Teoria da Computação

Autômatos Finitos Não Determinísticos

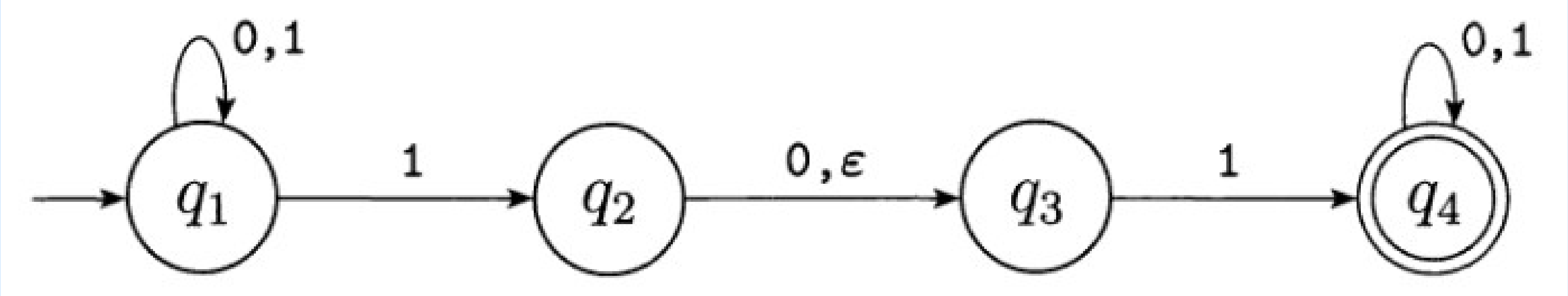
Prof. Thiago Alves

Introdução

- Transições dos AFD são determinísticas
- Podemos estender com a capacidade ir para mais de um estado com um mesmo símbolo
- Pode estar em vários estados ao mesmo tempo

Introdução

N_1:



Autômato Finito Não Determinístico

- Um autômato finito *não determinístico* tem a habilidade de estar em vários estados ao mesmo tempo
- Transições de um estado com um símbolo podem levar a um vários estados

Autômato Finito Não Determinístico

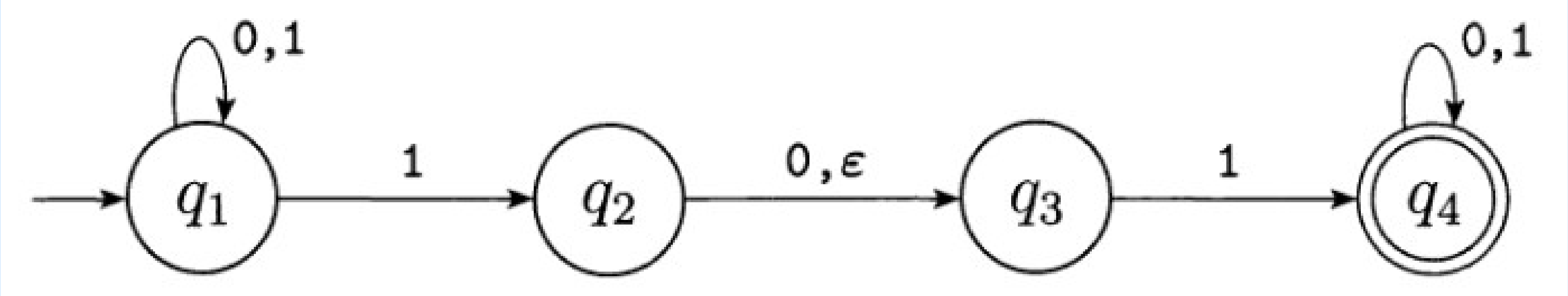
- O AFN também permite "pular" estados usando transições ε
- A execução de um dos estados pode parar caso não tenha saída com o símbolo da string

Processamento

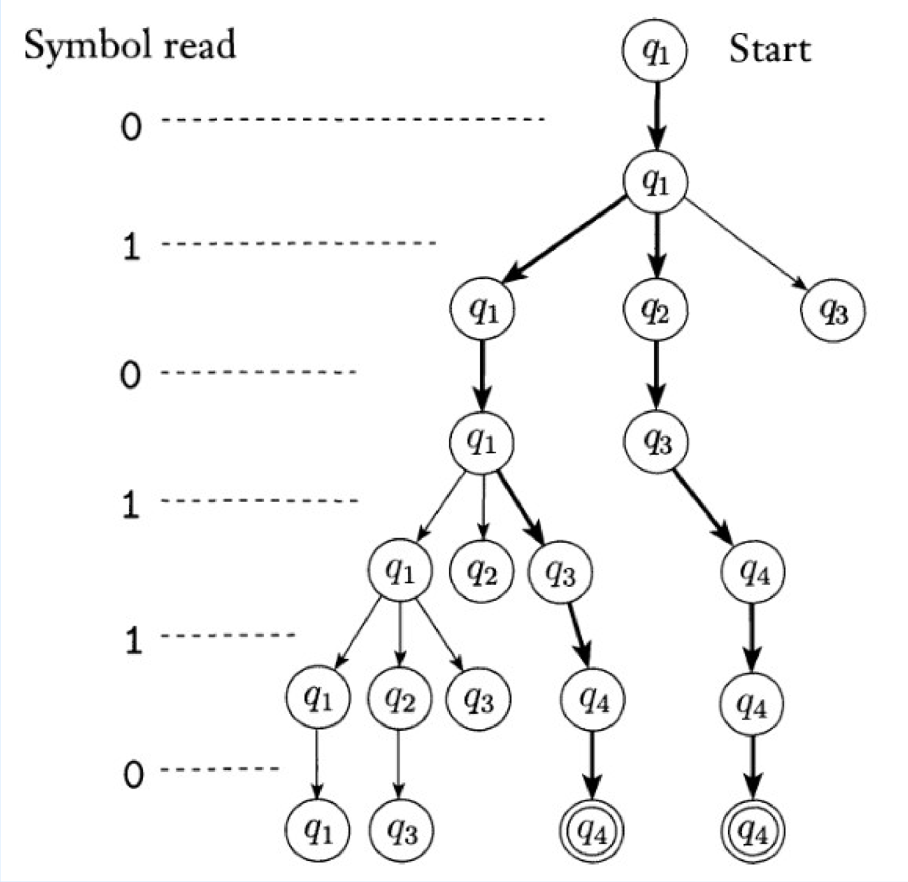
- Inicia no estado inicial
- Pode terminar em vários estados
- AFN aceita uma string se alguma sequência de escolhas leva a um estado final
- Intuitivamente: o AFN sempre "escolhe corretamente."

Processamento

- Processamento da string 010110

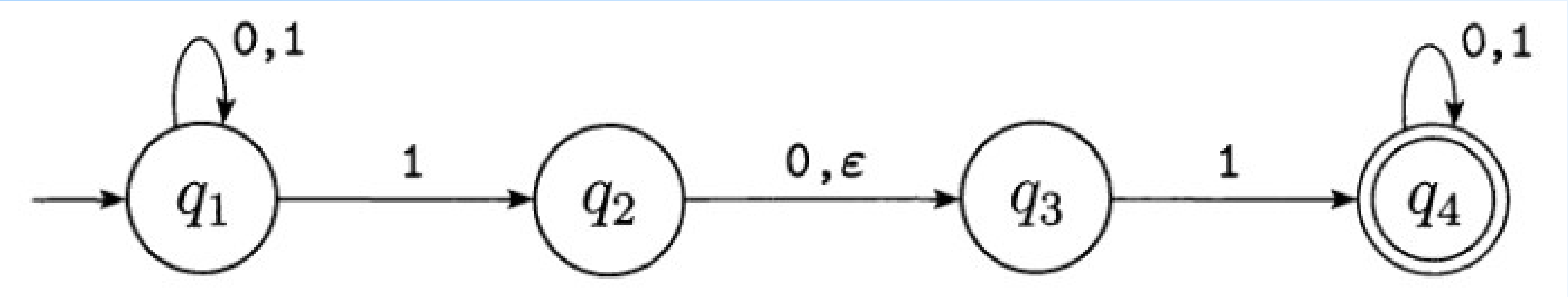


Processamento



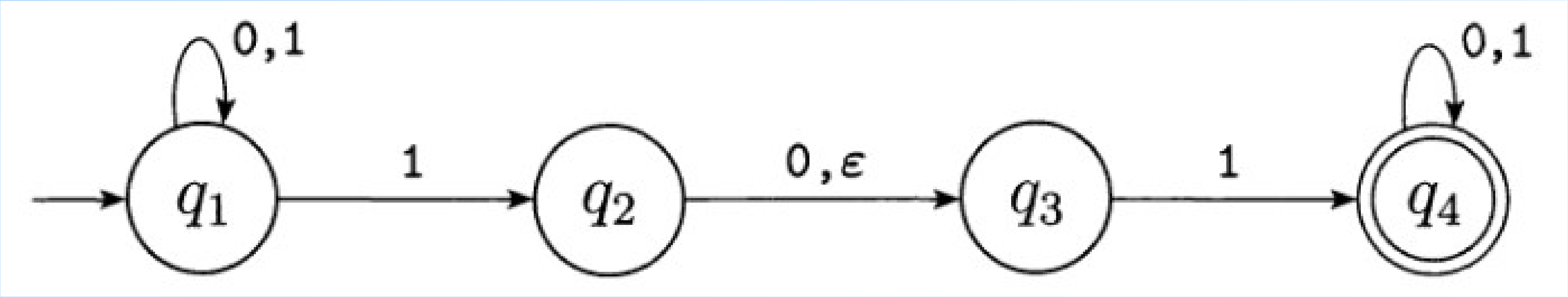
Processamento

- Processamento da string 010



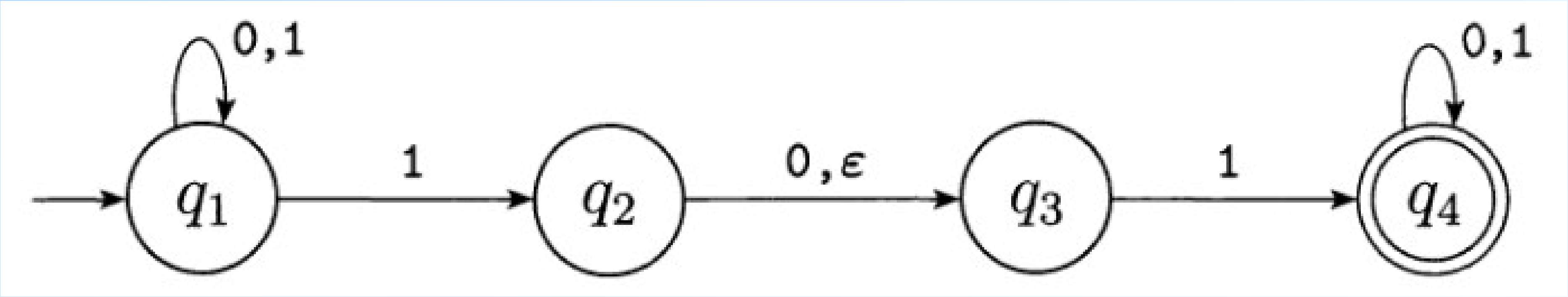
Processamento

- Que strings o AFN N_1 aceita?



Processamento

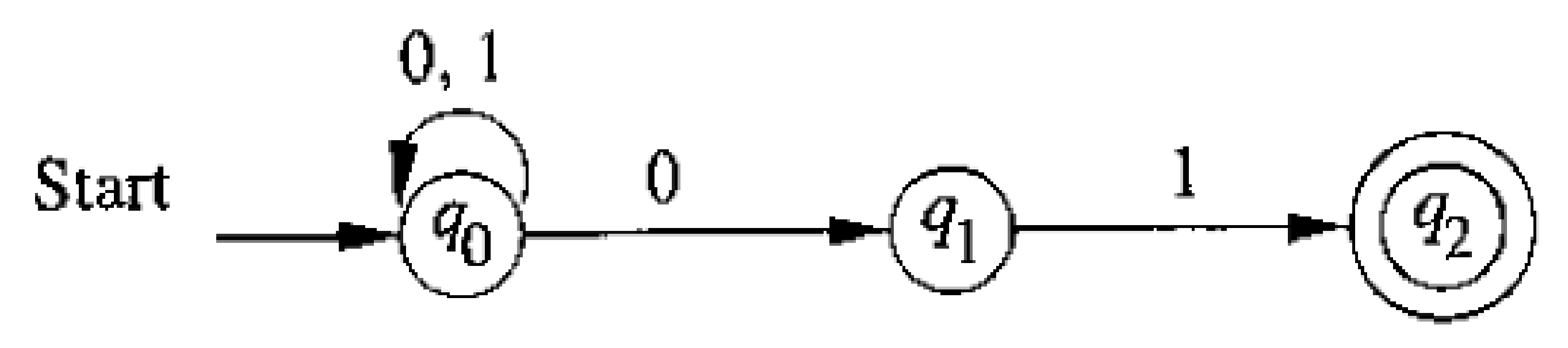
- Que strings o AFN N_1 aceita?



- Strings com 11 ou 101 como substring

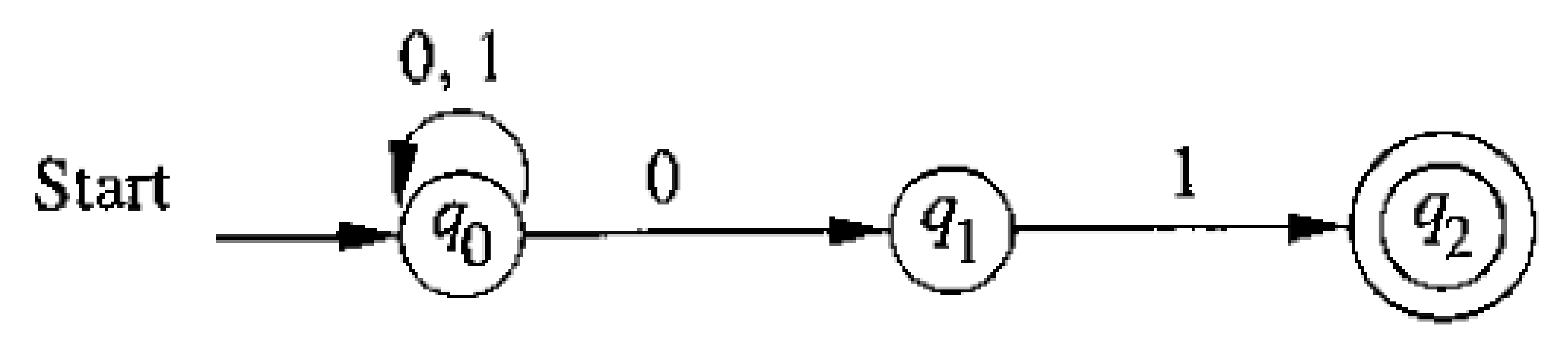
Exemplo

- Processamento da string 0101



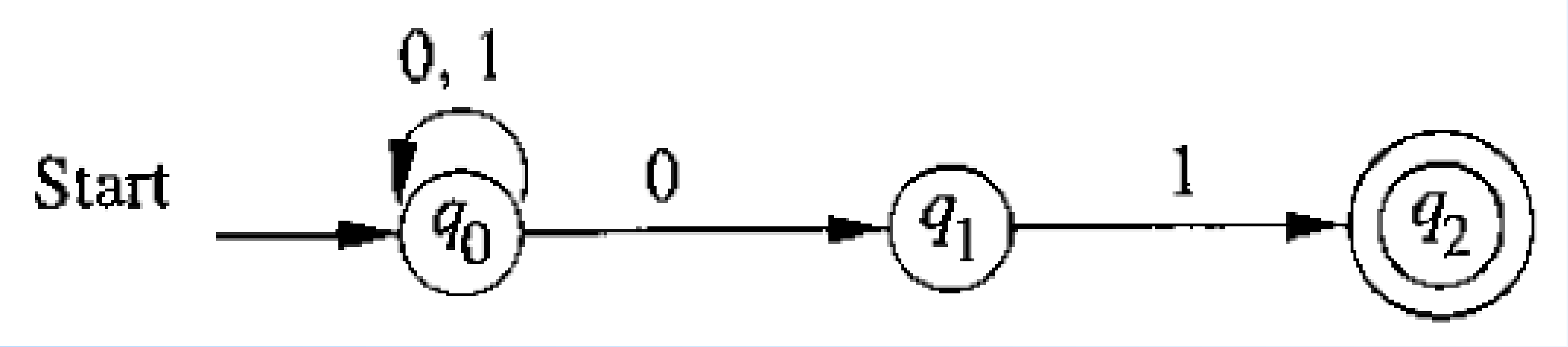
Exemplo

- Processamento da string 0110



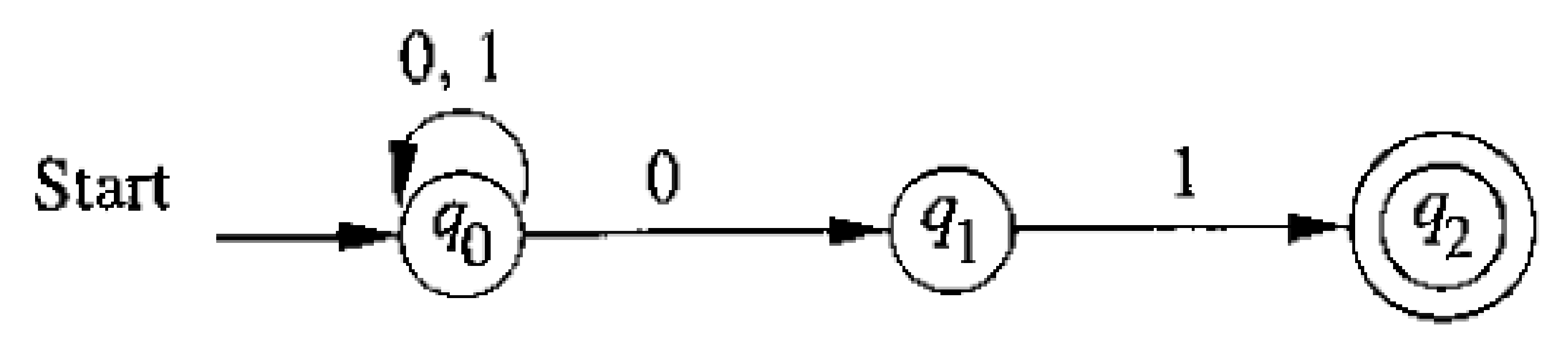
Exemplo

- Que strings são aceitas?



Exemplo

- Que strings são aceitas?
- Strings que terminam em 01



Formalização do AFN

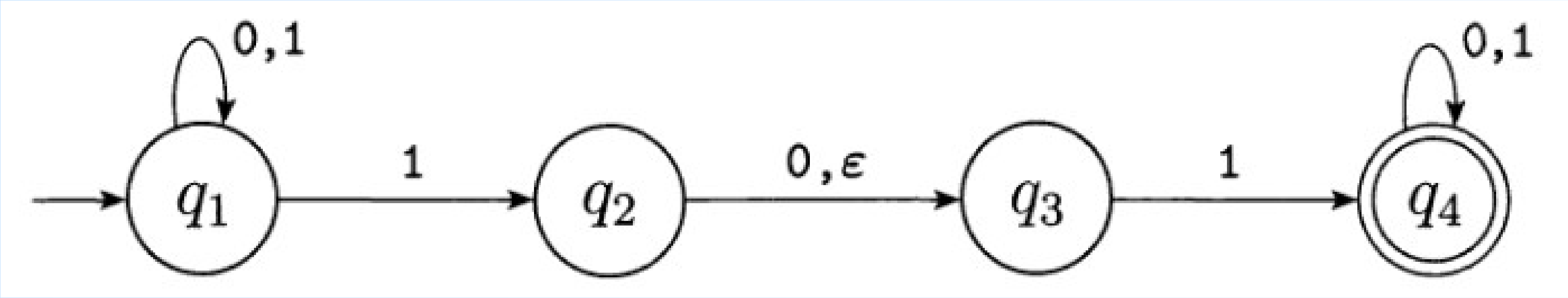
- Um AFN é uma tupla (Q,Σ,δ,q_0,F)
- Um conjunto finito de estados Q
- Um alfabeto de entrada Σ
- Uma função de transição δ
  - $\delta:Q\text{x}(\Sigma \cup \{\varepsilon\}) \rightarrow P(Q)$
- Um estado inicial q_0 em Q
- Um conjunto de estado finais $F \subseteq Q$

Função de Transição AFN

- $\delta(q, a)$ é um conjunto de estados
  - Todos os estados alcançáveis a partir de q com o símbolo a
  - $\delta(q, a) = \{p_1, \ldots, p_k\}$
- $\delta(q, \varepsilon)$ é o conjunto de estados alcançáveis a partir de q com ε
  - $\delta(q, \varepsilon) = \{p_1, \ldots, p_k\}$

AFN

- Vamos formalizar o AFN N_1



AFN

- $N1 = (Q,\Sigma,\delta,q_1,F)$
- $Q = \{q_1,q_2,q_3,q_4\}$
- $\Sigma = \{0,1\}$
- $F = \{q_4\}$

AFN

- N1 = (Q,Σ,δ,q_1,F)
- δ =

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

Função de Transição Estendida

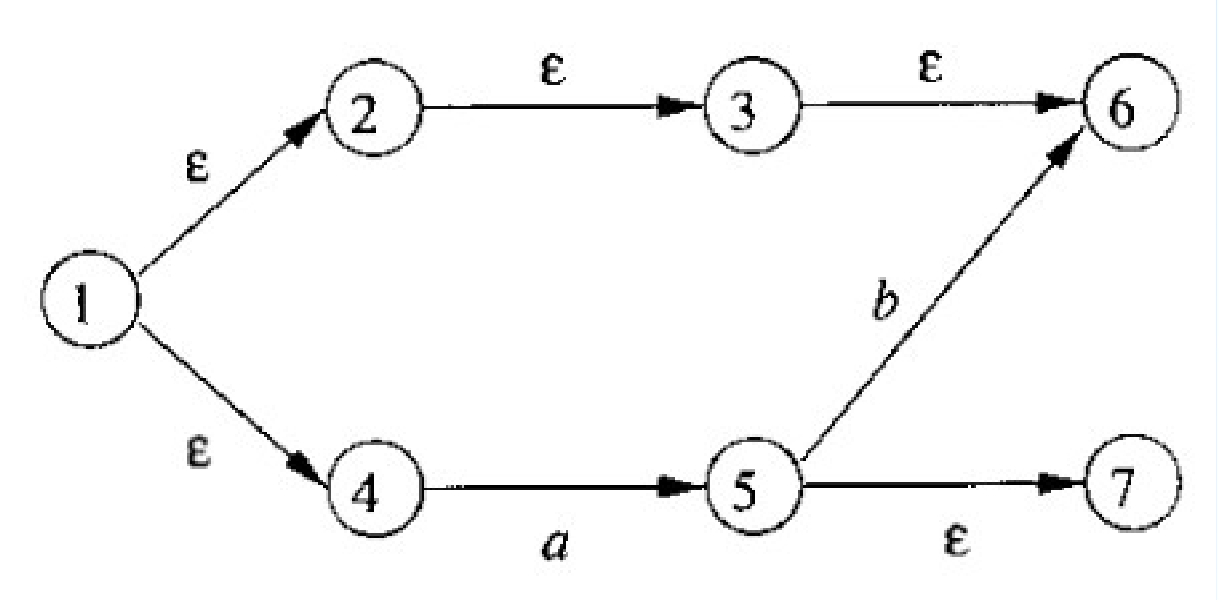
- Podemos estender a função de transição dos AFNs para lidar com strings
- É necessário definir os estados alcançáveis com transições ε
- Vamos definir o fecho das transições ε indutivamente para conjuntos de estados

Função de Transição Estendida

- Vamos definir o fecho das transições ε indutivamente para conjuntos de estados
  - Base: $R \subseteq \varepsilon\text{-closure}(R)$
  - Recursão: se $p \in \varepsilon\text{-closure}(R)$ então $\delta(p, \varepsilon) \subseteq \varepsilon\text{-closure}(R)$

Exemplo

- Computar ε-closure({1})



Função de Transição Estendida

- Seja q um estado

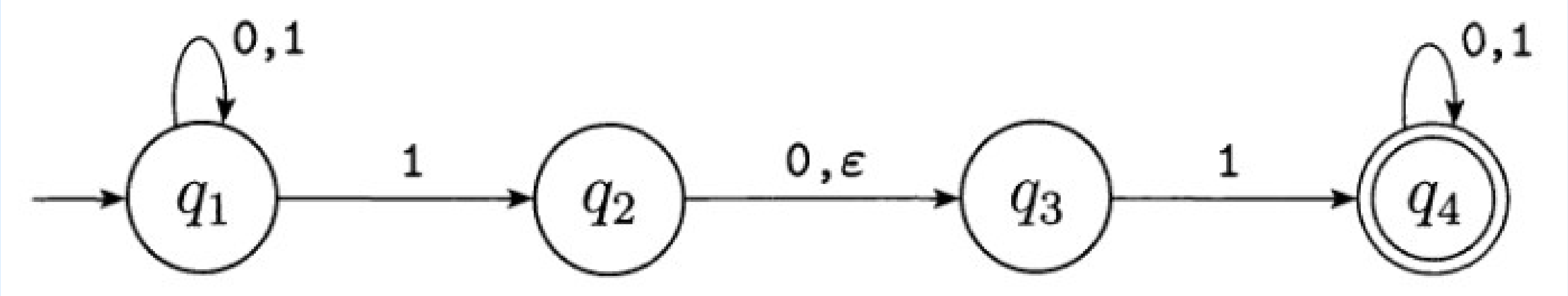
- $\delta^*(q, \varepsilon) = \varepsilon\text{-closure}(\{q\})$
- $\delta^*(q, ua) =$
  $\varepsilon\text{-closure}(\mathrm{U}_{p \in \delta^*(q,\, u)}\ \delta(p, a))$

Aceitação

- Uma string w é aceita por um AFN se $\delta^*(q_0, w)$ contém algum estado final
- $\delta^*(q_0, w) \cap F \neq \emptyset$

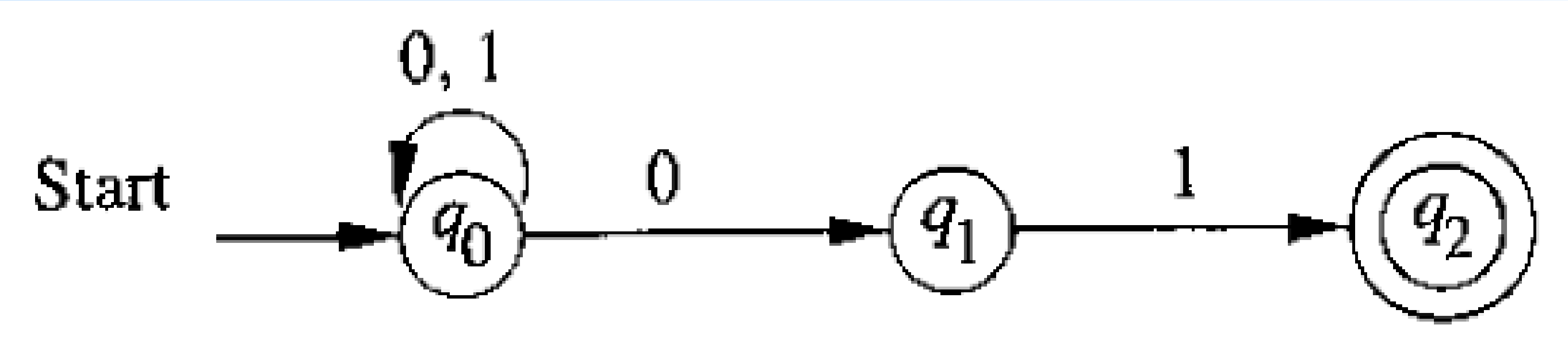
Exemplo

- Computar δ*(q0, 0110)
- N_1 aceita 0110?



Exemplo

- Computar δ*(q0, 1101)
- N_2 aceita 1101?

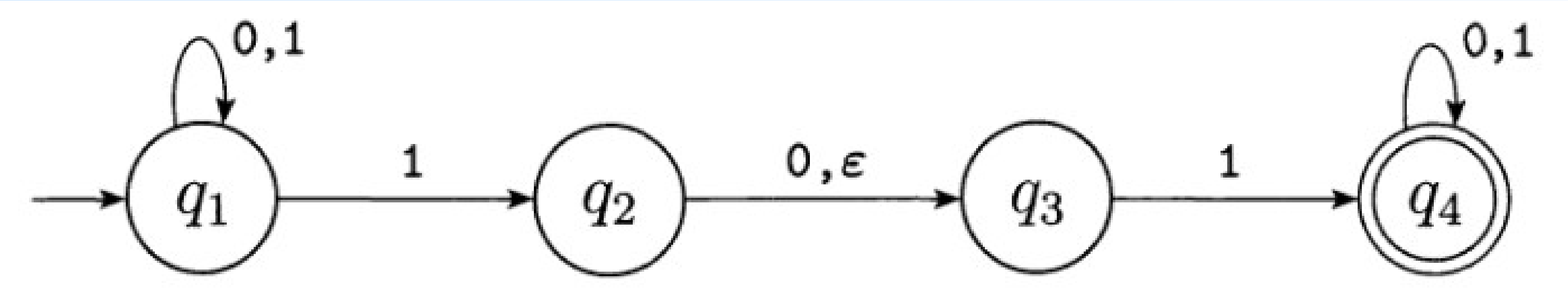


Linguagem de um AFN

- A linguagem de um AFN N é o conjunto de strings que ele aceita
- $L(N) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \varnothing\}$
- Se B é uma linguagem e N é um AFN tal que L(N) = B então dizemos que N aceita B.
  - Também usamos N reconhece B.

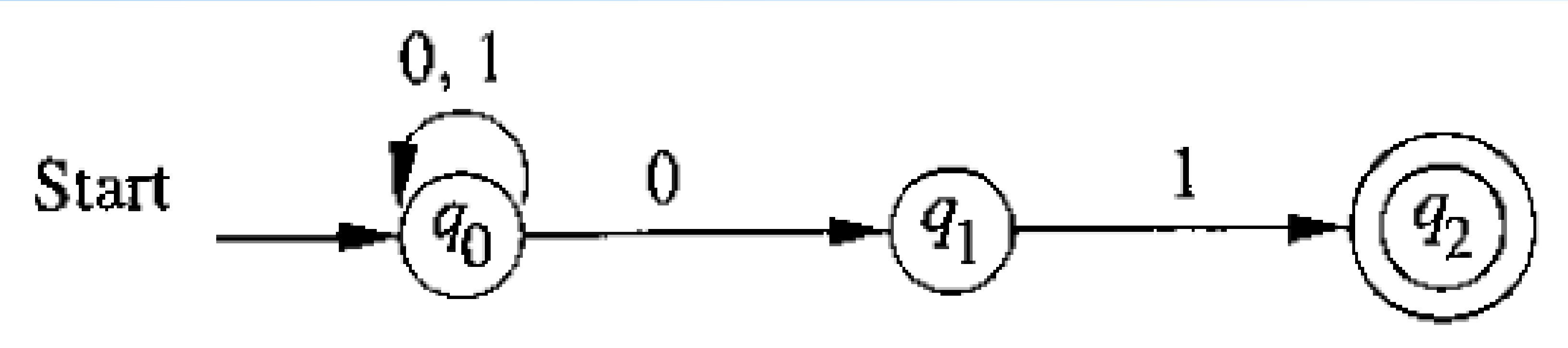
Exemplo

- $L(N_1) = \{xyz \in \{0,1\}^* \mid x \in \{0,1\}^*, z \in \{0,1\}^* \text{ e } y = 11 \text{ ou } y = 101\}$



Exemplo

- $L(N_2) = \{w \in \{0,1\}^* \mid w$ termina em 01$\}$

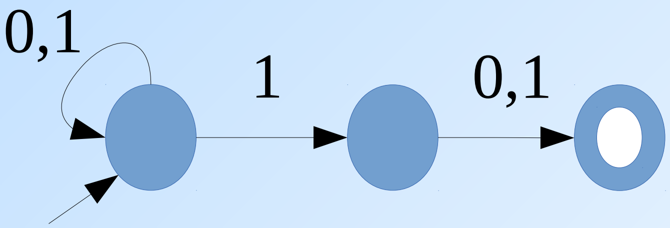


Exemplo

- Construa um AFN para reconhecer a linguagem $B_1 = \{w \in \{0,1\}^* \mid$ o penúltimo símbolo é 1$\}$.

Exemplo

- Seja N_3 o AFN definido abaixo
- N_3 reconhece B_1 pois $L(N_3) = B_1$

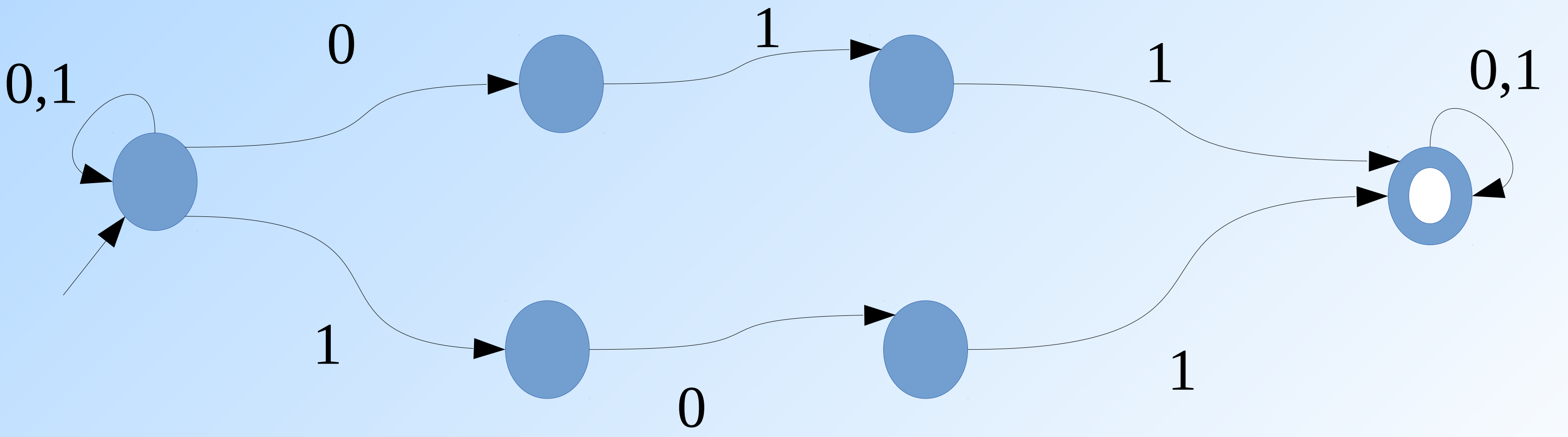


Exemplo

- Construa um AFN para aceitar a linguagem $B_2 = \{w \in \{0,1\}^* \mid w$ tem como substring 101 ou 011$\}$.

Exemplo

- Seja N_4 o AFN abaixo
- $L(N_4) = B_2$



Exercício

- Mostre que todo AFN pode ser convertido em um equivalente que possui apenas um único estado final.

Exercício

- Construa um AFN para reconhecer a linguagem $B_3 = \{w \in \{0\}^* \mid$ a quantidade de 0's é múltipla de 2 ou de 3$\}$.

Exercício

- Seja N_5 no diagrama abaixo
- $L(N_5) = B_3$