Autômatos Finitos Não Determinísticos

MCTA015-13 - Linguagens Formais e Autômata

Profa. Carla Negri Lintzmayer

carla.negri@ufabc.edu.br

www.professor.ufabc.edu.br/∼carla.negri

Centro de Matemática, Computação e Cognição – Universidade Federal do ABC

UFABC

Autômato Finito Não Determinístico (AFN)

Descrição formal

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;
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$ 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)

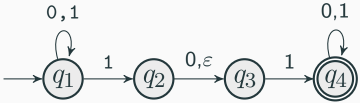
- A única diferença entre um AFD e um AFN é a sua função de transição:
  - um estado pode ter zero ou mais transições para cada símbolo do alfabeto ou ε saindo dele;
  - uma transição rotulada com ε indica que a máquina pode mudar de estado sem ler um símbolo da entrada;
  - após ler um símbolo, a máquina divide-se em várias cópias de sim mesma e segue todas as possibilidades em paralelo;
    - cada cópia continua como antes;
    - se existirem escolhas subsequentes, a máquina divide-se novamente;
    - se o próximo símbolo da entrada não aparece nas transições, a cópia morre;
    - se alguma das cópias está em um estado final ao fim da leitura da entrada, o AFN aceita a cadeia.
- Observe que todo AFD é um AFN!

Autômato Finito Não Determinístico (AFN)

Considere o seguinte AFN $N_1 = (Q, \Sigma, \delta, q_1, F)$, onde $Q = \{q_1, q_2, q_3, q_4\}$, $\Sigma = \{0, 1\}$, $F = \{q_4\}$ e δ é definida 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	$\{q_4\}$	$\{q_4\}$	$\emptyset$

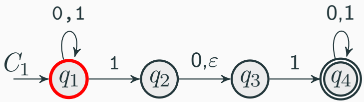
Seu diagrama de estados é:



Processando $\omega = 01101$ no AFN N_1

$$\underset{\uparrow}{}01101$$

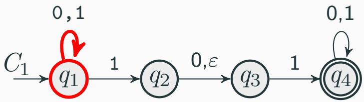
Antes de qualquer símbolo ser lido, nenhuma transição ε sai de nenhum estado, portanto apenas q_1 fica ativo e há uma cópia da máquina:



Processando $\omega = 01101$ no AFN N_1

$$\underset{\uparrow}{0}1101$$

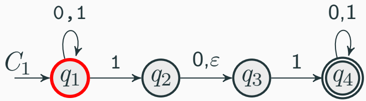
Quando o primeiro símbolo, 0, é lido, C_1 segue a transição para q_1:



Processando $\omega = 01101$ no AFN N_1

$$0\underset{\uparrow}{\,}1101$$

Logo após a leitura do primeiro símbolo, não há transições ε a partir de nenhum estado ativo, então nada acontece:

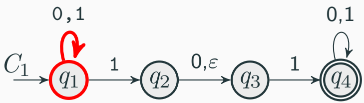


Processando $\omega = 01101$ no AFN N_1

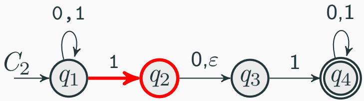
$$\underset{\uparrow}{0}1101$$

Quando o segundo símbolo, 1, é lido, a máquina é duplicada:

Uma cópia segue a transição que leva a q_1:



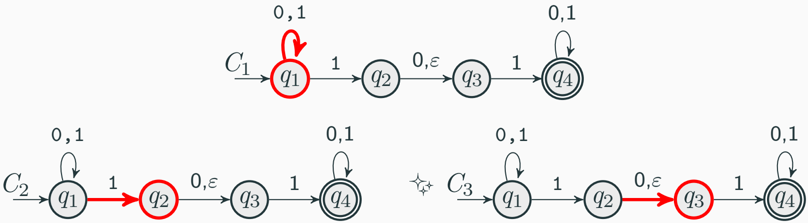
A outra cópia segue a transição que leva a q_2:



Processando $\omega = 01101$ no AFN N_1

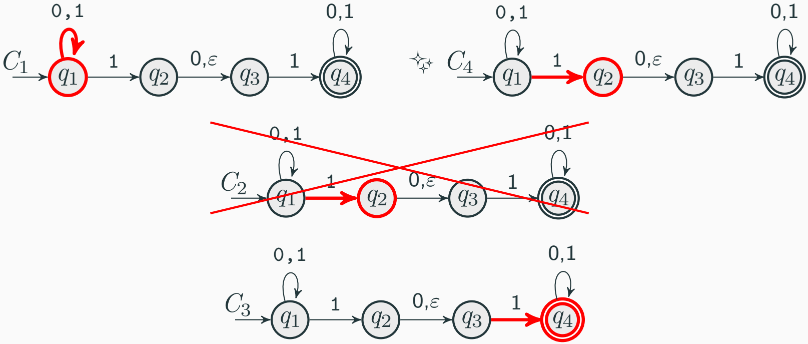
$$01\underset{\uparrow}{\,}101$$

Logo após a leitura do segundo símbolo, 1, existe um estado ativo com transição ε saindo dele, por isso C_2 é duplicada para seguir tal transição:



$$\underset{\uparrow}{01101}$$

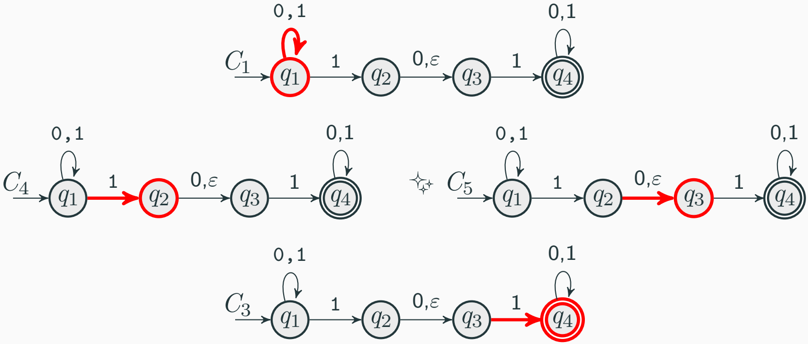
Quando o terceiro símbolo é lido, 1, cada cópia segue sua devida transição, se houver, podendo ser duplicada, ou então morre:



Processando $\omega = 01101$ no AFN N_1

$$011\underset{\uparrow}{\ }01$$

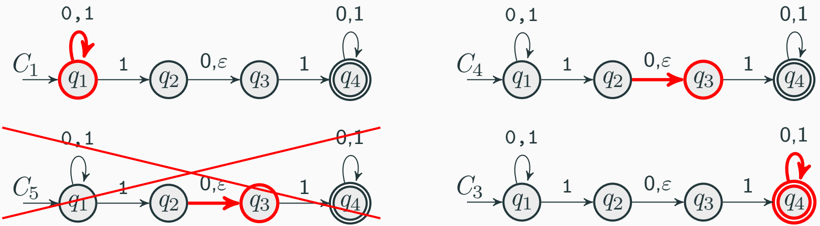
Logo após a leitura do terceiro símbolo, 1, existe um estado ativo com transição ε saindo dele, por isso C_4 é duplicada para seguir tal transição:



Processando $\omega = 01101$ no AFN N_1

$$\underset{\uparrow}{01101}$$

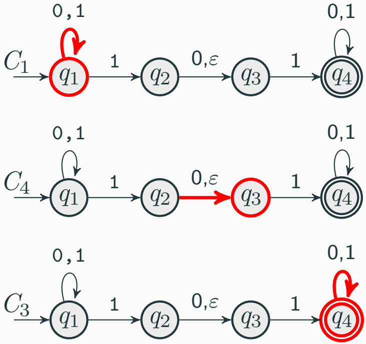
Quando o quarto símbolo é lido, 0, cada cópia segue sua devida transição, se houver, podendo ser duplicada, ou então morre:



Processando $\omega = 01101$ no AFN N_1

$$0110\underset{\uparrow}{\ }1$$

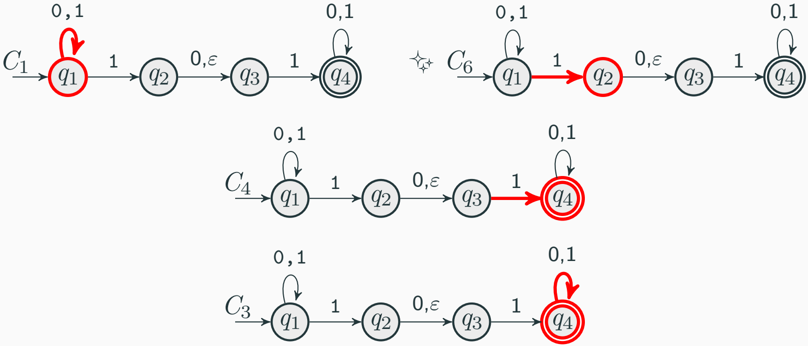
Logo após a leitura do quarto símbolo, 0, nenhum dos estados ativos tem transição ε saindo dele, por isso nada acontece:



Processando $\omega = 01101$ no AFN N_1

$$\underset{\uparrow}{0110\,1}$$

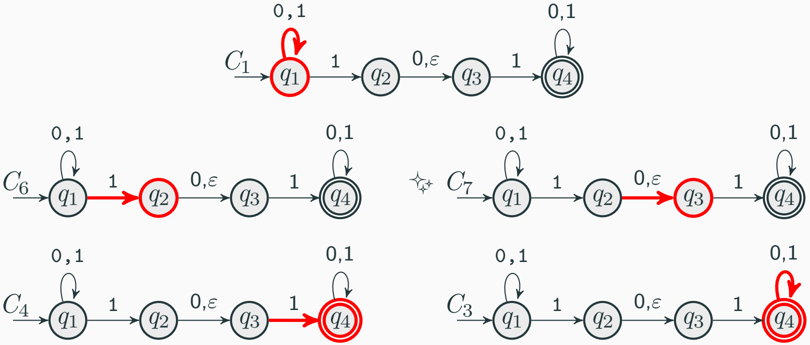
Quando o último símbolo é lido, 1, cada cópia segue sua devida transição, se houver, podendo ser duplicada, ou então morre:



Processando $\omega = 01101$ no AFN N_1

$$\underset{\uparrow}{01101}$$

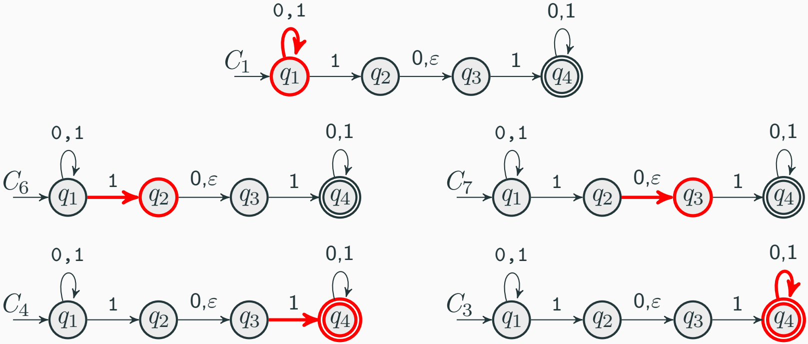
Logo após a leitura do último símbolo, 1, existe um estado ativo com transição ε saindo dele, por isso C_6 é duplicada para seguir tal transição:



Processando $\omega = 01101$ no AFN N_1

$$\underset{\uparrow}{01101}$$

Não há mais símbolos na entrada, e existe pelo menos uma cópia com estado ativo que é final. Assim, aceitamos a cadeia 01101.



Processando $\omega = 01101$ no AFN N_1

Os passos anteriores podem ser resumidos pela seguinte tabela:

estados ativos	símbolo lido
q_1	
q_1	$\_0\ 1\ 1\ 0\ 1$
q_1	$\underline{0}\ 1\ 1\ 0\ 1$
q_1 q_2	$0\_1\ 1\ 0\ 1$
q_1 q_2 q_3	$0\ \underline{1}\ 1\ 0\ 1$
q_1 q_2 ☠ q_4	$0\ 1\_1\ 0\ 1$
q_1 q_2 q_3 q_4	$0\ 1\ \underline{1}\ 0\ 1$
q_1 q_3 ☠ q_4	$0\ 1\ 1\_0\ 1$
q_1 q_3 q_4	$0\ 1\ 1\ \underline{0}\ 1$
q_1 q_2 q_4 q_4	$0\ 1\ 1\ 0\_1$
q_1 q_2 q_4 q_4	$0\ 1\ 1\ 0\ \underline{1}$
q_1 q_2 q_3 q_4 q_4	$0\ 1\ 1\ 0\ 1\_$

Processando $\omega = 01101$ no AFN N_1

Observações:

- O processamento da cadeia 01101 foi simulado como se ela fosse a cadeia $\varepsilon 0 \varepsilon 1 \varepsilon 1 \varepsilon 0 \varepsilon 1 \varepsilon$.
- ε não é de fato lido pela máquina.
- A simulação da leitura de um ε só vai disparar alguma ação caso haja algum estado ativo com uma transição ε saindo dele.

Definição formal de computação

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 $(r_0, r_1, \ldots, r_m)$ tais que

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

Linguagem reconhecida por um AFN

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

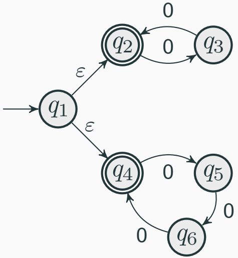
- X é a **linguagem** de N
- $L(N) = X$
- N **reconhece** X

Outro exemplo

Considere o seguinte AFN $N_2 = (Q, \Sigma, \delta, q_1, F)$, onde
$Q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$, $\Sigma = \{0, 1\}$, $F = \{q_2, q_4\}$ e δ é definida como

δ	0	1	ε
q_1	$\emptyset$	$\emptyset$	$\{q_2, q_4\}$
q_2	$\{q_3\}$	$\emptyset$	$\emptyset$
q_3	$\{q_2\}$	$\emptyset$	$\emptyset$
q_4	$\{q_5\}$	$\emptyset$	$\emptyset$
q_5	$\{q_6\}$	$\emptyset$	$\emptyset$
q_6	$\{q_4\}$	$\emptyset$	$\emptyset$

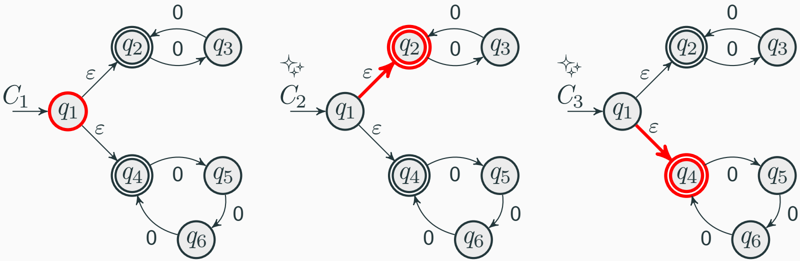
Seu diagrama de estados é:



Processando $\omega = 11$ no AFN N_2

$$\overset{}{\uparrow}11$$

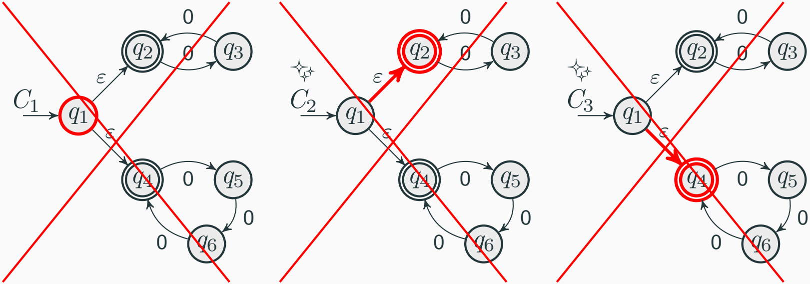
Antes de qualquer símbolo ser lido, existem transições ε que saem de estados ativos (no caso, apenas o estado inicial está ativo no início), portanto a máquina é duplicada:



Processando $\omega = 11$ no AFN N_2

$$\underset{\uparrow}{1}1$$

Quando o primeiro símbolo é lido, 1, cada cópia segue sua devida transição, se houver, podendo ser duplicada, ou então morre:



Processando $\omega = 11$ no AFN N_2

$$\underset{\uparrow}{1\ 1}$$

Logo após a leitura do primeiro símbolo, 1, nenhum dos estados ativos tem transição ε saindo dele, por isso nada acontece:

(não há mais cópias)

Processando $\omega = 11$ no AFN N_2

$$\begin{array}{c} 11 \\ \uparrow \end{array}$$

Quando o segundo símbolo é lido, 1, cada cópia segue sua devida transição, se houver, podendo ser duplicada, ou então morre:

(não há mais cópias)

Processando $\omega = 11$ no AFN N_2

$$\underset{\uparrow}{11}$$

Logo após a leitura do último símbolo, 1, nenhum estado ativo tem transição ε saindo dele, por isso nada acontece:

(não há mais cópias)

Processando $\omega = 11$ no AFN N_2

$$\underset{\uparrow}{11}$$

Não há mais símbolos na entrada, e nenhuma cópia está em estado ativo que é final. Assim, rejeitamos a cadeia 11.

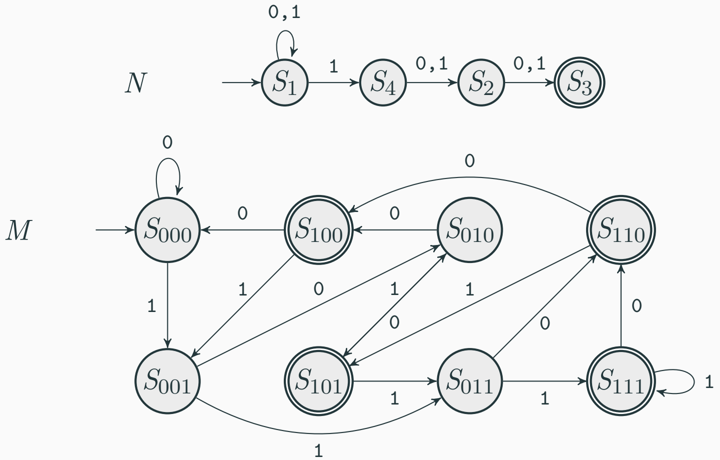
(não há mais cópias)

Processando $\omega = 11$ no AFN N_2

Os passos anteriores podem ser resumidos pela seguinte tabela:

estados ativos	símbolo lido
q_1	
q_1 q_2 q_4	$\_1\ 1$
☠ ☠ ☠	$\underline{1}\ 1$
	1_1
	$1\ \underline{1}$
	$1\ 1\_$

AFN vs. AFD



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

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 linguagem.

Equivalência entre AFD e AFN

Teorema

Todo AFN tem um AFD equivalente.

Demonstração.

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN.

Equivalência entre AFD e AFN

Teorema

Todo AFN tem um AFD equivalente.

Demonstração.

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN. Precisamos mostrar que existe um AFD que reconhece $L(N)$.

Equivalência entre AFD e AFN

Teorema

Todo AFN tem um AFD equivalente.

Demonstração.

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN. Precisamos mostrar que existe um AFD que reconhece $L(N)$.
Construa $M = (B, \Sigma, \varphi, p_0, T)$, que reconhece $L(N)$, da seguinte forma:

- $B = \mathcal{P}(Q)$
- Para $R \in B$ e $a \in \Sigma$, $\varphi(R, a) = \bigcup_{r \in R} E(\delta(r, a))$, onde
  - $E(X) =$ estados que podem ser atingidos a partir dos estados em X por transições ε, incluindo os próprios estados de X
- $p_0 = E(\{q_0\})$
- $T = \{R \in B \mid R \text{ contém um estado de } F\}$

□

Linguagens regulares

Corolário

Uma linguagem é regular se e somente se algum autômato finito não determinístico a reconhece.

Linguagens regulares

Corolário

Uma linguagem é regular se e somente se algum autômato finito não determinístico a reconhece.