Autômatos Finitos Não Determinísticos

MCTA015-13 - Linguagens Formais e Automata

Prof. Maycon Sambinelli

m.sambinelli@ufabc.edu.br

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC

UFABC

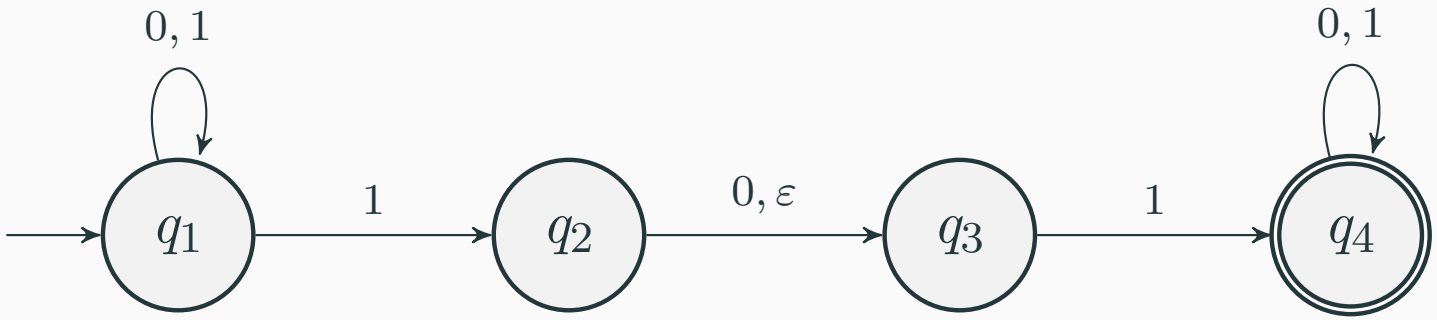
Aula de Hoje

Objetivos de Aprendizagem

- Compreender o conceito de não determinismo
- Compreender o mecanismo de funcionamento de um Autômato Finito Não Determinístico (AFN)
- Aprender a projetar AFN para uma dada linguagem
- Compreender a relação entre um AFN e AFD
- Transformar um AFN em um AFD

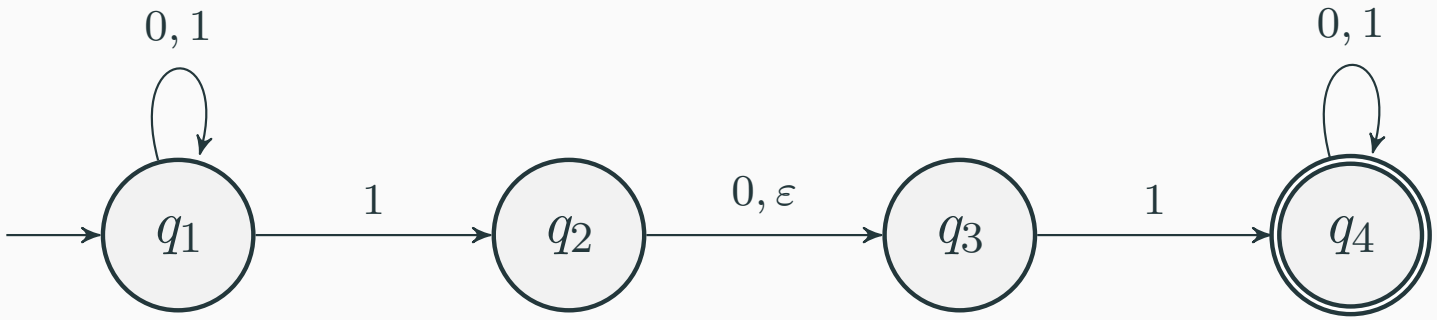
Autômato Finito Não Determinístico (AFN)

Diagrama de Estados do AFN M^*



Autômato Finito Não Determinístico (AFN)

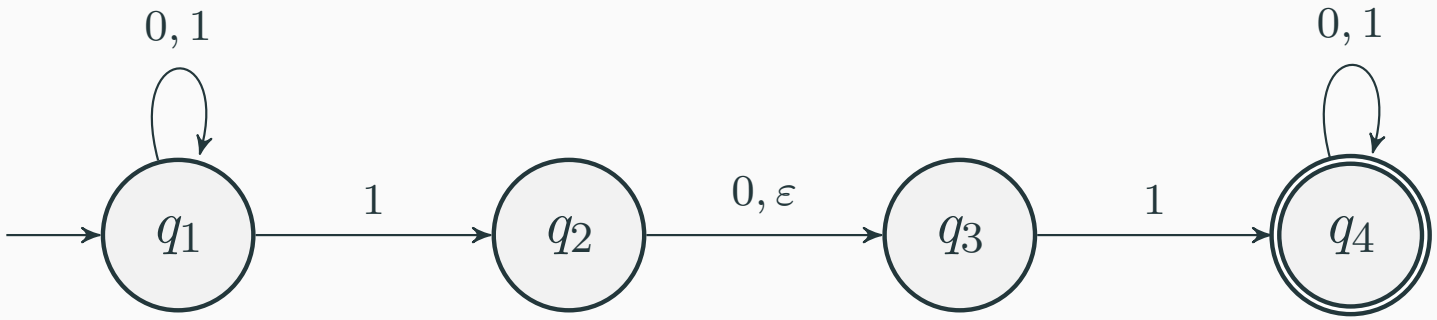
Diagrama de Estados do AFN M^*



Diferenças:

Autômato Finito Não Determinístico (AFN)

Diagrama de Estados do AFN M^*

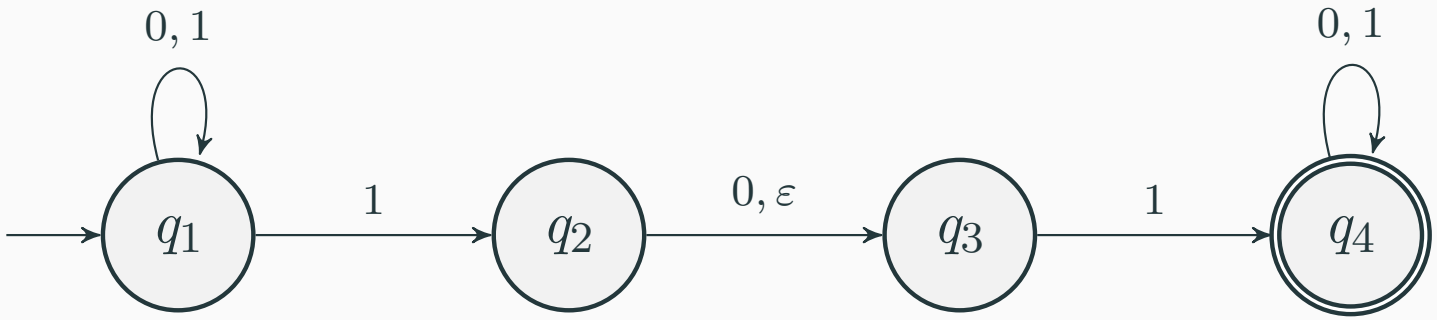


Diferenças:

- q_1 tem duas transições com o símbolo 1

Autômato Finito Não Determinístico (AFN)

Diagrama de Estados do AFN M^*

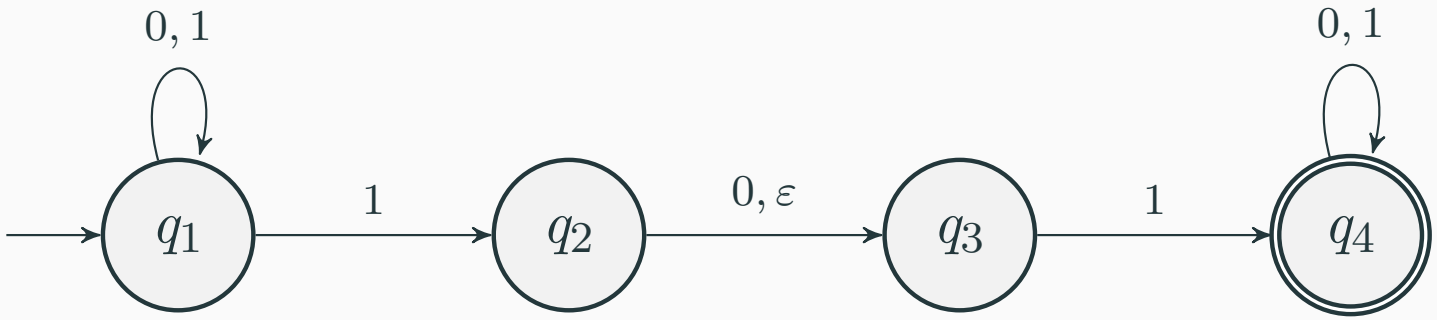


Diferenças:

- q_1 tem duas transições com o símbolo 1
- q_2 tem uma transição com ε

Autômato Finito Não Determinístico (AFN)

Diagrama de Estados do AFN M^*

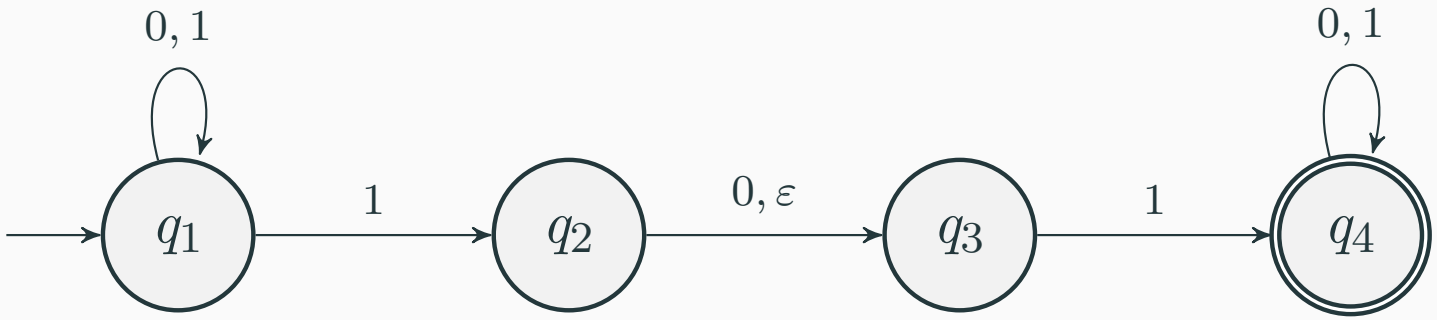


Diferenças:

- q_1 tem duas transições com o símbolo 1
- q_2 tem uma transição com ε
- q_2 não tem uma transição com 1

Autômato Finito Não Determinístico (AFN)

Diagrama de Estados do AFN M^*



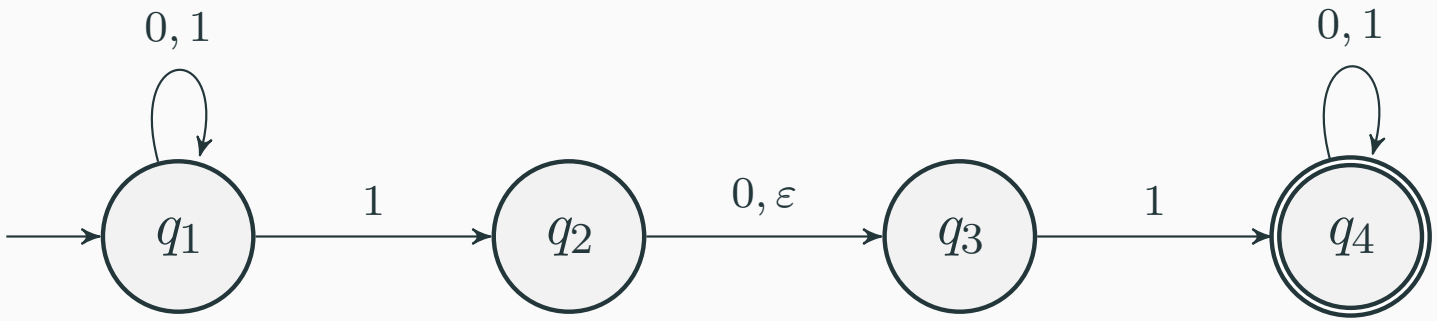
Diferenças:

- q_1 tem duas transições com o símbolo 1
- q_2 tem uma transição com ε
- q_2 não tem uma transição com 1

Não Determinismo: pode estar em mais de um estado ao mesmo tempo

Autômato Finito Não Determinístico (AFN)

Diagrama de Estados do AFN M^*



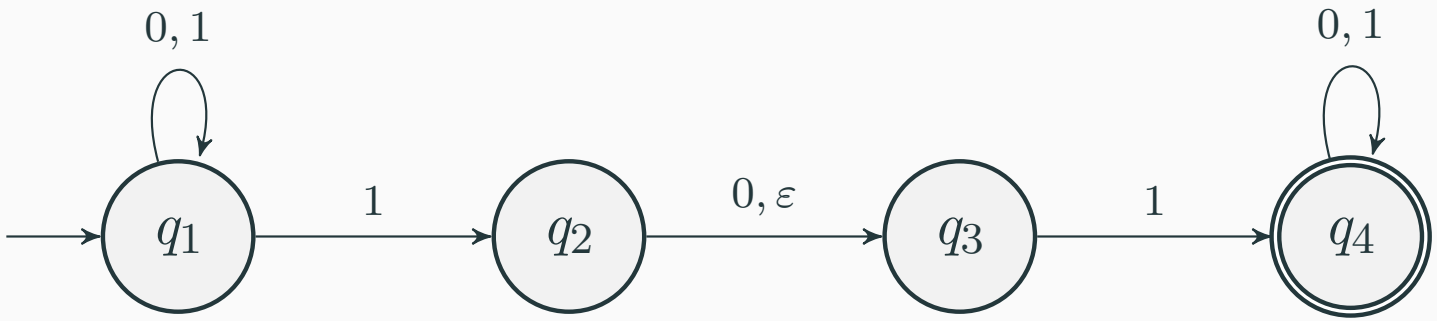
✎ Exemplo

Vamos processar as seguintes cadeias:

- 010110

Autômato Finito Não Determinístico (AFN)

Diagrama de Estados do AFN M^*



✎ Exemplo

Vamos processar as seguintes cadeias:

- 010110
- 010

Autômato Finito Não Determinístico (AFN)

Definição

Um **Autômato Finito Não Determinístico (AFN)** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

Autômato Finito Não Determinístico (AFN)

Definição

Um **Autômato Finito Não Determinístico (AFN)** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

- Q é um conjunto finito de elementos chamados **estados**;

Autômato Finito Não Determinístico (AFN)

Definição

Um **Autômato Finito Não Determinístico (AFN)** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

- Q é um conjunto finito de elementos chamados **estados**;
- Σ é um alfabeto

Autômato Finito Não Determinístico (AFN)

Definição

Um **Autômato Finito Não Determinístico (AFN)** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

- Q é um conjunto finito de elementos chamados **estados**;
- Σ é um alfabeto
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ é a **função de transição**

Autômato Finito Não Determinístico (AFN)

Definição

Um **Autômato Finito Não Determinístico (AFN)** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

- Q é um conjunto finito de elementos chamados **estados**;
- Σ é um alfabeto
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ é a **função de transição**
- $q_0 \in Q$ é o estado inicial

Autômato Finito Não Determinístico (AFN)

Definição

Um **Autômato Finito Não Determinístico (AFN)** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

- Q é um conjunto finito de elementos chamados **estados**;
- Σ é um alfabeto
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ é a **função de transição**
- $q_0 \in Q$ é o estado inicial
- $F \subseteq Q$ é o conjunto de **estados finais** (ou de **aceitação**)

Autômato Finito Não Determinístico (AFN)

Definição

Um **Autômato Finito Não Determinístico (AFN)** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

- Q é um conjunto finito de elementos chamados **estados**;
- Σ é um alfabeto
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ é a **função de transição**
- $q_0 \in Q$ é o estado inicial
- $F \subseteq Q$ é o conjunto de **estados finais** (ou de **aceitação**)

Nota

- A única diferença entre um AFD e um AFN é a sua função de transição

Autômato Finito Não Determinístico (AFN)

Definição

Um **Autômato Finito Não Determinístico (AFN)** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

- Q é um conjunto finito de elementos chamados **estados**;
- Σ é um alfabeto
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ é a **função de transição**
- $q_0 \in Q$ é o estado inicial
- $F \subseteq Q$ é o conjunto de **estados finais** (ou de **aceitação**)

Nota

- A única diferença entre um AFD e um AFN é a sua função de transição
- Todo AFD é um AFN

Sobre a função $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

- Um estado pode ter 0 ou mais transições para cada símbolo do alfabeto ou ε

Sobre a função $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

- Um estado pode ter 0 ou mais transições para cada símbolo do alfabeto ou ε
- Uma transição rotulada com ε indica que a máquina pode mudar de estado sem ler um símbolo da entrada

Sobre a função $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

- Um estado pode ter 0 ou mais transições para cada símbolo do alfabeto ou ε
- Uma transição rotulada com ε indica que a máquina pode mudar de estado sem ler um símbolo da entrada
- Após ler um símbolo, a máquina "divide-se em várias cópias de si mesma" e segue todas as possibilidades em paralelo

Sobre a função $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

- Um estado pode ter 0 ou mais transições para cada símbolo do alfabeto ou ε
- Uma transição rotulada com ε indica que a máquina pode mudar de estado sem ler um símbolo da entrada
- Após ler um símbolo, a máquina "divide-se em várias cópias de si mesma" e segue todas as possibilidades em paralelo
    - Cada cópia prossegue a execução normalmente

Sobre a função $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

- Um estado pode ter 0 ou mais transições para cada símbolo do alfabeto ou ε
- Uma transição rotulada com ε indica que a máquina pode mudar de estado sem ler um símbolo da entrada
- Após ler um símbolo, a máquina "divide-se em várias cópias de si mesma" e segue todas as possibilidades em paralelo
  - Cada cópia prossegue a execução normalmente
  - Se existirem escolhas subsequentes, a máquina divide-se novamente

Sobre a função $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

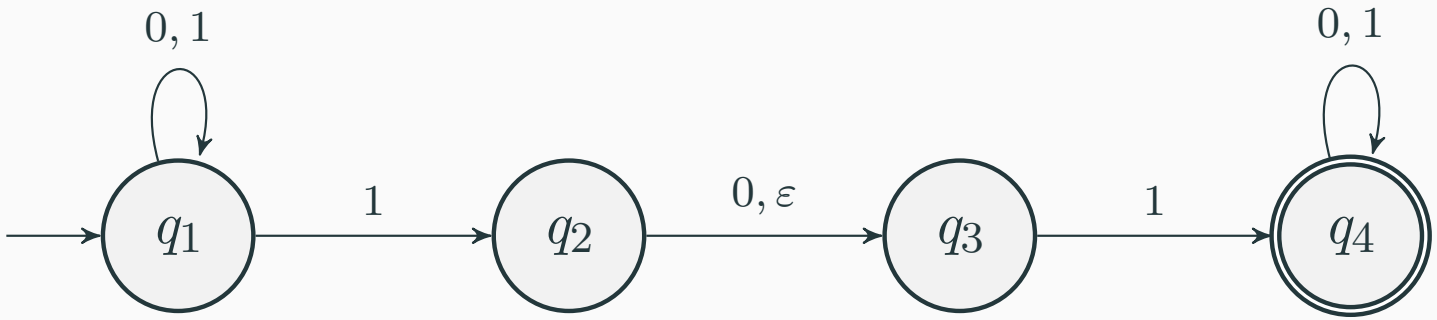
- Um estado pode ter 0 ou mais transições para cada símbolo do alfabeto ou ε
- Uma transição rotulada com ε indica que a máquina pode mudar de estado sem ler um símbolo da entrada
- Após ler um símbolo, a máquina "divide-se em várias cópias de si mesma" e segue todas as possibilidades em paralelo
  - Cada cópia prossegue a execução normalmente
  - Se existirem escolhas subsequentes, a máquina divide-se novamente
  - Se o próximo símbolo da entrada não aparece nas transições, a cópia "morre"

Sobre a função $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

- Um estado pode ter 0 ou mais transições para cada símbolo do alfabeto ou ε
- Uma transição rotulada com ε indica que a máquina pode mudar de estado sem ler um símbolo da entrada
- Após ler um símbolo, a máquina "divide-se em várias cópias de si mesma" e segue todas as possibilidades em paralelo
  - Cada cópia prossegue a execução normalmente
  - Se existirem escolhas subsequentes, a máquina divide-se novamente
  - Se o próximo símbolo da entrada não aparece nas transições, a cópia "morre"
  - Ao final do processamento da cadeia de entrada, se alguma cópia está em um estado final, então o AFN aceita a cadeia; caso contrário, rejeita

Exemplo: Definição Formal de Um AFN

Diagrama de Estados do AFN M^*



$M^* = (Q, \Sigma, \delta, q_1, F)$, onde $Q = \{q_1, q_2, q_3, q_4\}$, $\Sigma = \{0, 1\}$, $F = \{q_4\}$ e δ é definido como

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	$\{q_4\}$	$\{q_4\}$	$\emptyset$

Definição formal de Aceita

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN e seja ω uma cadeia sobre Σ. Dizemos que N **aceita** ω se podemos escrever $\omega = \alpha_1 \alpha_2 \cdots \alpha_m$, onde $\alpha_i \in \Sigma \cup \{\varepsilon\}$ para $1 \leq i \leq m$, e existe uma sequência de estados $(r_0, r_1, \ldots, r_m)$

- $r_0 = q_0$
- $r_{i+1} \in \delta(r_i, \alpha_{i+1}) \quad \forall i = 0, \ldots, m-1$
- $r_m \in F$

Linguagem reconhecida por um AFN

Se X é o conjunto de todas as cadeias que um AFN N aceita, então dizemos que

- X é a **linguagem** de N

Linguagem reconhecida por um AFN

Se X é o conjunto de todas as cadeias que um AFN N aceita, então dizemos que

- X é a **linguagem** de N
- $L(N) = X$

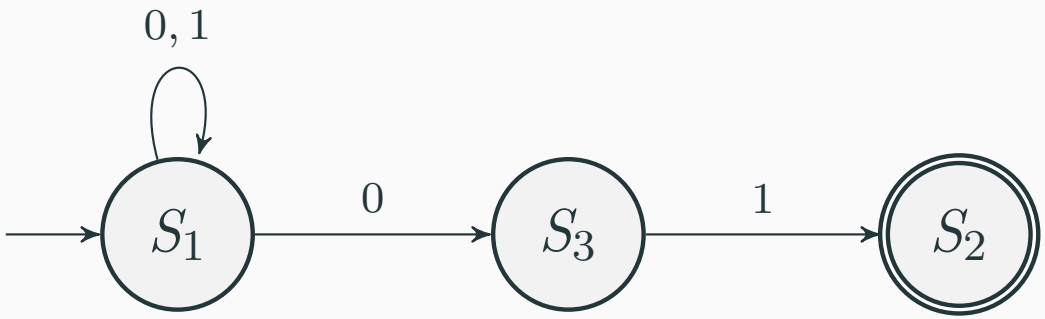
Linguagem reconhecida por um AFN

Se X é o conjunto de todas as cadeias que um AFN N aceita, então dizemos que

- X é a **linguagem** de N
- $L(N) = X$
- N **reconhece** X

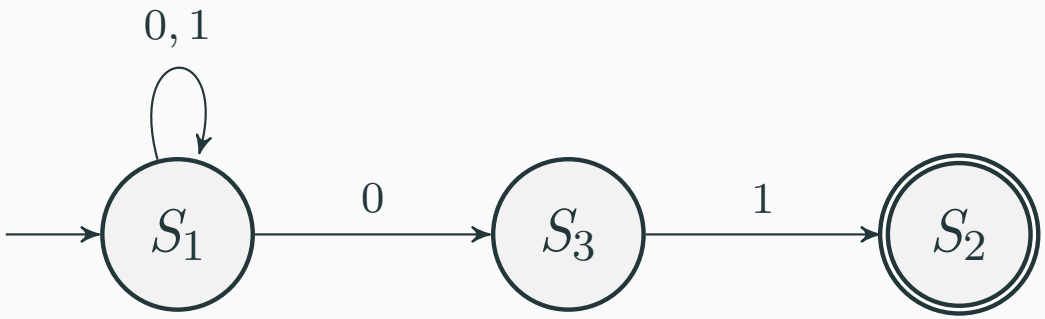
Exemplos de AFN

Diagrama de Estados do AFN M_1^*



Exemplos de AFN

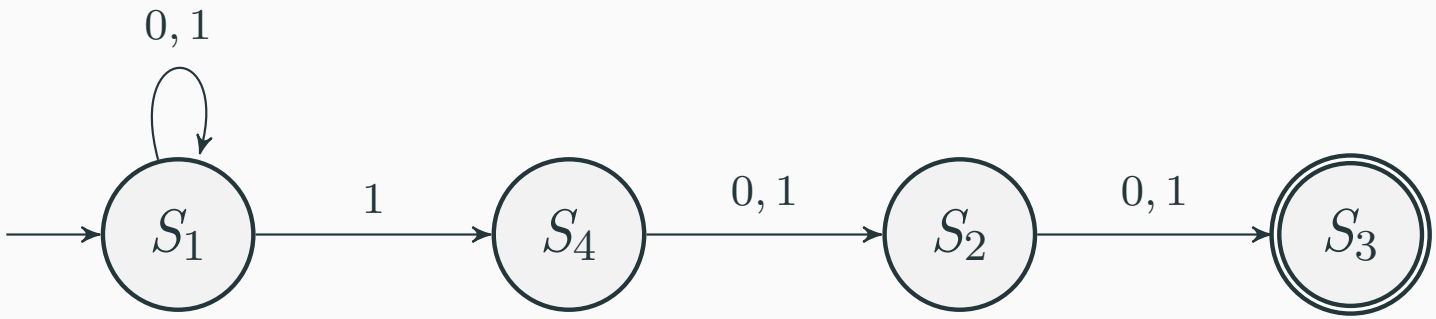
Diagrama de Estados do AFN M_1^*



$L(M_1^*) = \{\omega \in \{0,1\}^* : \text{ termina com } 01\}$

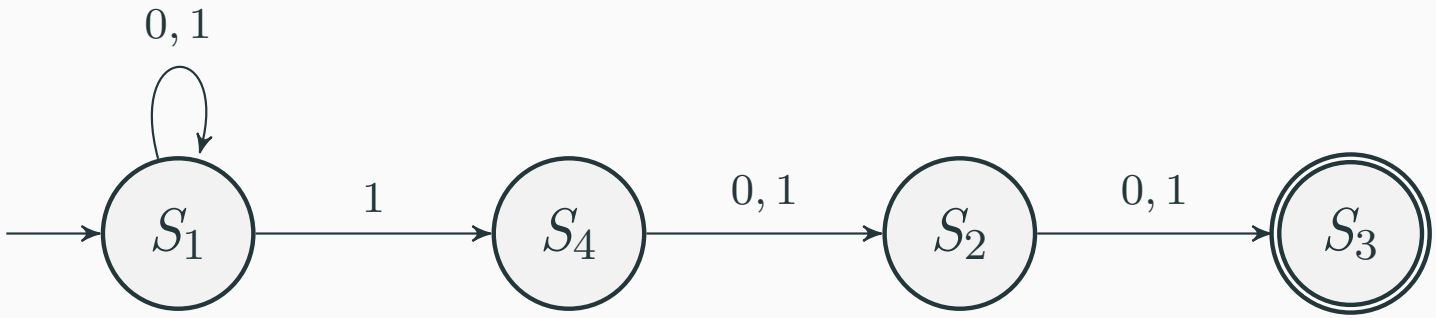
Exemplos de AFN

Diagrama de Estados do AFN N



Exemplos de AFN

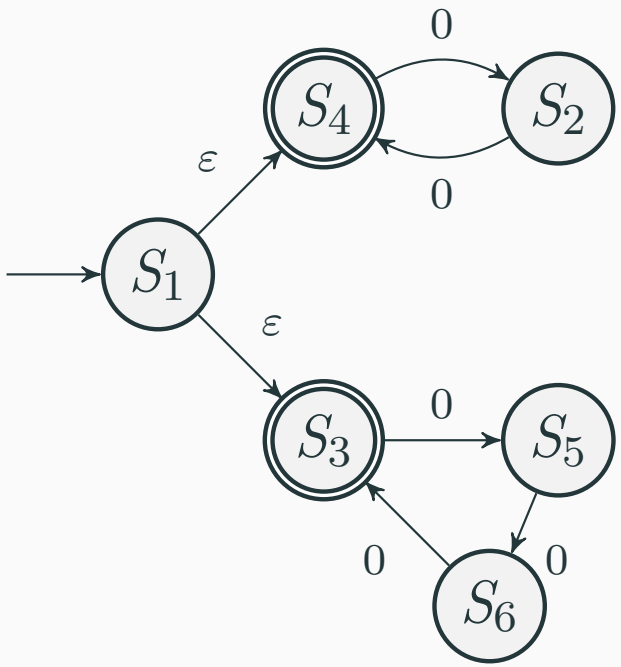
Diagrama de Estados do AFN N



$L(N) = \{\omega \in \{0,1\}^* :$ o antepenúltimo símbolo de ω é $1\}$

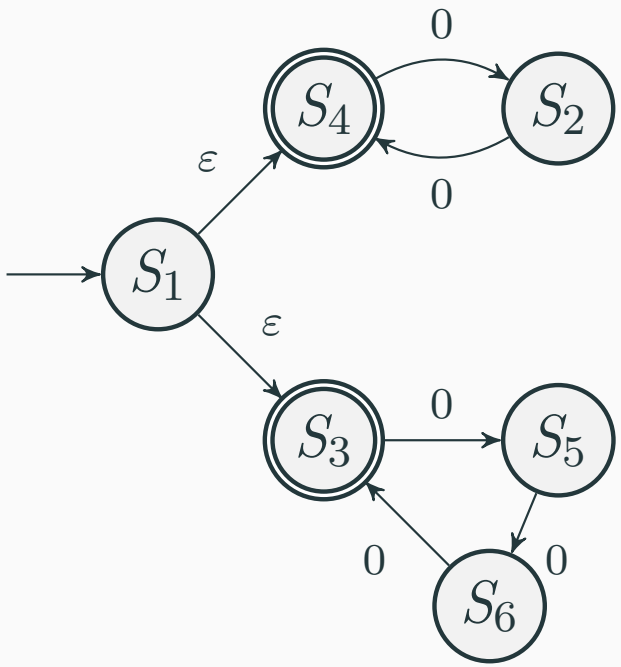
Exemplos de AFN

Diagrama de Estados do AFN N^*



Exemplos de AFN

Diagrama de Estados do AFN N^*



$L(N^*) = \{\omega \in \{0\}^* : \omega = 0^k$ e k é múltiplo de 2 ou $3\}$

Exercício

Exercício

Projete um AFN que reconhaça a linguagem:

$$X = \{\omega \in \{a, b, c\}^* : \omega \text{ termina com } abc \text{ ou } bca\}$$

AFN vs AFD

Diagrama de Estados do AFN N

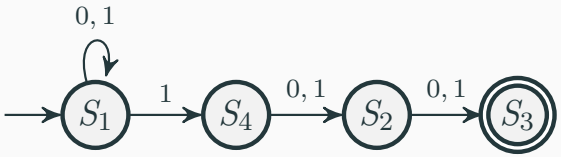
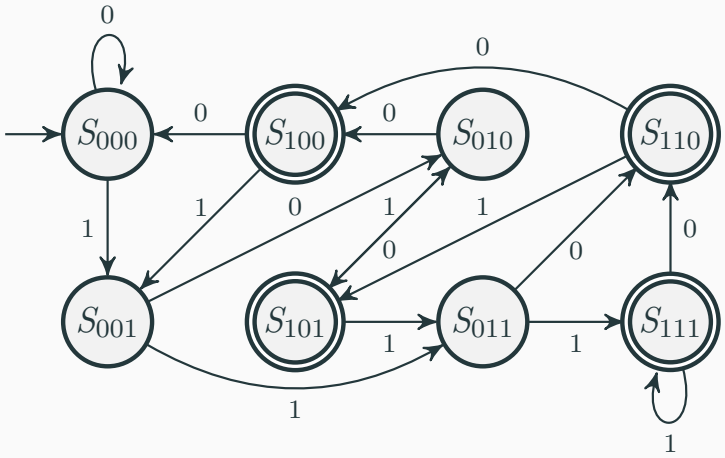


Diagrama de Estados do AFD M



$L(N) = L(M) = \{\omega \in \{0,1\}^* :$ o antepenúltimo símbolo de ω é $1\}$

Considerações

Quem é mais poderoso, AFD ou AFN?

Considerações

Quem é mais poderoso, AFD ou AFN?

- **Empate!** Ambos reconhecem o mesmo conjunto de linguagens

Equivalência entre AFD e AFN

Definição de Equivalência

Dois autômatos M e N são **equivalentes** se $L(M) = L(N)$, i.e., se ambos reconhecem a mesma linhagem

Equivalência entre AFD e AFN

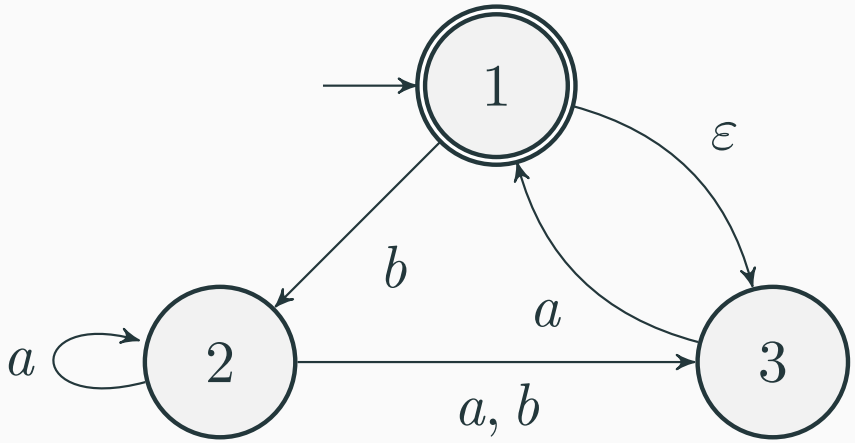
Teorema

Todo AFN tem um AFD equivalente

Convertendo um AFN em AFD

Autômato finito não determinístico:

$$N = (Q, \Sigma, \delta, q_0, F)$$



Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN
- **função de transição:** deve imitar o funcionamento do AFN

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN
- **função de transição:** deve imitar o funcionamento do AFN
  - Sejam $R \in B$ e $a \in \Sigma$, definimos

$$\varphi(R, a) = \{p \in E(\delta(r, a)) \colon r \in R\}$$

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN
- **função de transição:** deve imitar o funcionamento do AFN
  - Sejam $R \in B$ e $a \in \Sigma$, definimos

$$\varphi(R, a) = \{p \in E(\delta(r, a)) : r \in R\}$$

$E(X) = \{q \in Q$: q pode ser atingido a partir de um $p \in X$ seguindo 0 ou mais transições com o rótulo $\varepsilon\}$

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN
- **função de transição:** deve imitar o funcionamento do AFN
  - Sejam $R \in B$ e $a \in \Sigma$, definimos

$$\varphi(R, a) = \{p \in E(\delta(r, a)) : r \in R\}$$

$E(X) = \{q \in Q$: q pode ser atingido a partir de um $p \in X$ seguindo 0 ou mais transições com o rótulo $\varepsilon\}$

- **estado inicial:** deve imitar o estado em que o AFN se encontra quando nenhum símbolo foi processado

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN
- **função de transição:** deve imitar o funcionamento do AFN
  - Sejam $R \in B$ e $a \in \Sigma$, definimos

$$\varphi(R, a) = \{p \in E(\delta(r, a)) : r \in R\}$$

$E(X) = \{q \in Q$: q pode ser atingido a partir de um $p \in X$ seguindo 0 ou mais transições com o rótulo $\varepsilon\}$

- **estado inicial:** deve imitar o estado em que o AFN se encontra quando nenhum símbolo foi processado

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN
- **função de transição:** deve imitar o funcionamento do AFN
  - Sejam $R \in B$ e $a \in \Sigma$, definimos

$$\varphi(R, a) = \{p \in E(\delta(r, a)) : r \in R\}$$

$E(X) = \{q \in Q$: q pode ser atingido a partir de um $p \in X$ seguindo 0 ou mais transições com o rótulo $\varepsilon\}$

- **estado inicial:** deve imitar o estado em que o AFN se encontra quando nenhum símbolo foi processado
  - $p_0 = E(\{q_0\})$

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN
- **função de transição:** deve imitar o funcionamento do AFN
  - Sejam $R \in B$ e $a \in \Sigma$, definimos

$$\varphi(R, a) = \{p \in E(\delta(r, a)) : r \in R\}$$

$E(X) = \{q \in Q$: q pode ser atingido a partir de um $p \in X$ seguindo 0 ou mais transições com o rótulo $\varepsilon\}$
- **estado inicial:** deve imitar o estado em que o AFN se encontra quando nenhum símbolo foi processado
  - $p_0 = E(\{q_0\})$
- **estados de finais/aceitação:** quais estados imitam o comportamento de aceitação do AFN?

Convertendo um AFN em AFD (Cont.)

Construindo o AFD $M = (B, \Sigma, \varphi, p_0, A)$ tal que $L(M) = L(N)$:

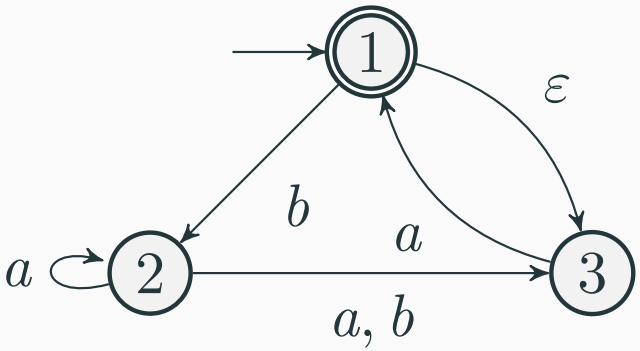
- **estados:** um estado pra cada possível estado do AFN
  - $B = \mathcal{P}(Q)$
- **alfabeto:** idêntico ao do AFN
- **função de transição:** deve imitar o funcionamento do AFN
  - Sejam $R \in B$ e $a \in \Sigma$, definimos

$$\varphi(R, a) = \{p \in E(\delta(r, a)) \colon r \in R\}$$

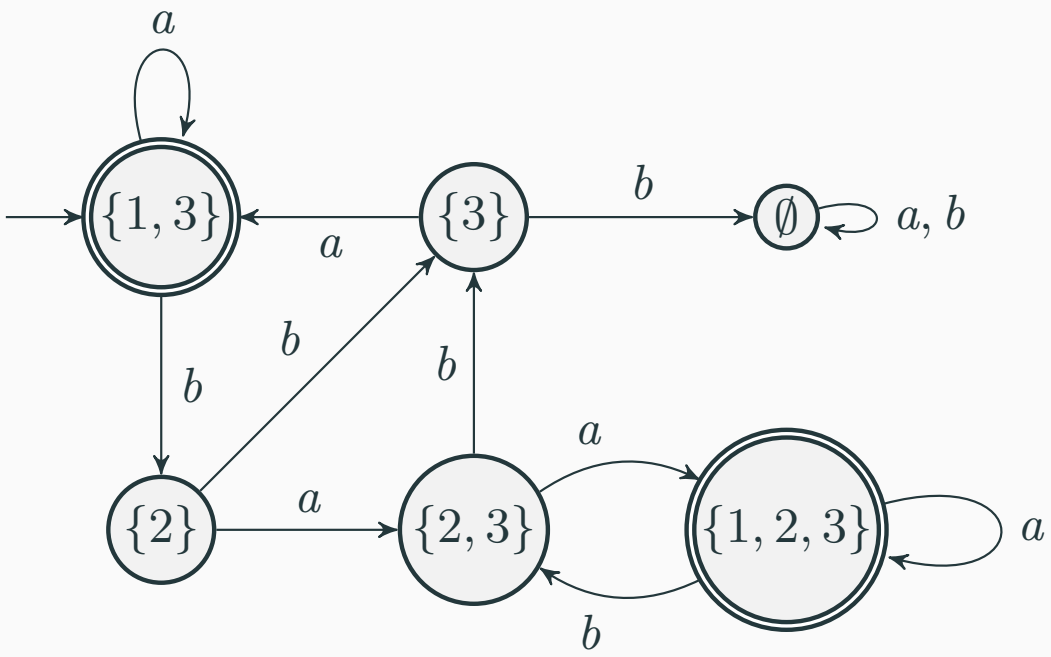
    $E(X) = \{q \in Q \colon q$ pode ser atingido a partir de um $p \in X$ seguindo 0 ou mais transições com o rótulo $\varepsilon\}$
- **estado inicial:** deve imitar o estado em que o AFN se encontra quando nenhum símbolo foi processado
  - $p_0 = E(\{q_0\})$
- **estados de finais/aceitação:** quais estados imitam o comportamento de aceitação do AFN?
  - $A = \{S \in B \colon S \cap F \neq \emptyset\}$

Convertendo um AFN em AFD

$N = (Q, \Sigma, \delta, q_0, F)$

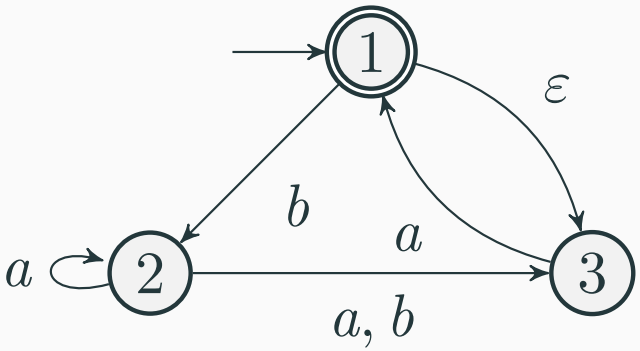


$M = (B, \Sigma, \varphi, p_0, A)$

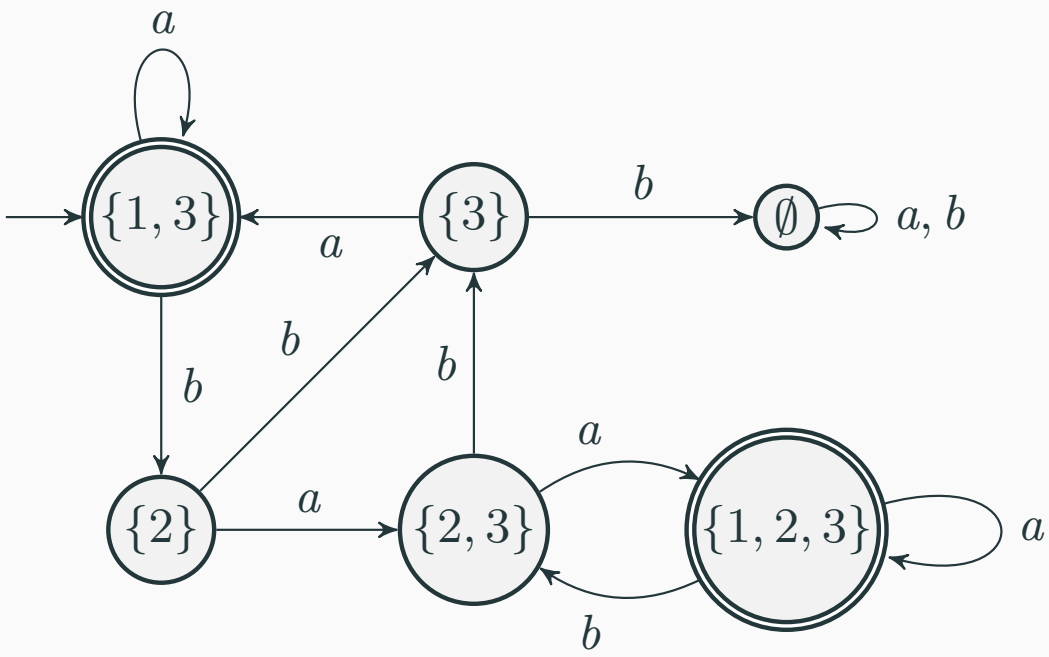


Convertendo um AFN em AFD

$N = (Q, \Sigma, \delta, q_0, F)$



$M = (B, \Sigma, \varphi, p_0, A)$



✎ Exemplo

Vamos processar as cadeias **baaa**, **aabbb**

Equivalência entre AFD e AFN

Corolário

Uma linguagem é **regular** se e somente se algum AFN a reconhece

Simulando um AFN

Simulando um AFN

```
# Definição da função de transição do AFD M^*
delta = {('q1', '0' ): {'q1'},
         ('q1', '1'): {'q1', 'q2'},
         ('q2', '0'): {'q3'},
         ('q2', 'ε'): {'q3'},
         ('q3', '1'): {'q4'},
         ('q4', '0'): {'q4'},
         ('q4', '1'): {'q4'}}

afn(delta, 'q1', {'q4'}, "001100") # -> True
```

Simulando um AFN

```
def afn(delta, q0, F, w):
    states = E({q0}, delta)
    for a in w:
        new_states = set()
        for q in states:
            if (q, a) in delta:
                new_states.update(E(delta[(q, a)],
                ↪  delta))
        states = new_states

    return len(states.intersection(F)) != 0
```

Simulando um AFN

```
def E(states, delta):
    S = set(states)
    nvisited = list(states)  # non-visited states
    while len(nvisited) > 0:
        q = nvisited.pop()
        if (q, 'ε') in delta:
            diff = delta[(q, 'ε')].difference(S)
            if len(diff) > 0:
                S.update(diff)
                nvisited.extend(diff)
    return S
```