Teoria da Computação

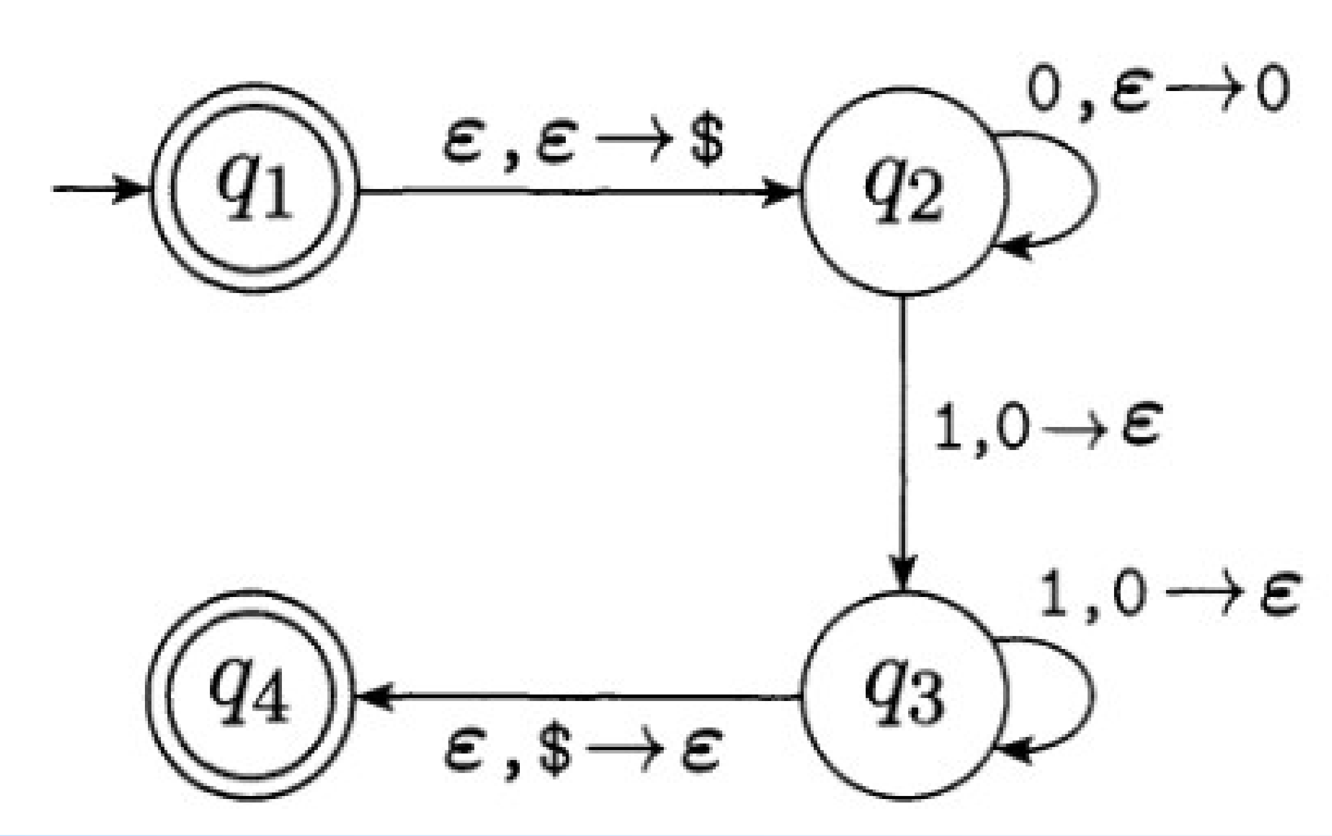
Autômatos de Pilha

Professor Thiago Alves

Introdução

- São como autômatos finitos não determinísticos mas possuem uma pilha para guardar símbolos

Introdução



q_1
$\varepsilon, \varepsilon \rightarrow \$$
q_2
$0, \varepsilon \rightarrow 0$
$1, 0 \rightarrow \varepsilon$
q_3
$1, 0 \rightarrow \varepsilon$
q_4
$\varepsilon, \$ \rightarrow \varepsilon$

Autômato de Pilha

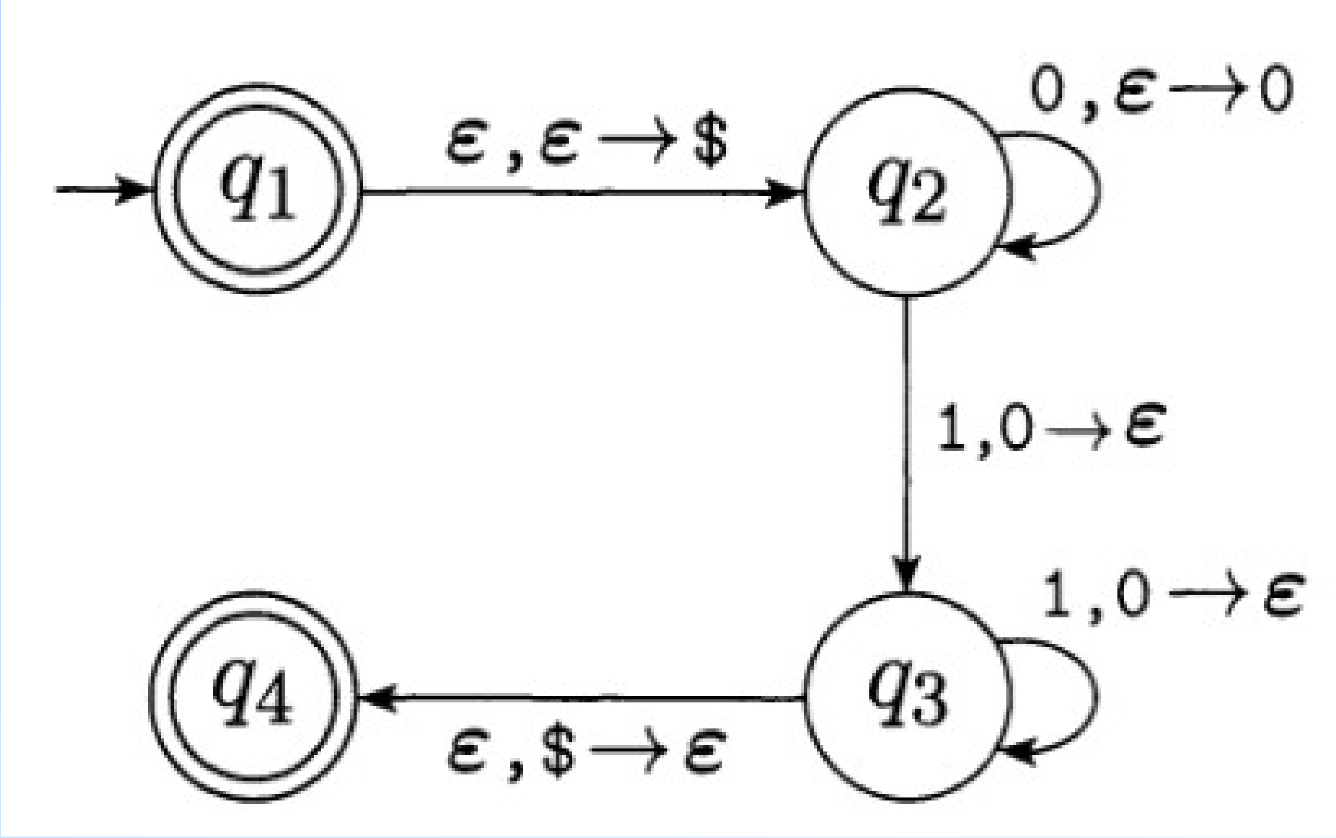
- Pode escrever símbolos na pilha para ler posteriormente
- Por conta da pilha, um símbolo é escrito no topo e lido do topo

Processamento

- Inicia no estado inicial e com a pilha vazia
- Vai para os próximos estados dependendo do estado atual, do símbolo da entrada e do símbolo da pilha
- Pode inserir e/ou remover símbolos na pilha

Exemplo

- Processamento da string 0011

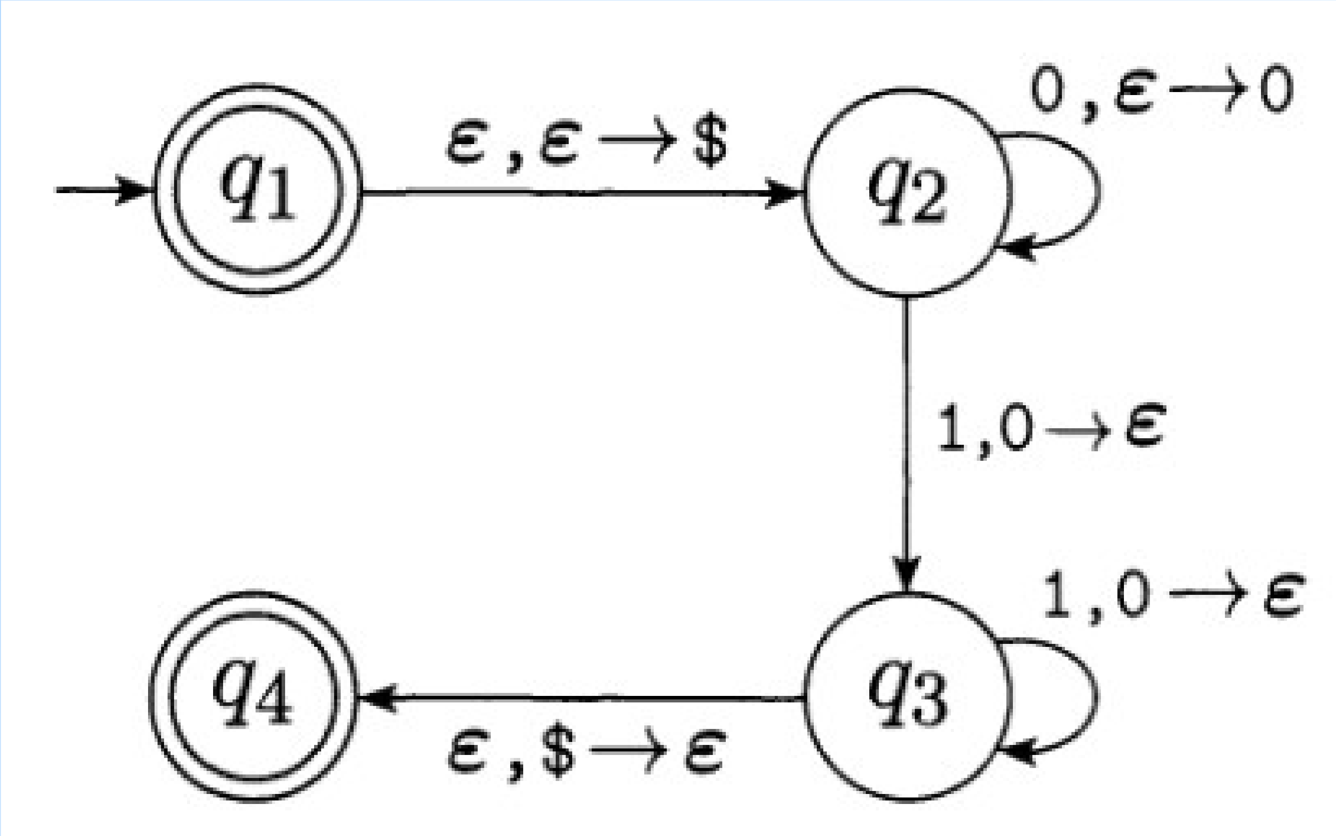


Processamento

- Final do processamento ocorre quando todos os símbolos da string forem consumidos
- Aceita uma string se algum dos estados ao final do processamento for final
- Não determinístico

Exemplo

- Processamento da string 00111

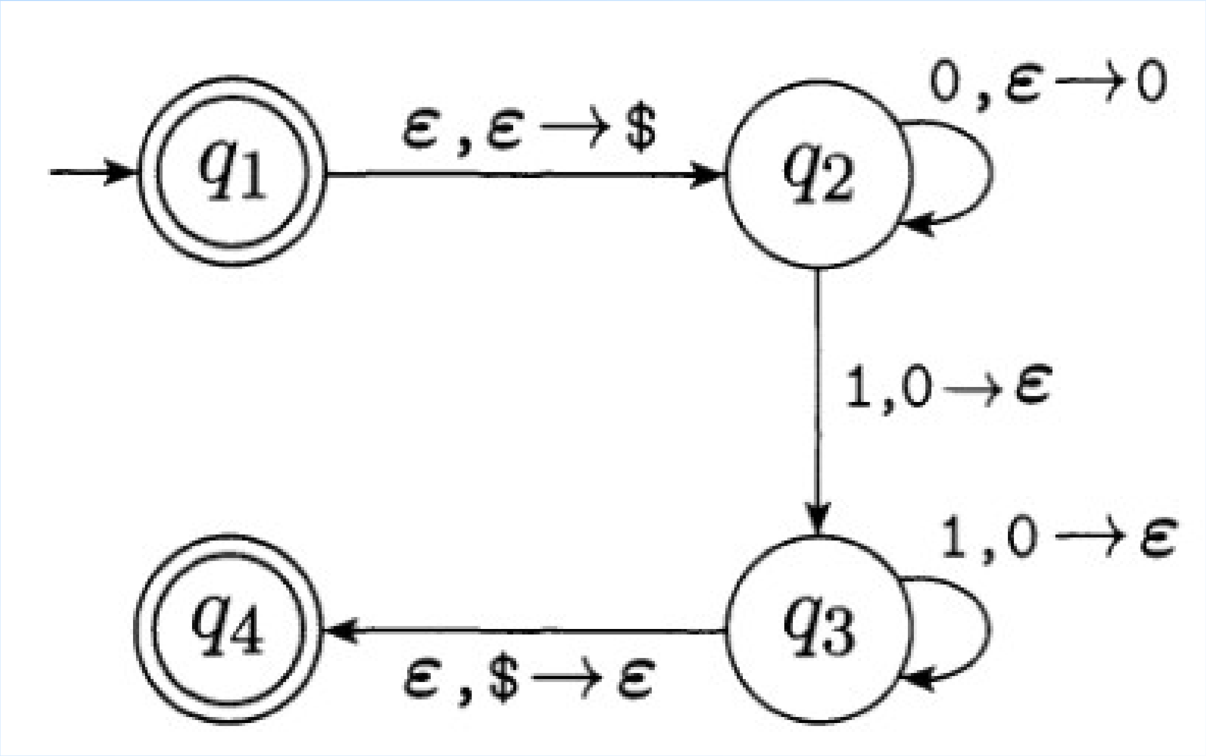


Exemplo

- Uso da pilha para guardar os 0's vistos
- Depois tira um 0 para cada 1 visto

Exemplo

- Aceita exatamente as strings em $\{0^n1^n \mid n \geq 0\}$



Formalização

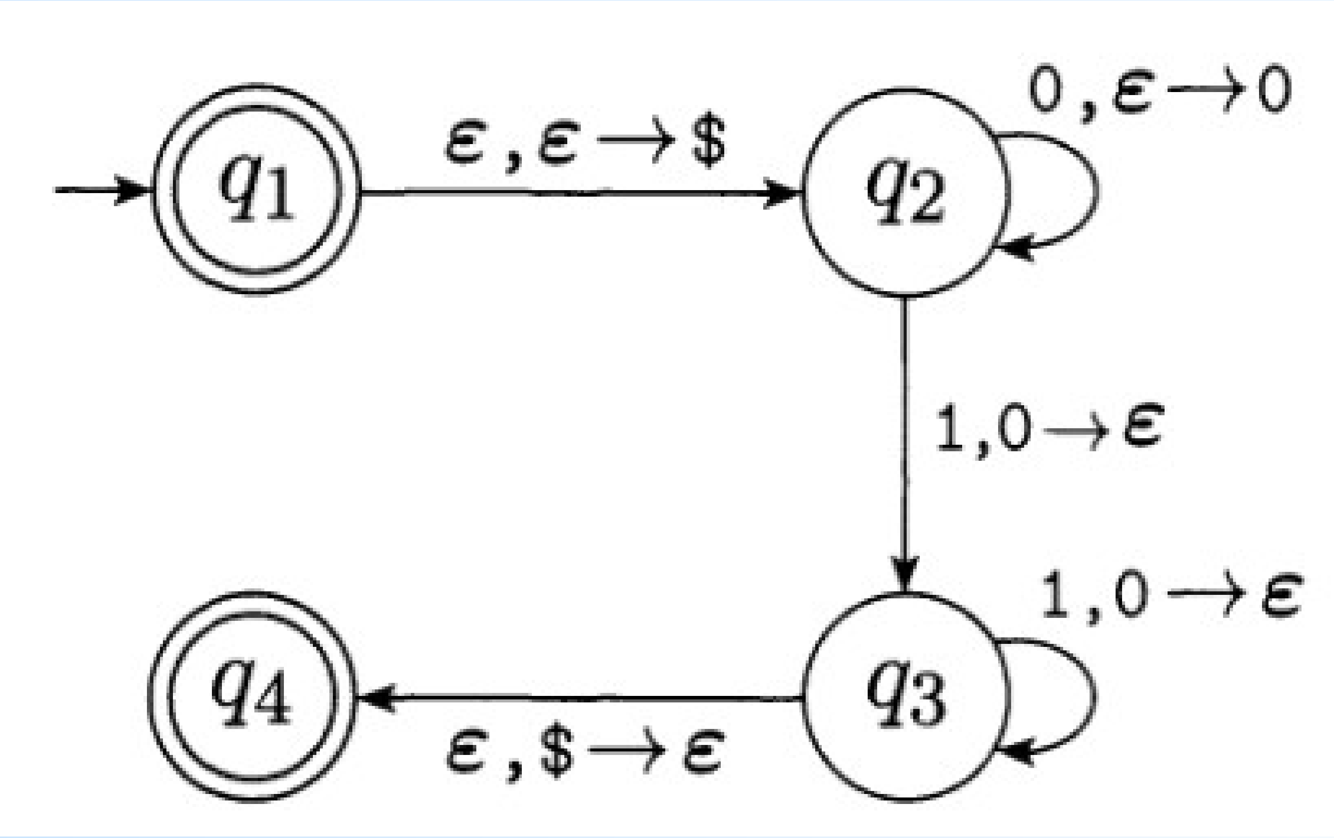
- Uma tupla $(Q,\Sigma,\Gamma,\delta,q_0,F)$
- Um conjunto finito de estados Q
- Um alfabeto de entrada Σ
- Um alfabeto da pilha Γ
- Uma função de transição δ
  - $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \rightarrow P(Q \times (\Gamma \cup \{\varepsilon\}))$
- Um estado inicial q_0 em Q
- Um conjunto de estado finais $F \subseteq Q$

Função de Transição

- $\delta(q, a, b) = \{(p_1,b_1),\ldots,(p_k,b_k)\}$
- De forma não determinística vai para um estado, retira b da pilha e escreve um símbolo na pilha
- $\delta(q, a, \varepsilon) = \{(p_1,b_1)\}$ não retira da pilha
- $\delta(q, a, b) = \{(p_1,\varepsilon)\}$ não coloca na pilha

Exemplo

- Vamos formalizar:



Exemplo

- $P = (Q,\Sigma,\Gamma,\delta,q_1,F)$
- $Q = \{q_1,q_2,q_3,q_4\}$
- $\Sigma = \{0,1\}$
- $\Gamma = \{\$, 0\}$
- $F = \{q_1,q_4\}$

Exemplo

- $P = (Q,\Sigma,\Gamma,\delta,q_1,F)$
- $\delta(q_1,0,0) = \{\}$
- $\delta(q_1,\varepsilon,\varepsilon) = \{(q_2,\$)\}$
- $\delta(q_2,0,\varepsilon) = \{(q_2,0)\}$
- ...
- $\delta(q_2,1,0) = \{(q_3,\varepsilon)\}$

Exemplo

- $P = (Q, \Sigma, \Gamma, \delta, q_1, F)$
- $\delta =$

Input:	0			1			ε		
Stack:	0	\$	ε	0	\$	ε	0	\$	ε
q_1									$\{(q_2, \$)\}$
q_2			$\{(q_2, 0)\}$	$\{(q_3, \varepsilon)\}$					
q_3				$\{(q_3, \varepsilon)\}$				$\{(q_4, \varepsilon)\}$	
q_4									

Exemplo

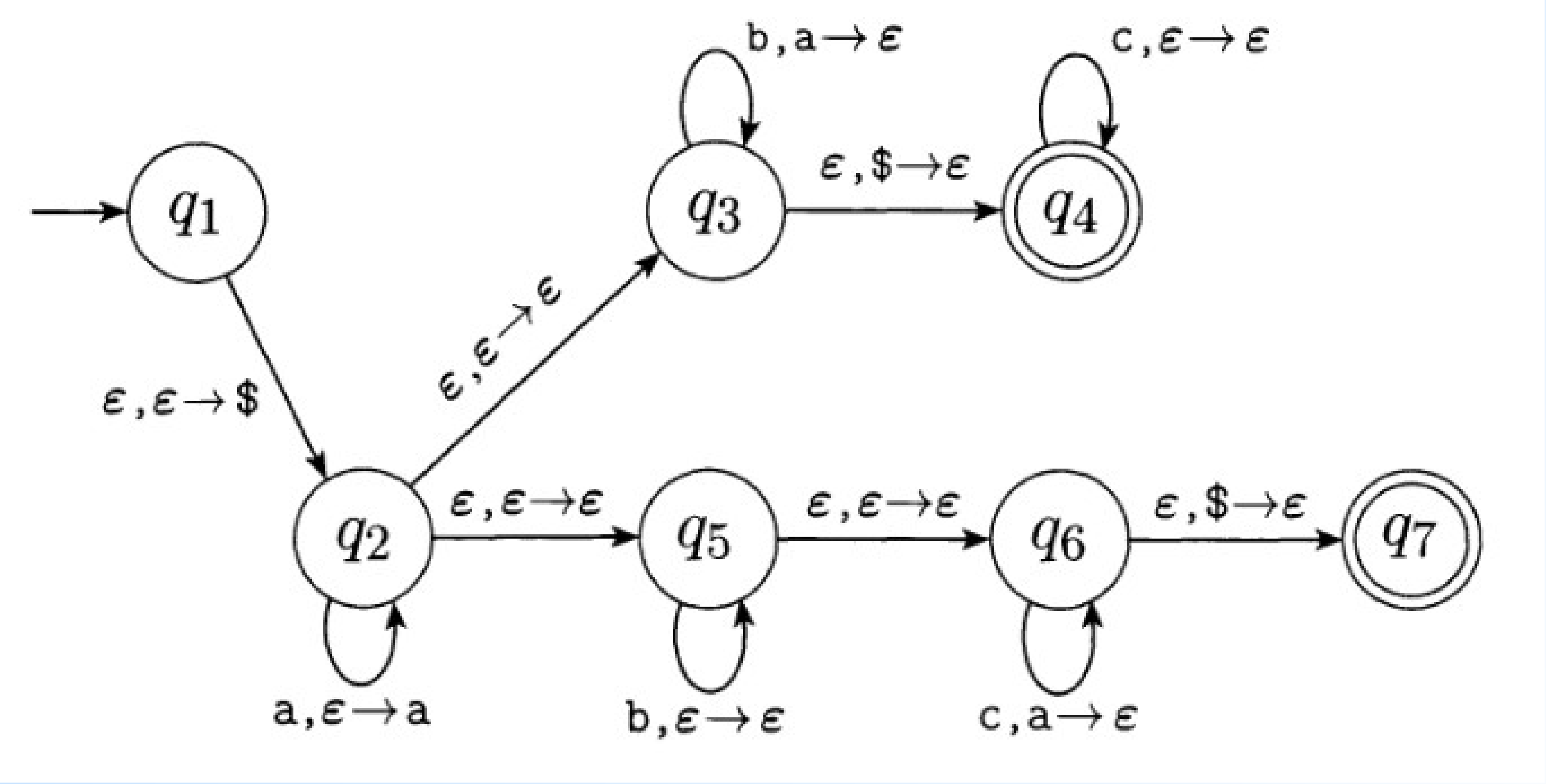
- Fazer fazer um autômato de pilha para aceitar as strings da linguagem $\{a^i b^j c^k \mid i=j \text{ ou } i=k\}$

Exemplo

- Fazer fazer um autômato de pilha para aceitar as strings da linguagem $\{a^i b^j c^k \mid i=j \text{ ou } i=k\}$
- Guardar os a's vistos
- Usar o não determinismo para verificar se a quantidade de a's é igual a de b's ou de c's

Exemplo

- $\{a^i b^j c^k \mid i=j \text{ ou } i=k\}$



Exemplo

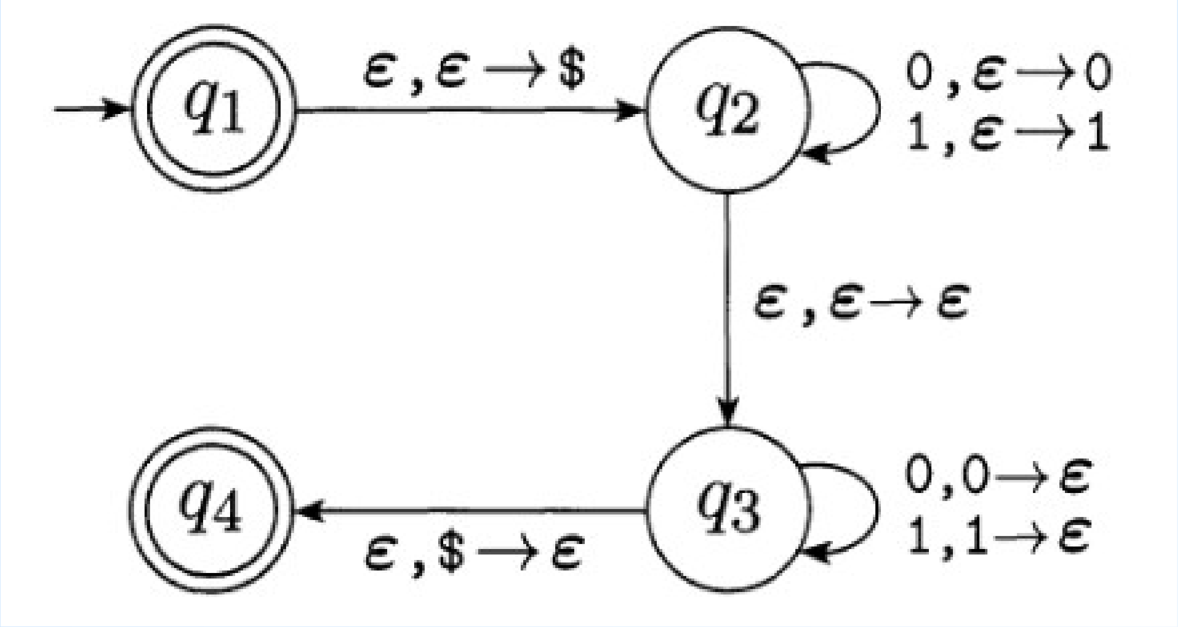
- Vamos fazer um autômato de pilha para $\{ww^r \mid w \in \{0,1\}^*\}$

Exemplo

- Vamos fazer um autômato de pilha para $\{ww^r \mid w \in \{0,1\}^*\}$
  - Colocar o símbolo visto na pilha
  - A cada passo usar o não determinismo para considerar o meio da string já alcançado
  - Retirar da pilha verificando se é igual ao próximo símbolo da entrada

Exemplo

- Vamos fazer um autômato de pilha para $\{ww^r \mid w \in \{0,1\}^*\}$



Resultados

- A linguagem L(P) de um autômato de pilha P é o conjunto de strings que ele aceita

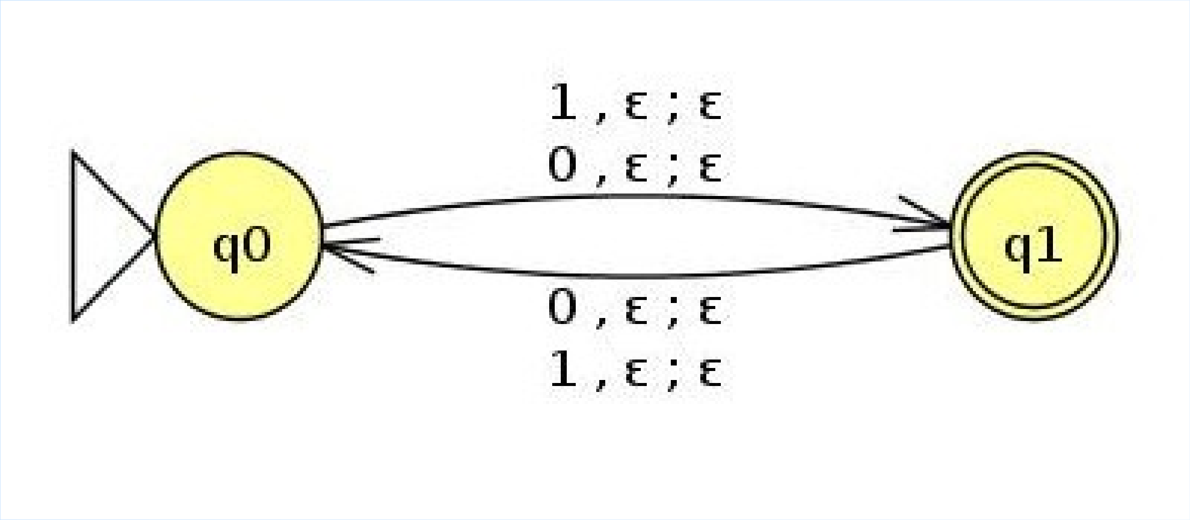
- Se A é uma linguagem e P é um autômato de pilha tal que L(P) = A então dizemos que P reconhece A

Exercício

- Faça um autômato de pilha para reconhecer a linguagem $\{w \in \{0,1\}^* \mid w$ tem tamanho ímpar$\}$

Exercício

- {w ∈ {0,1}* | w tem tamanho ímpar}



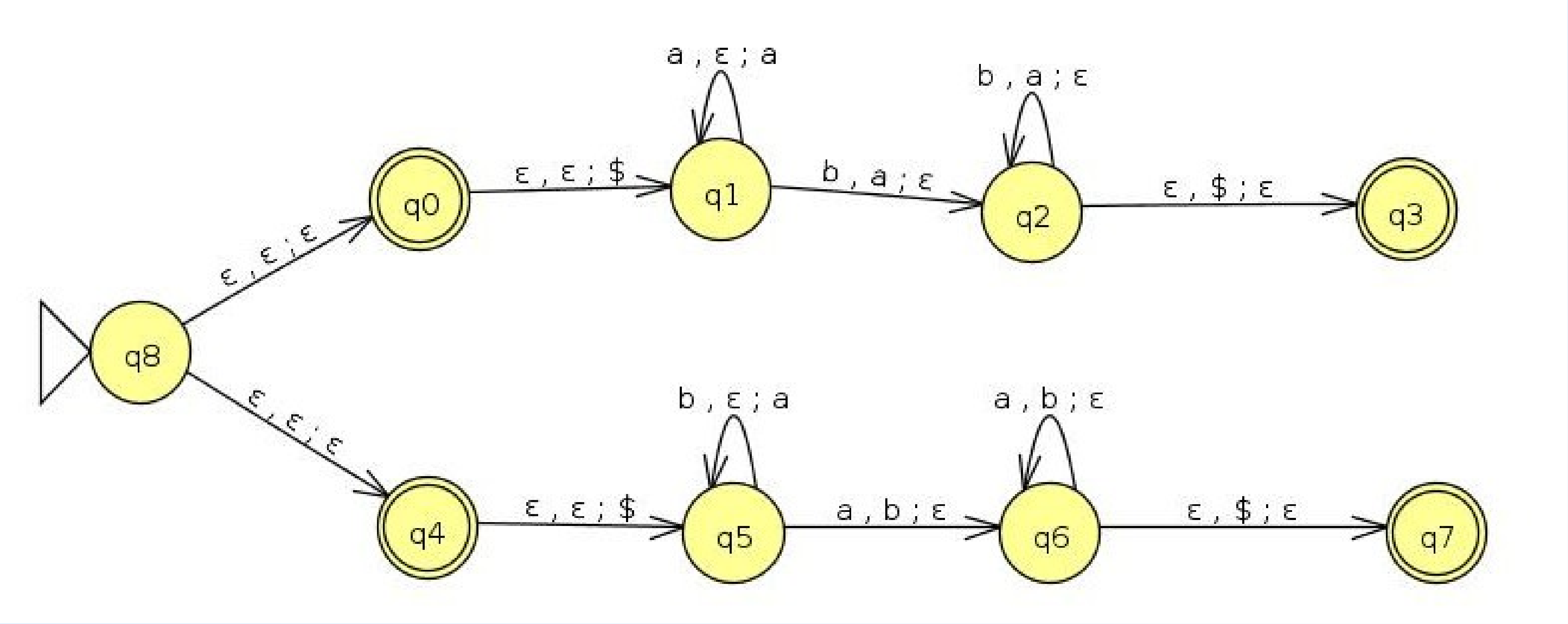
Exercício

- Faça um autômato de pilha para reconhecer a linguagem

  $\{w \in \{a,b\}^* \mid w = a^n b^n$ ou $b^n a^n$ com $n \geq 0 \}$

Exercício

- $\{w \in \{a,b\}^* \mid w = a^n b^n \text{ ou } b^n a^n \text{ com } n \geq 0\}$



Exercício

- Faça um autômato de pilha para reconhecer a linguagem

  $\{w \in \{a,b\}^* \mid$ a quantidade de a's em w é igual a quantidade de b's$\}$

Exercício

- Ideia:
  - Se o símbolo visto for a e no topo da pilha tiver b, tirar o b
  - Se o símbolo visto for a e no topo da pilha tiver b, tirar o b
  - Nos outros casos coloca o símbolo visto na pilha
  - Aceita se chegar no fundo da pilha assim que processar toda a string de entrada

Exercício

- {w ∈ {a,b}* | a quantidade de a's em w é igual a quantidade de b's}