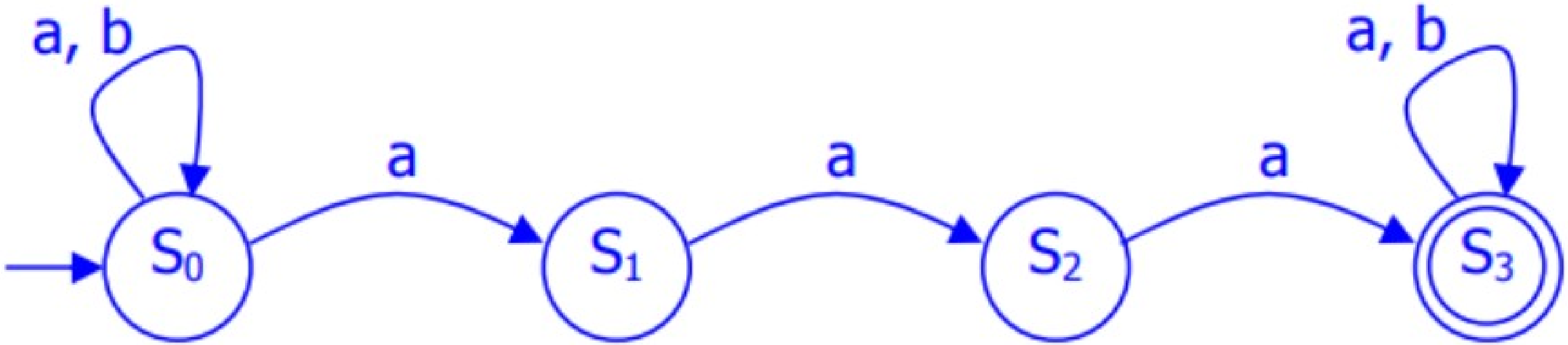


Exercícios em sala

- Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$
- $\{w \mid aaa$ é sub-palavra de $w\}$

Exercícios em sala

- Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$
- $\{w \mid aaa$ é sub-palavra de $w\}$

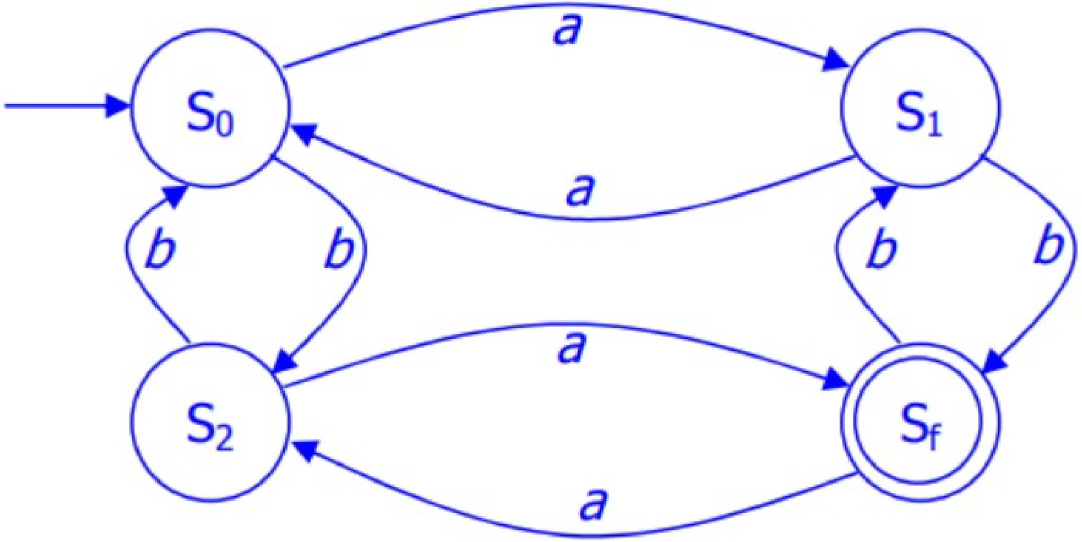


Exercícios em sala

- Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$
- $\{w \mid w$ possui uma quantidade ímpar de a e de b$\}$

Exercícios em sala

- Desenvolva autômatos que reconheçam as seguintes linguagens. $\Sigma = \{a,b\}$
- $\{w \mid w$ possui uma quantidade ímpar de a e de b$\}$

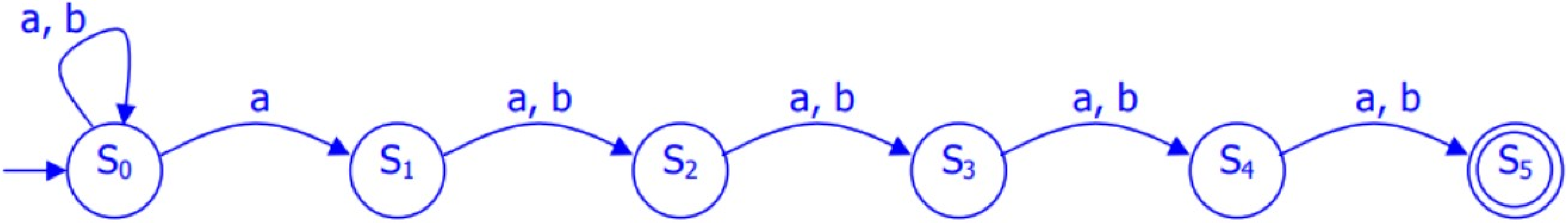


Exercícios em sala

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



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.
- Observe que a função δ neste caso não retorna um estado, mas sim um conjunto de estados, em particular pode ser o conjunto vazio. Assim a assinatura da função deve ser $\delta : Q \times \Sigma_\varepsilon \rightarrow \wp(Q)$ onde $\Sigma_\varepsilon = \Sigma \cup \varepsilon$.

AFN: Definição Formal

- Um autômato finito não-determinístico é uma 5-upla (Q,Σ,δ,q_0,F)
  - Q é um conjunto finito denominado os estados,
  - Σ é um conjunto finito denominado alfabeto,
  - $\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)$ é a função de transição, P(Q) conjunto das partes de Q.
  - $q0 \in Q$ é o estado inicial,
  - $F \subseteq Q$ é o conjunto de estados de aceitação (ou finais).

Definição formal: exemplo

A definição formal desse AFN é (Q,Σ,δ,q_1,F), onde:

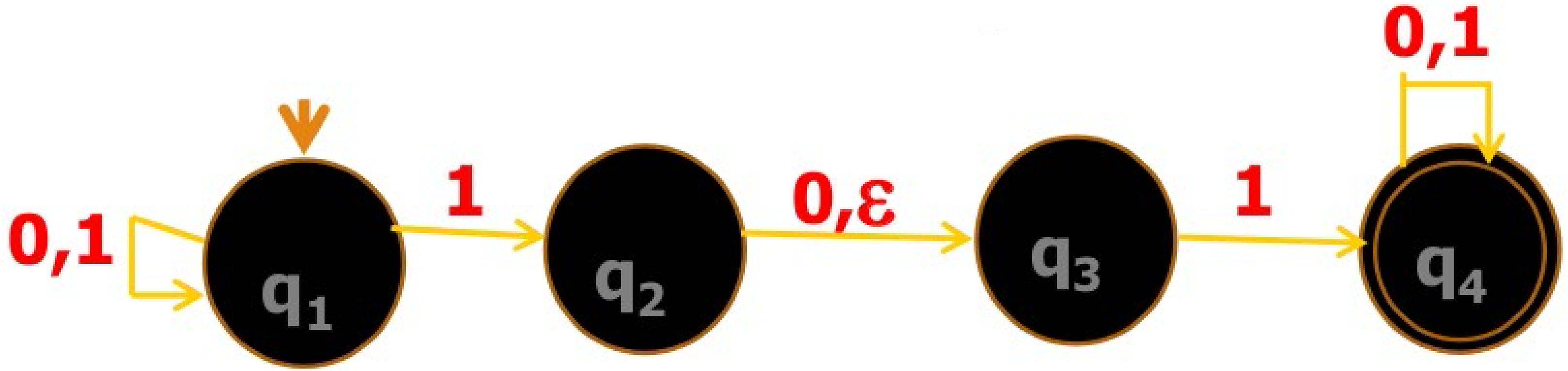
$Q=\{q_1,q_2,q_3,q_4\}$

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

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 $N = (Q, \Sigma, \delta, q_0, F)$ um AFN e w uma cadeia de Σ;
- Dizemos que N aceita w se podemos escrever w como $w = y_1 y_2 \ldots y_n$ onde cada $y_i \in \Sigma_\varepsilon$ e uma sequência de estados $r_0, r_1 \ldots r_n$ em Q existe com três condições:

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

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

Referências

- PROFESSOR: LUCAS CAMBUIM do site https://www.cin.ufpe.br/~lfsc/cursos/teoriadainformacao/unidade%201/cap%201_4%20-%20automatos%20e%20linguagem%20nao-determinismo.pdf
- Em 07/03/2020