Autômatos finitos

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

Determinismo

Autômatos finitos determinísticos

Um autômato finito determinístico (AFD) é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$ em que

- Q é um conjunto finito de estados
- Σ é um alfabeto
- $\delta : Q \times \Sigma \to Q$ é uma função de transição
- $q_0 \in Q$ é o estado inicial
- $F \subseteq Q$ é o conjunto de estados finais

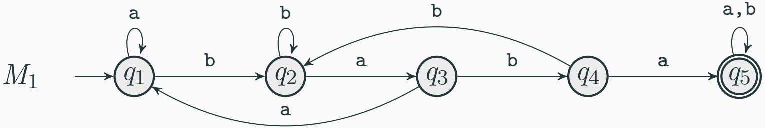
Autômatos finitos determinísticos - Exemplo

$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ com $Q_1 = \{q_1, q_2, q_3, q_4, q_5\}$, $\Sigma_1 = \{\mathtt{a}, \mathtt{b}\}$, $F_1 = \{q_5\}$ e δ_1 dada por

δ_1	a	b
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_1	q_4
q_4	q_5	q_2
q_5	q_5	q_5

é um AFD.

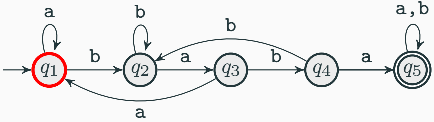
M_1 também pode ser definido por um diagrama de estados:



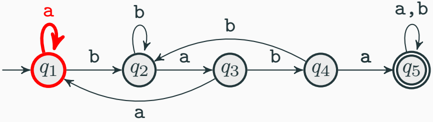
Autômatos finitos determinísticos - Computação

Ideia: o estado inicial fica ativo; uma transição com rótulo x pode ser seguida a partir de um estado ativo sempre que x for lido na entrada; o estado atingido pela transição que foi seguida fica ativo.

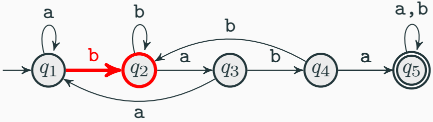
A todo momento, existe uma (e apenas uma) possibilidade de transição a ser seguida.



Leu a:



Leu b:



Autômatos finitos determinísticos

Dado um AFD $M = (Q, \Sigma, \delta, q_0, F)$, definimos a função de transição estendida de M, $\hat{\delta} : Q \times \Sigma^* \to Q$ como

$$\hat{\delta}(q, \omega) = \begin{cases} q & \text{se } \omega = \varepsilon \\ \delta(\hat{\delta}(q, \alpha), x) & \text{se } \omega = \alpha x \text{ e } x \in \Sigma \end{cases}$$

Autômatos finitos determinísticos

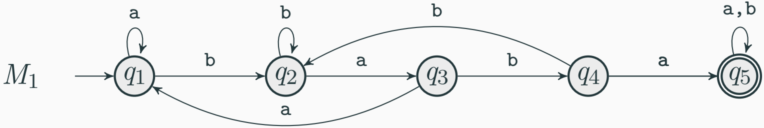
Dado um AFD $M = (Q, \Sigma, \delta, q_0, F)$, definimos a função de transição estendida de M, $\hat{\delta} : Q \times \Sigma^* \to Q$ como

$$\hat{\delta}(q, \omega) = \begin{cases} q & \text{se } \omega = \varepsilon \\ \delta(\hat{\delta}(q, \alpha), x) & \text{se } \omega = \alpha x \text{ e } x \in \Sigma \end{cases}$$

Ou seja, $\hat{\delta}(q, \omega)$ é o estado ativo em M após computar toda uma cadeia ω a partir do estado q.

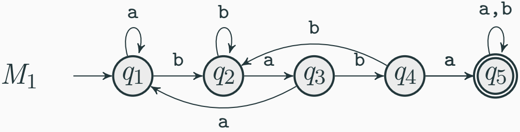
Autômatos finitos determinísticos - Transição estendida

Considere M_1 novamente:



- $\hat{\delta}_1(q_1, \texttt{aabba}) = q_3$
- $\hat{\delta}_1(q_2, \varepsilon) = q_2$
- $\hat{\delta}_1(q_4, \texttt{abbba}) = q_5$

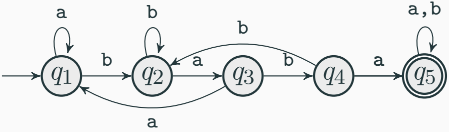
Computando $\omega = \texttt{aabbabaa}$ em M_1



estados ativos	símbolo lido
q_1	

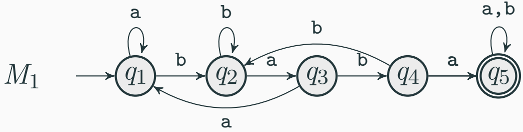
Computando $\omega =$ `aabbabaa` em M_1

M_1



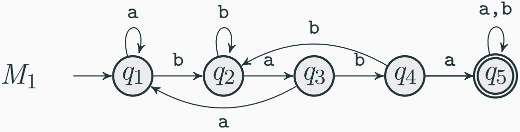
estados ativos	símbolo lido
q_1	<u>a</u>abbabaa
q_1	

Computando $\omega = \mathtt{aabbabaa}$ em M_1



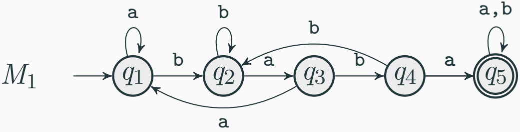
estados ativos	símbolo lido
q_1	$\underline{\mathtt{a}}\mathtt{abbabaa}$
q_1	$\mathtt{a}\underline{\mathtt{a}}\mathtt{bbabaa}$
q_1	

Computando $\omega = \texttt{aabbabaa}$ em M_1



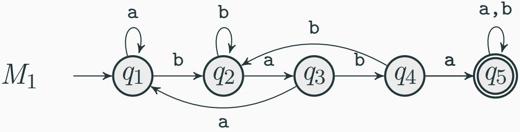
estados ativos	símbolo lido
q_1	$\texttt{\underline{a}abbabaa}$
q_1	$\texttt{a\underline{a}bbabaa}$
q_1	$\texttt{aa\underline{b}babaa}$
q_2	

Computando $\omega = \texttt{aabbabaa}$ em M_1



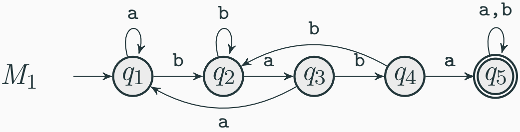
estados ativos	símbolo lido
q_1	aabbabaa
q_1	aabbabaa
q_1	aabbabaa
q_2	aabbabaa
q_2	

Computando $\omega =$ aabbabaa em M_1



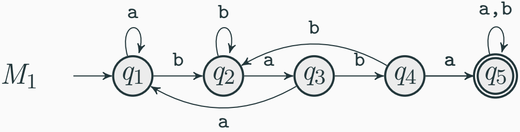
estados ativos	símbolo lido
q_1	aabbabaa
q_1	aabbabaa
q_1	aabbabaa
q_2	aabbabaa
q_2	aabbabaa
q_3	

Computando $\omega =$ aabbabaa em M_1



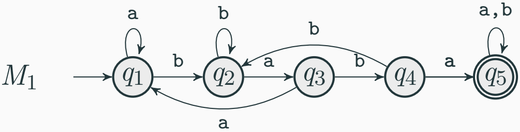
estados ativos	símbolo lido
q_1	aabbabaa
q_1	aabbabaa
q_1	aabbabaa
q_2	aabbabaa
q_2	aabbabaa
q_3	aabbabaa
q_4	aabbabaa
q_5	aabbabaa
q_5	aabbabaa_

Computando $\omega = \texttt{aabbabaa}$ em M_1



estados ativos	símbolo lido
q_1	$\texttt{\underline{a}abbabaa}$
q_1	$\texttt{a\underline{a}bbabaa}$
q_1	$\texttt{aa\underline{b}babaa}$
$q_2 = \hat{\delta}_1(q_1, \texttt{aab})$	$\texttt{aab\underline{b}abaa}$
q_2	$\texttt{aabb\underline{a}baa}$
q_3	$\texttt{aabba\underline{b}aa}$
q_4	$\texttt{aabbab\underline{a}a}$
q_5	$\texttt{aabbaba\underline{a}}$
q_5	$\texttt{aabbabaa\_}$

Computando $\omega =$ aabbabaa em M_1



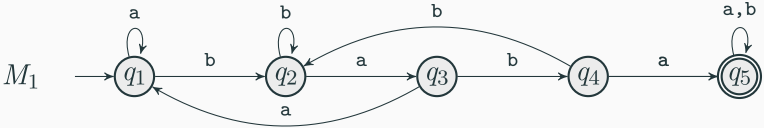
estados ativos	símbolo lido
$q_1 = \hat{\delta}_1(q_1, \varepsilon)$	aabbabaa
$q_1 = \hat{\delta}_1(q_1, \texttt{a})$	aabbabaa
$q_1 = \hat{\delta}_1(q_1, \texttt{aa})$	aabbabaa
$q_2 = \hat{\delta}_1(q_1, \texttt{aab})$	aabbabaa
$q_2 = \hat{\delta}_1(q_1, \texttt{aabb})$	aabbabaa
$q_3 = \hat{\delta}_1(q_1, \texttt{aabba})$	aabbabaa
$q_4 = \hat{\delta}_1(q_1, \texttt{aabbab})$	aabbabaa
$q_5 = \hat{\delta}_1(q_1, \texttt{aabbaba})$	aabbabaa
$q_5 = \hat{\delta}_1(q_1, \texttt{aabbabaa})$	aabbabaa_

Autômatos finitos determinísticos

Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD e seja $\omega \in \Sigma^*$.
Dizemos que M aceita ω se $\hat{\delta}(q_0, \omega) \in F$.
Caso contrário, M rejeita ω.

Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD.
Dizemos que $L(M) = \{\omega \in \Sigma^* \mid M \text{ aceita } \omega\}$ é a linguagem reconhecida por M.

Autômatos finitos determinísticos



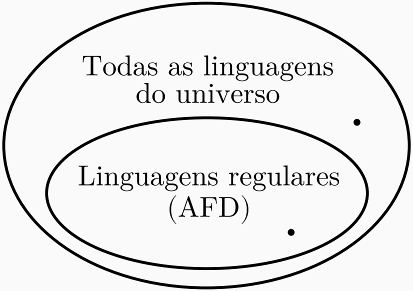
$$L(M_1) = \{\omega \in \{\mathtt{a}, \mathtt{b}\}^* \mid \omega \text{ contém baba como subcadeia } \}$$

Pois:

- $\hat{\delta}_1(q_1, \alpha) = q_1 \iff \alpha \in \Sigma^*$ não contém o padrão desejado
- $\hat{\delta}_1(q_1, \alpha) = q_2 \iff \alpha = \beta\text{b}$ para $\beta \in \Sigma^*$
- $\hat{\delta}_1(q_1, \alpha) = q_3 \iff \alpha = \beta\text{ba}$ para $\beta \in \Sigma^*$
- $\hat{\delta}_1(q_1, \alpha) = q_4 \iff \alpha = \beta\text{bab}$ para $\beta \in \Sigma^*$
- $\hat{\delta}_1(q_1, \alpha) = q_5 \iff \alpha = \beta\text{baba}\gamma$ para $\beta, \gamma \in \Sigma^*$

Linguagens regulares

Uma linguagem é dita regular se algum AFD a reconhece.



Não determinismo

Autômatos finitos não determinísticos

Um autômato finito não determinístico (AFN) é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$ em que

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

Autômatos finitos não determinísticos - Exemplo

$N_1 = (Q_1, \Sigma_1, \delta_1, 1, F_1)$ com $Q_1 = \{1, 2, 3, 4, 5\}$, $\Sigma_1 = \{\mathtt{a}, \mathtt{b}\}$, $F_1 = \{5\}$ e δ_1 dada por

δ_1	a	b	ε
1	$\{1\}$	$\{1, 2\}$	$\emptyset$
2	$\{3\}$	$\emptyset$	$\{3\}$
3	$\emptyset$	$\{4\}$	$\emptyset$
4	$\{5\}$	$\emptyset$	$\emptyset$
5	$\{5\}$	$\{5\}$	$\emptyset$

é um AFN.

Autômatos finitos não determinísticos - Exemplo

N_1 também pode ser definido por um diagrama de estados:

N_1

Autômatos finitos não determinísticos - Exemplo

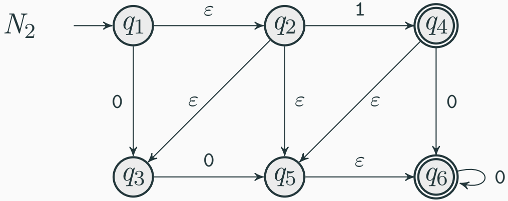
$N_2 = (Q_2, \Sigma_2, \delta_2, q_1, F_2)$ com $Q_2 = \{q_1, q_2, q_3, q_4, q_5, q_6\}$, $\Sigma_2 = \{0, 1\}$, $F_2 = \{q_4, q_6\}$ e δ_2 dada por

δ_2	0	1	ε
q_1	$\{q_3\}$	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\{q_4\}$	$\{q_3, q_5\}$
q_3	$\{q_5\}$	$\emptyset$	$\emptyset$
q_4	$\{q_6\}$	$\emptyset$	$\{q_5\}$
q_5	$\emptyset$	$\emptyset$	$\{q_6\}$
q_6	$\{q_6\}$	$\emptyset$	$\emptyset$

é um AFN.

Autômatos finitos não determinísticos - Exemplo

N_2 também pode ser definido por um diagrama de estados:



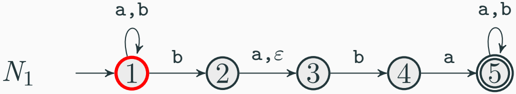
Autômatos finitos não determinísticos - Computação

Ideia: o estado inicial fica ativo; uma transição com rótulo x pode ser seguida a partir de um estado ativo apenas quando x for lido na entrada; uma transição com rótulo ε pode ser seguida a partir de um estado ativo sem que nada da entrada seja lido; o estado atingido pela transição que foi seguida fica ativo.

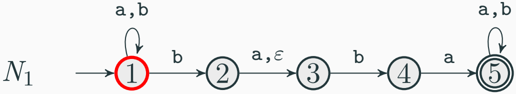
A todo momento pode haver mais de uma possibilidade de transição a ser seguida. Nesse caso, é criada uma cópia da máquina para cada possibilidade.

A cópia desaparece quando não consegue mais seguir transições.

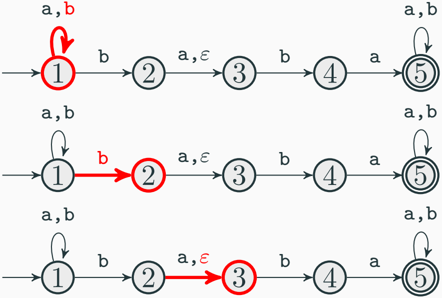
Autômatos finitos não determinísticos - Computação



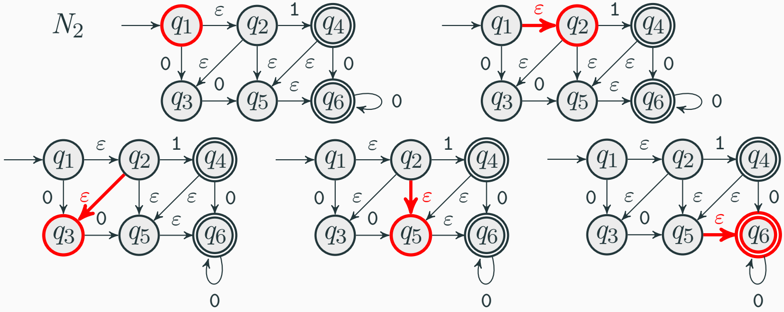
Leu `b`:



Leu b:



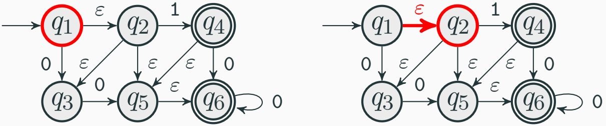
Autômatos finitos não determinísticos - Computação

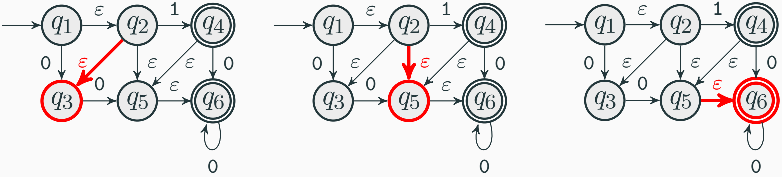


Leu 0:

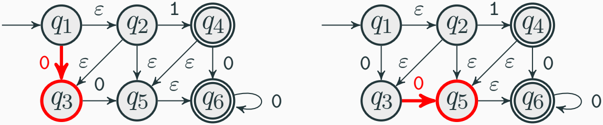
Autômatos finitos não determinísticos - Computação

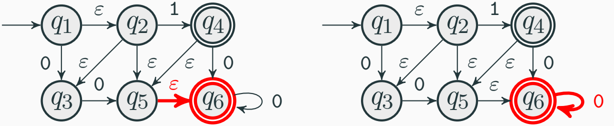
N_2





Leu 0:





Autômatos finitos não determinísticos

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN.
Para $q \in Q$, o ε-fechamento de q, denotado $E(q)$, é o conjunto definido da seguinte forma:

- $q \in E(q)$;
- se $r \in E(q)$ e $s \in \delta(q, \varepsilon)$, então $s \in E(q)$.

Autômatos finitos não determinísticos

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN.
Para $q \in Q$, o ε-fechamento de q, denotado $E(q)$, é o conjunto definido da seguinte forma:

- $q \in E(q)$;
- se $r \in E(q)$ e $s \in \delta(q, \varepsilon)$, então $s \in E(q)$.

Ou seja:

- $E(q)$ é o conjunto de estados que são alcançáveis a partir de q seguindo 0 ou mais transições com rótulo ε.

Autômatos finitos não determinísticos

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN.
Para $q \in Q$, o ε-fechamento de q, denotado $E(q)$, é o conjunto definido da seguinte forma:

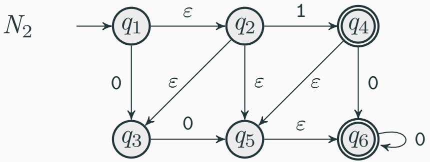
- $q \in E(q)$;
- se $r \in E(q)$ e $s \in \delta(q, \varepsilon)$, então $s \in E(q)$.

Ou seja:

- $E(q)$ é o conjunto de estados que são alcançáveis a partir de q seguindo 0 ou mais transições com rótulo ε.
- $E(q)$ é o conjunto dos estados que ficam ativos no mesmo instante em que q fica ativo.

Autômatos finitos não determinísticos - ε-fechamento

Considere N_2 novamente:



- $E(q_1) = \{q_1, q_2, q_3, q_5, q_6\}$
- $E(q_2) = \{q_2, q_3, q_5, q_6\}$
- $E(q_3) = \{q_3\}$
- $E(q_4) = \{q_4, q_5, q_6\}$
- $E(q_5) = \{q_5, q_6\}$
- $E(q_6) = \{q_6\}$

Abusos de notação

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN, seja $S \subseteq Q$ e seja $x \in \Sigma$.

Usaremos

$$\delta(S, x) = \bigcup_{q \in S} \delta(q, x)$$

para denotar quais estados ficam ativos ao seguir transições rotuladas com x a partir dos estados de S.

Usaremos

$$E(S) = \bigcup_{q \in S} E(q)$$

para denotar quais estados ficam ativos no mesmo instante em que os estados de S ficam ativos.

Autômatos finitos não determinísticos

Dado um AFN $N = (Q, \Sigma, \delta, q_0, F)$, definimos a função de transição estendida de N, $\hat{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$ como

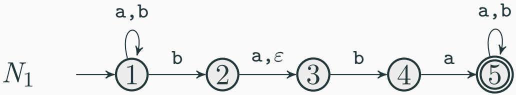
$$\hat{\delta}(q, \omega) = \begin{cases} E(q) & \text{se } \omega = \varepsilon \\ E(\delta(\hat{\delta}(q, \alpha), x)) & \text{se } \omega = \alpha x \text{ e } x \in \Sigma \end{cases}$$

Dado um AFN $N = (Q, \Sigma, \delta, q_0, F)$, definimos a função de transição estendida de N, $\hat{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$ como

$$\hat{\delta}(q, \omega) = \begin{cases} E(q) & \text{se } \omega = \varepsilon \\ E(\delta(\hat{\delta}(q, \alpha), x)) & \text{se } \omega = \alpha x \text{ e } x \in \Sigma \end{cases}$$

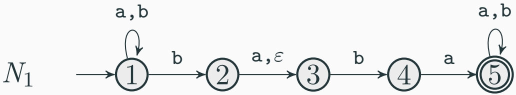
Ou seja, $\hat{\delta}(q, \omega)$ é o *conjunto* de estados ativos em N após computar toda uma cadeia ω a partir do estado q.

Autômatos finitos não determinísticos - Transição estendida



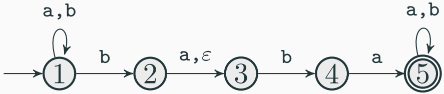
- $\hat{\delta}_1(1, \varepsilon) = E(1) = \{1\}$

Autômatos finitos não determinísticos - Transição estendida



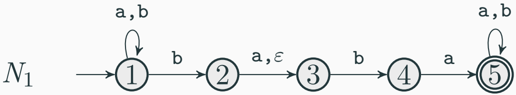
- $\hat{\delta}_1(1, \varepsilon) = E(1) = \{1\}$
- $\hat{\delta}_1(1, \mathtt{a}) = E(\delta_1(\hat{\delta}_1(1, \varepsilon), \mathtt{a})) = E(\delta_1(\{1\}, \mathtt{a})) = E(\{1\}) = \{1\}$

N_1



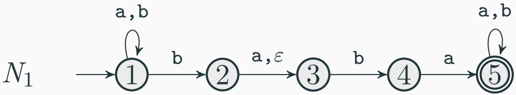
- $\hat{\delta_1}(1, \varepsilon) = E(1) = \{1\}$
- $\hat{\delta_1}(1, \mathtt{a}) = E(\delta_1(\hat{\delta_1}(1, \varepsilon), \mathtt{a})) = E(\delta_1(\{1\}, \mathtt{a})) = E(\{1\}) = \{1\}$
- $\hat{\delta_1}(1, \mathtt{ab}) = E(\delta_1(\hat{\delta_1}(1, \mathtt{a}), \mathtt{b})) = E(\delta_1(\{1\}, \mathtt{b})) = E(\{1, 2\}) = \{1, 2, 3\}$

Autômatos finitos não determinísticos - Transição estendida



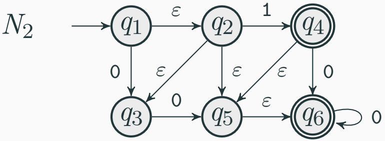
- $\hat{\delta_1}(1, \varepsilon) = E(1) = \{1\}$
- $\hat{\delta_1}(1, \mathtt{a}) = E(\delta_1(\hat{\delta_1}(1, \varepsilon), \mathtt{a})) = E(\delta_1(\{1\}, \mathtt{a})) = E(\{1\}) = \{1\}$
- $\hat{\delta_1}(1, \mathtt{ab}) = E(\delta_1(\hat{\delta_1}(1, \mathtt{a}), \mathtt{b})) = E(\delta_1(\{1\}, \mathtt{b})) = E(\{1, 2\}) = \{1, 2, 3\}$
- $\hat{\delta_1}(1, \mathtt{abb}) = E(\delta_1(\hat{\delta_1}(1, \mathtt{ab}), \mathtt{b})) = E(\delta_1(\{1, 2, 3\}, \mathtt{b})) = E(\{1, 2, 4\}) = \{1, 2, 3, 4\}$

Autômatos finitos não determinísticos - Transição estendida



- $\hat{\delta_1}(1, \varepsilon) = E(1) = \{1\}$
- $\hat{\delta_1}(1, \mathtt{a}) = E(\delta_1(\hat{\delta_1}(1, \varepsilon), \mathtt{a})) = E(\delta_1(\{1\}, \mathtt{a})) = E(\{1\}) = \{1\}$
- $\hat{\delta_1}(1, \mathtt{ab}) = E(\delta_1(\hat{\delta_1}(1, \mathtt{a}), \mathtt{b})) = E(\delta_1(\{1\}, \mathtt{b})) = E(\{1, 2\}) = \{1, 2, 3\}$
- $\hat{\delta_1}(1, \mathtt{abb}) = E(\delta_1(\hat{\delta_1}(1, \mathtt{ab}), \mathtt{b})) = E(\delta_1(\{1, 2, 3\}, \mathtt{b})) = E(\{1, 2, 4\}) = \{1, 2, 3, 4\}$
- $\hat{\delta_1}(2, \mathtt{a}) = E(\delta_1(\hat{\delta_1}(2, \varepsilon), \mathtt{a})) = E(\delta_1(\{2, 3\}, \mathtt{a})) = E(\{3\}) = \{3\}$

Autômatos finitos não determinísticos - Transição estendida



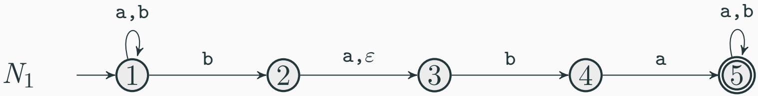
- $\hat{\delta}_2(q_1, \varepsilon) = E(q_1) = \{q_1, q_2, q_3, q_5, q_6\}$
- $\hat{\delta}_2(q_1, \mathtt{0}) = E(\delta_2(\hat{\delta}_2(q_1, \varepsilon), \mathtt{0})) = E(\delta_2(\{q_1, q_2, q_3, q_5, q_6\}, \mathtt{0})) = E(\{q_3, q_5, q_6\}) = \{q_3, q_5, q_6\}$
- $\hat{\delta}_2(q_1, \mathtt{01}) = E(\delta_2(\hat{\delta}_2(q_1, \mathtt{0}), \mathtt{1})) = E(\delta_2(\{q_3, q_5, q_6\}, \mathtt{1})) = E(\emptyset) = \emptyset$

Computando $\omega = \texttt{abbaab}$ em N_1

N_1

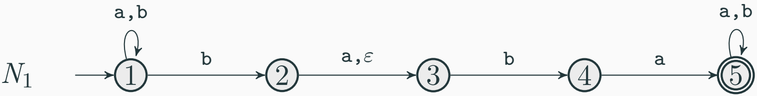
estados ativos	símbolo lido
1	

Computando $\omega =$ `abbaab` em N_1



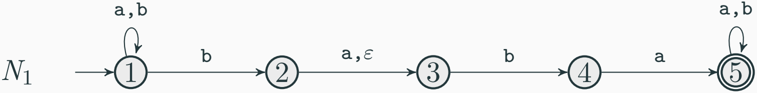
estados ativos	símbolo lido
1	<u>a</u>bbaab
1	

Computando $\omega =$ `abbaab` em N_1



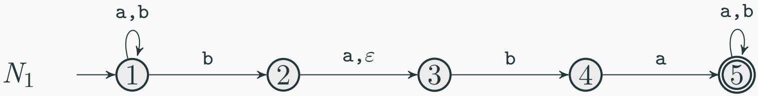
estados ativos	símbolo lido
1	abbaab
1	abbaab
$1, 2, 3$	

Computando $\omega = \texttt{abbaab}$ em N_1



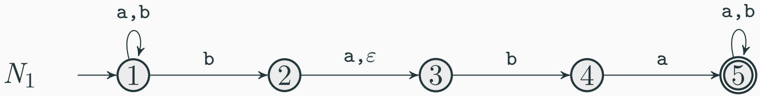
estados ativos	símbolo lido
1	$\underline{\texttt{a}}\texttt{bbaab}$
1	$\texttt{a}\underline{\texttt{b}}\texttt{baab}$
$1, 2, 3$	$\texttt{ab}\underline{\texttt{b}}\texttt{aab}$
$1, 2, 3, 4$	

Computando $\omega = \texttt{abbaab}$ em N_1

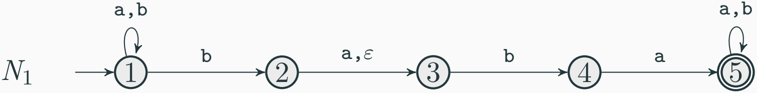


estados ativos	símbolo lido
1	abbaab
1	abbaab
$1, 2, 3$	abbaab
$1, 2, 3, 4$	abbaab
$1, 3, 5$	

Computando $\omega =$ abbaab em N_1

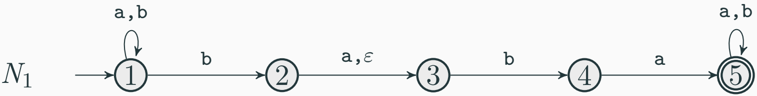


estados ativos	símbolo lido
1	abbaab
1	abbaab
$1, 2, 3$	abbaab
$1, 2, 3, 4$	abbaab
$1, 3, 5$	abbaab
$1, 5$	



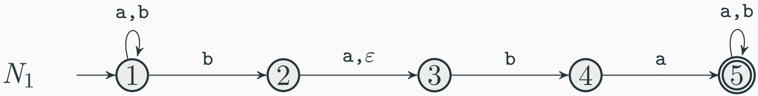
estados ativos	símbolo lido
1	$\texttt{\underline{a}bbaab}$
1	$\texttt{a\underline{b}baab}$
$1, 2, 3$	$\texttt{ab\underline{b}aab}$
$1, 2, 3, 4$	$\texttt{abb\underline{a}ab}$
$1, 3, 5$	$\texttt{abba\underline{a}b}$
$1, 5$	$\texttt{abbaa\underline{b}}$
$1, 2, 3, 5$	

Computando $\omega = \texttt{abbaab}$ em N_1



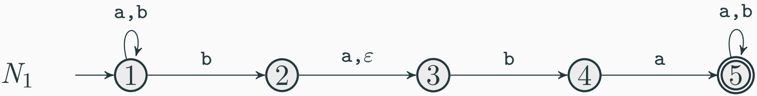
estados ativos	símbolo lido
1	$\texttt{\underline{a}bbaab}$
1	$\texttt{a\underline{b}baab}$
$1, 2, 3$	$\texttt{ab\underline{b}aab}$
$1, 2, 3, 4$	$\texttt{abb\underline{a}ab}$
$1, 3, 5$	$\texttt{abba\underline{a}b}$
$1, 5$	$\texttt{abbaa\underline{b}}$
$1, 2, 3, 5$	$\texttt{abbaab\_}$

Computando $\omega = \texttt{abbaab}$ em N_1



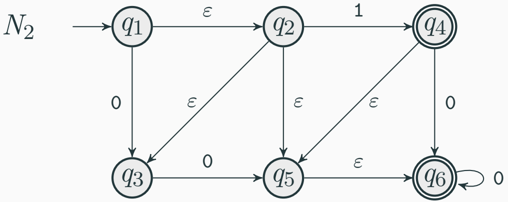
estados ativos	símbolo lido
1	$\texttt{\underline{a}bbaab}$
1	$\texttt{a\underline{b}baab}$
$1, 2, 3$	$\texttt{ab\underline{b}aab}$
$\{1, 2, 3, 4\} = \hat{\delta}_1(1, \texttt{abb})$	$\texttt{abb\underline{a}ab}$
$1, 3, 5$	$\texttt{abba\underline{a}b}$
$1, 5$	$\texttt{abbaa\underline{b}}$
$1, 2, 3, 5$	$\texttt{abbaab\_}$

Computando $\omega = \texttt{abbaab}$ em N_1



estados ativos	símbolo lido
$\{1\} = \hat{\delta}_1(1, \varepsilon)$	$\texttt{\underline{a}bbaab}$
$\{1\} = \hat{\delta}_1(1, \texttt{a})$	$\texttt{a\underline{b}baab}$
$\{1, 2, 3\} = \hat{\delta}_1(1, \texttt{ab})$	$\texttt{ab\underline{b}aab}$
$\{1, 2, 3, 4\} = \hat{\delta}_1(1, \texttt{abb})$	$\texttt{abb\underline{a}ab}$
$\{1, 3, 5\} = \hat{\delta}_1(1, \texttt{abba})$	$\texttt{abba\underline{a}b}$
$\{1, 5\} = \hat{\delta}_1(1, \texttt{abbaa})$	$\texttt{abbaa\underline{b}}$
$\{1, 2, 3, 5\} = \hat{\delta}_1(1, \texttt{abbaab})$	$\texttt{abbaab\_}$

Computando $\omega = 100$ em N_2



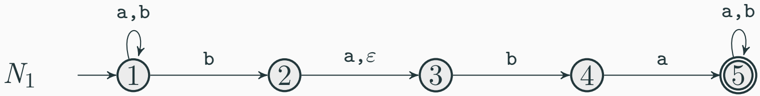
estados ativos	símbolo lido
$\{q_1, q_2, q_3, q_5, q_6\} = \hat{\delta}_2(q_1, \varepsilon)$	$\underline{1}00$
$\{q_4, q_5, q_6\} = \hat{\delta}_2(q_1, 1)$	$1\underline{0}0$
$\{q_6\} = \hat{\delta}_2(q_1, 10)$	$10\underline{0}$
$\{q_6\} = \hat{\delta}_2(q_1, 100)$	100_

Autômatos finitos não determinísticos

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN e seja $\omega \in \Sigma^*$.
Dizemos que N aceita ω se $\hat{\delta}(q_0, \omega) \cap F \neq \emptyset$.
Caso contrário, N rejeita ω.

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN.
Dizemos que $L(N) = \{\omega \in \Sigma^* \mid N \text{ aceita } \omega\}$ é a linguagem reconhecida por N.

Autômatos finitos não determinísticos



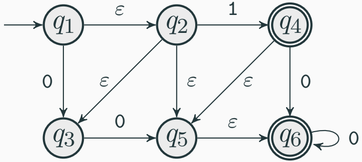
$L(N_1) = \{\omega \in \{\mathtt{a}, \mathtt{b}\}^* \mid \omega$ contém baba ou bba como subcadeia $\}$

Pois:

- $1 \in \hat{\delta}_1(1, \alpha) \;\Leftrightarrow\; \alpha \in \Sigma^*$
- $2 \in \hat{\delta}_1(1, \alpha) \;\Leftrightarrow\; \alpha = \beta\mathtt{b}$ para $\beta \in \Sigma^*$
- $3 \in \hat{\delta}_1(1, \alpha) \;\Leftrightarrow\; \alpha = \beta\mathtt{b}$ ou $\alpha = \beta\mathtt{ba}$ para $\beta \in \Sigma^*$
- $4 \in \hat{\delta}_1(1, \alpha) \;\Leftrightarrow\; \alpha = \beta\mathtt{bb}$ ou $\alpha = \beta\mathtt{bab}$ para $\beta \in \Sigma^*$
- $5 \in \hat{\delta}_1(1, \alpha) \;\Leftrightarrow\; \alpha = \beta\mathtt{bba}\gamma$ ou $\alpha = \beta\mathtt{baba}\gamma$ para $\beta, \gamma \in \Sigma^*$

Autômatos finitos não determinísticos

N_2



$$L(N_2) = \{\varepsilon\} \cup \{10^k \mid k \geq 0\} \cup \{00^k \mid k \geq 0\}$$

Pois:

- $q_1 \in \hat{\delta}_2(q_1, \alpha) \;\Leftrightarrow\; \alpha = \varepsilon$
- $q_2 \in \hat{\delta}_2(q_1, \alpha) \;\Leftrightarrow\; \alpha = \varepsilon$
- $q_3 \in \hat{\delta}_2(q_1, \alpha) \;\Leftrightarrow\; \alpha = \varepsilon$ ou $\alpha = 0$
- $q_4 \in \hat{\delta}_2(q_1, \alpha) \;\Leftrightarrow\; \alpha = 1$
- $q_5 \in \hat{\delta}_2(q_1, \alpha) \;\Leftrightarrow\; \alpha = \varepsilon$ ou $\alpha = 0$ ou $\alpha = 00$ ou $\alpha = 1$
- $q_6 \in \hat{\delta}_2(q_1, \alpha) \;\Leftrightarrow\; \alpha = 0^k$ ou $\alpha = 00^k$ ou $\alpha = 000^k$ ou $\alpha = 10^k$ ou $\alpha = 100^k$, com $k \geq 0$

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.

AFD vs. AFN

Teorema

Todo AFN tem um AFD equivalente.

Demonstração.

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

AFD vs. AFN

Teorema

Todo AFN tem um AFD equivalente.

Demonstração.

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN.
A ideia é criar um AFD que reconhece $L(N)$ fazendo-o simular N. Para isso, o AFD deve lembrar quais são os estados ativos do AFN em cada momento da entrada.

AFD vs. AFN

Teorema

Todo AFN tem um AFD equivalente.

Demonstração.

Seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFN.
A ideia é criar um AFD que reconhece $L(N)$ fazendo-o simular N. Para isso, o AFD deve lembrar quais são os estados ativos do AFN em cada momento da entrada.
Seja $M = (B, \Sigma, \varphi, p_0, T)$ em que:

- $B = \mathcal{P}(Q)$
- Para $R \in B$ e $x \in \Sigma$, $\qquad \varphi(R, x) = \bigcup_{q \in R} E(\delta(q, x))$
- $p_0 = E(q_0)$
- $T = \{R \in B \mid R \cap F \neq \emptyset\}$

AFD vs. AFN

Teorema

Todo AFN tem um AFD equivalente.

Demonstração.

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

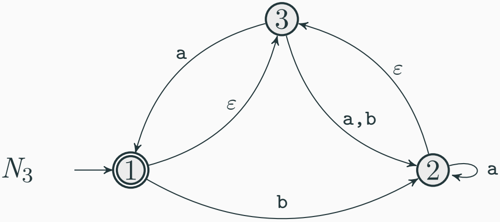
A ideia é criar um AFD que reconhece $L(N)$ fazendo-o simular N. Para isso, o AFD deve lembrar quais são os estados ativos do AFN em cada momento da entrada.

Seja $M = (B, \Sigma, \varphi, p_0, T)$ em que:

- $B = \mathcal{P}(Q)$
- Para $R \in B$ e $x \in \Sigma$, $\quad \varphi(R, x) = \bigcup_{q \in R} E(\delta(q, x))$
- $p_0 = E(q_0)$
- $T = \{R \in B \mid R \cap F \neq \emptyset\}$

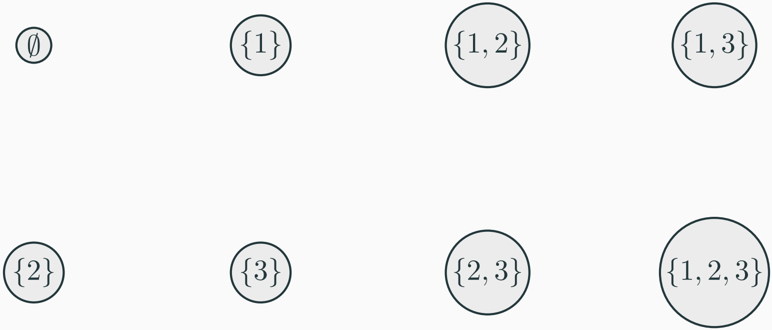
Note que $L(M) = L(N)$ porque $\hat{\delta}(q_0, \omega) = \hat{\varphi}(p_0, \omega)$ para qualquer $\omega \in \Sigma^*$. $\square$

AFD vs. AFN

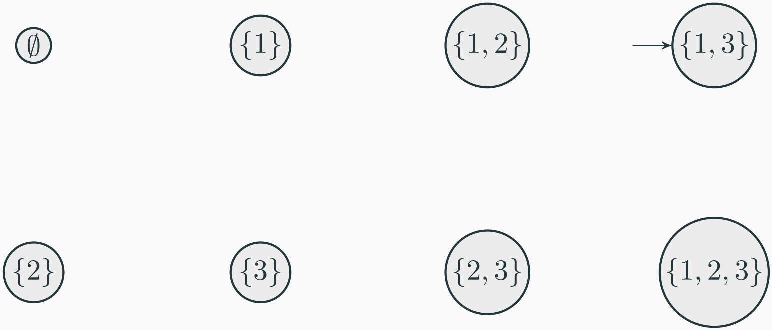


AFD vs. AFN

AFD gerado a partir de N_3:

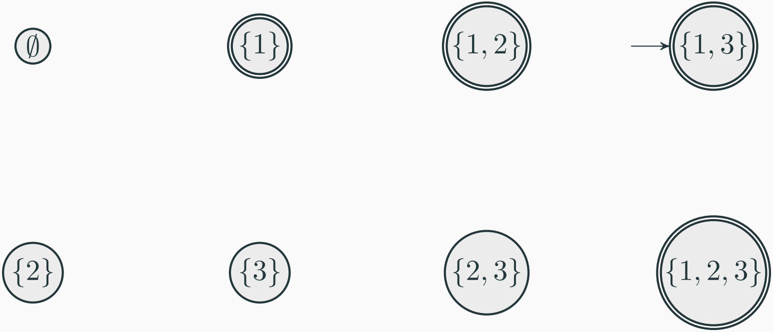


AFD gerado a partir de N_3:



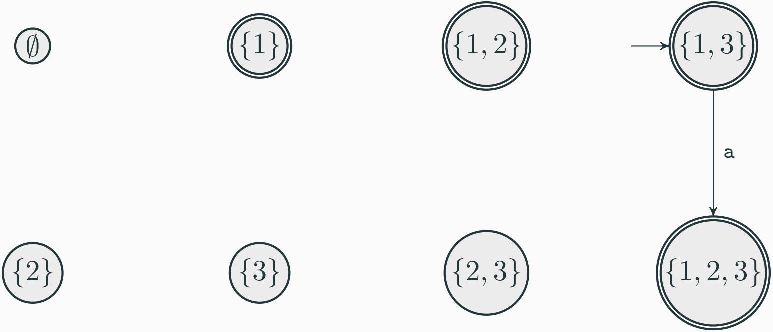
AFD vs. AFN

AFD gerado a partir de N_3:



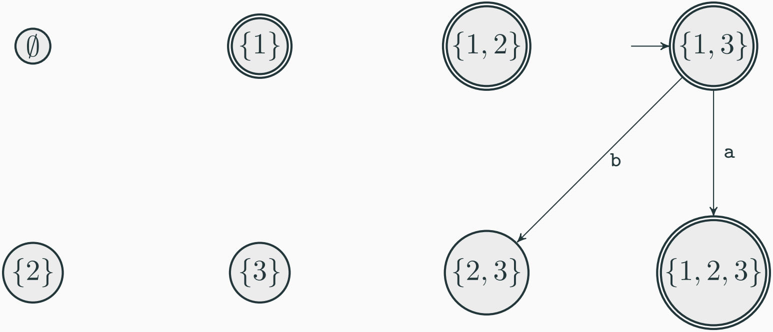
AFD vs. AFN

AFD gerado a partir de N_3:



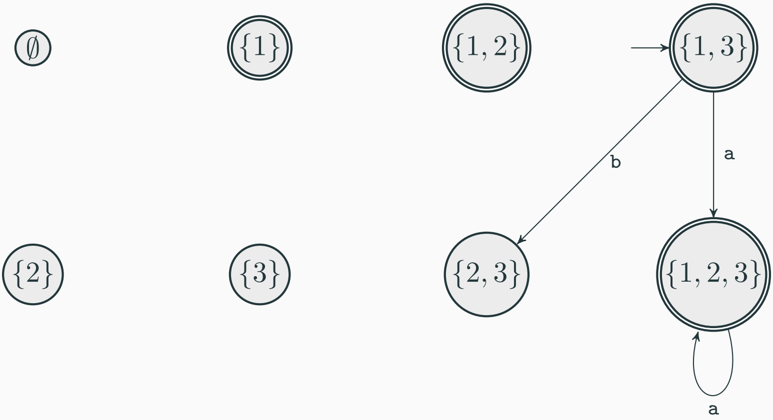
AFD vs. AFN

AFD gerado a partir de N_3:



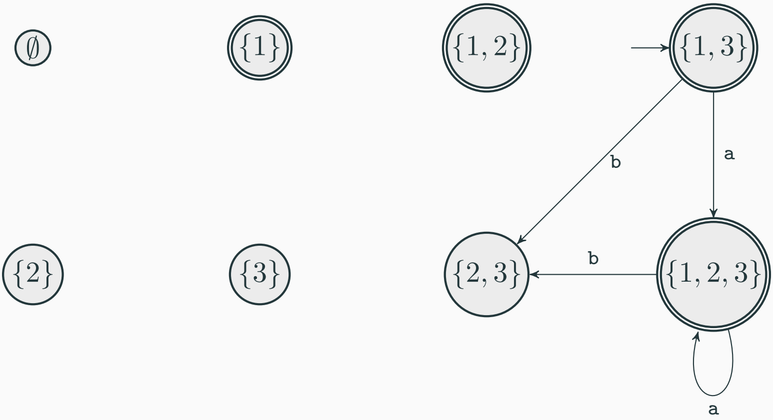
AFD vs. AFN

AFD gerado a partir de N_3:

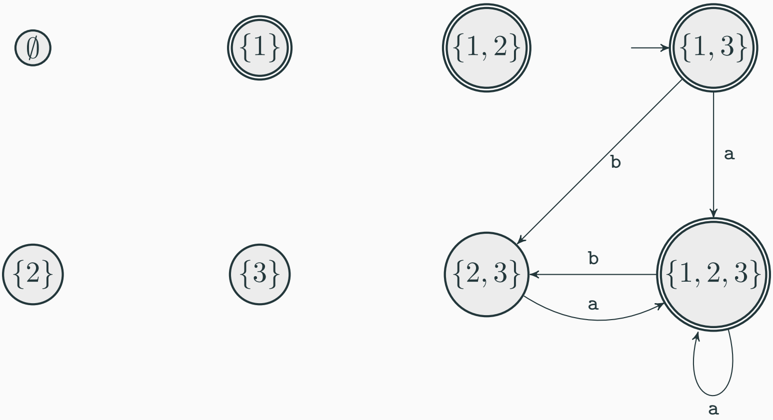


AFD vs. AFN

AFD gerado a partir de N_3:

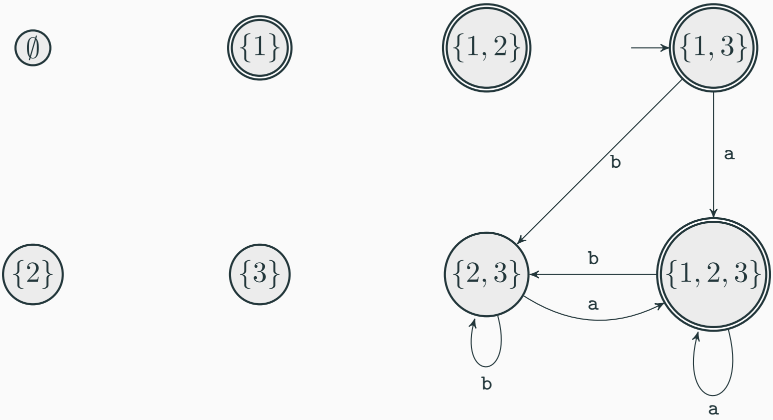


AFD gerado a partir de N_3:



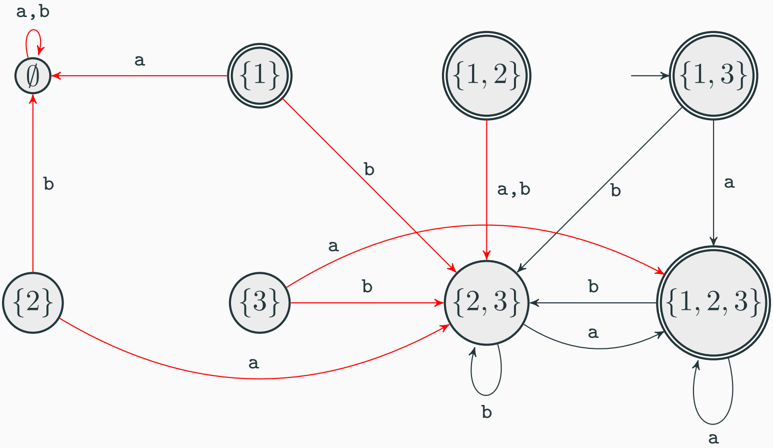
AFD vs. AFN

AFD gerado a partir de N_3:

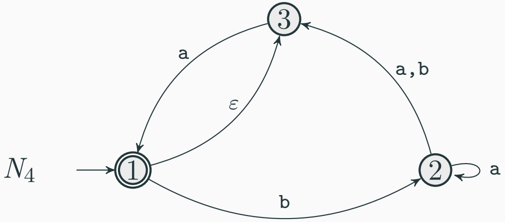


AFD vs. AFN

AFD gerado a partir de N_3:

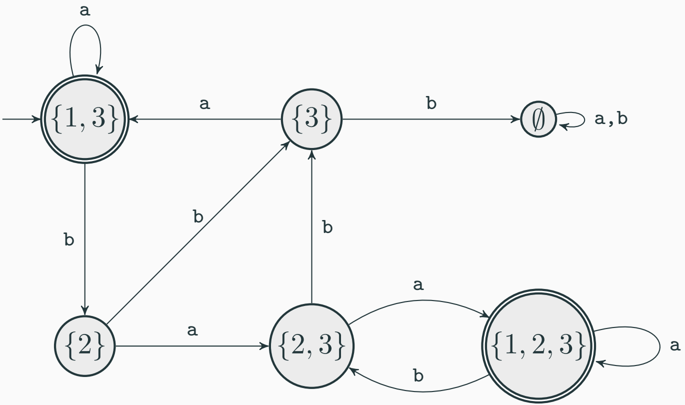


AFD vs. AFN - Exercício



AFD vs. AFN - Exercício

AFD gerado a partir de N_4 (após remoção de estados inalcançáveis):



Linguagens regulares

Corolário

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

Demonstração.

($\Rightarrow$) Se a linguagem é regular, então por definição um AFD a reconhece e todo AFD é um AFN.
($\Leftarrow$) Se um AFN reconhece a linguagem, então pelo teorema anterior podemos construir um AFD equivalente. Logo, por definição, a linguagem é regular. $\square$

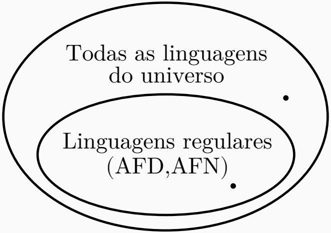
Linguagens regulares

Corolário

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

Demonstração.

($\Rightarrow$) Se a linguagem é regular, então por definição um AFD a reconhece e todo AFD é um AFN.
($\Leftarrow$) Se um AFN reconhece a linguagem, então pelo teorema anterior podemos construir um AFD equivalente. Logo, por definição, a linguagem é regular. $\square$



Propriedades de linguagens regulares

Operações regulares

Sejam A e B linguagens.

São operações regulares:

- **União**: $A \cup B = \{x \mid x \in A \text{ ou } x \in B\}$
- **Concatenação**: $AB = \{xy \mid x \in A \text{ e } x \in B\}$
- **Estrela**: $A^* = \{x_1 x_2 \ldots x_k \mid k \geq 0 \text{ e } x_i \in A\}$

Operações regulares

Considere $L = \{1, 01, 010\}$ e $M = \{\varepsilon, 11\}$.

$$L \cup M = \{\varepsilon, 1, 01, 11, 010\}$$

$$LM = \{1, 111, 01, 0111, 010, 01011\}$$

$$ML = \{1, 01, 010, 111, 1101, 11010\}$$

$$L^* = \{\varepsilon, 1, 11, 101, 010, 0101, 1010, 11101010, \ldots\}$$

Propriedades das operações regulares

1. $\emptyset \cup L = L \cup \emptyset = L$
2. $\{\varepsilon\}L = L\{\varepsilon\} = L$
3. $\emptyset L = L\emptyset = \emptyset$
4. $L(M \cup N) = LM \cup LN$
5. $(M \cup N)L = ML \cup NL$
6. $L \cup L = L$
7. $(L^*)^* = L^*$
8. $\emptyset^* = \{\varepsilon\}$
9. $\{\varepsilon\}^* = \{\varepsilon\}$

Propriedades de fechamento de linguagens regulares

- Uma coleção de objetos é *fechada sob* alguma operação se, aplicando-se essa operação a membros da coleção, recebe-se um objeto ainda na coleção.
  - Por exemplo, números naturais são fechados sob a operação de soma.
  - Por exemplo, números naturais não são fechados sob a operação de divisão.

Propriedades de fechamento de linguagens regulares

As linguagens regulares são fechadas sob:

1. União (teorema a seguir)
2. Concatenação (teorema a seguir)
3. Estrela (teorema a seguir)
4. Interseção (exercício)
5. Complemento (respondido na parte 3 da lista 1)
6. Diferença (exercício)
7. Reverso (exercício na parte 3 da lista 1)

União

Teorema

A união de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares.

União

Teorema

A união de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares. Por definição, existem AFDs $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que as reconhecem, respectivamente.

União

Teorema

A união de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares. Por definição, existem AFDs $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que as reconhecem, respectivamente.

A ideia é criar um AFD que reconhece $A_1 \cup A_2$ fazendo-o simular M_1 e M_2 simultaneamente. Para isso, o AFD precisa lembrar qual o estado ativo em M_1 e qual o estado ativo em M_2 em cada momento da entrada.

União

Teorema

A união de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares. Por definição, existem AFDs $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que as reconhecem, respectivamente.

A ideia é criar um AFD que reconhece $A_1 \cup A_2$ fazendo-o simular M_1 e M_2 simultaneamente. Para isso, o AFD precisa lembrar qual o estado ativo em M_1 e qual o estado ativo em M_2 em cada momento da entrada.

Seja $M = (Q, \Sigma, \delta, q_0, F)$ em que

- $Q = Q_1 \times Q_2 = \{(r_1, r_2) \mid r_1 \in Q_1 \text{ e } r_2 \in Q_2\}$
- Para $(r_1, r_2) \in Q$ e $x \in \Sigma$, $\quad \delta((r_1, r_2), x) = (\ \delta_1(r_1, x),\ \delta_2(r_2, x)\ )$
- $q_0 = (q_1, q_2)$
- $F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ ou } r_2 \in F_2\}$

União

Teorema

A união de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares. Por definição, existem AFDs $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que as reconhecem, respectivamente.

A ideia é criar um AFD que reconhece $A_1 \cup A_2$ fazendo-o simular M_1 e M_2 simultaneamente. Para isso, o AFD precisa lembrar qual o estado ativo em M_1 e qual o estado ativo em M_2 em cada momento da entrada.

Seja $M = (Q, \Sigma, \delta, q_0, F)$ em que

- $Q = Q_1 \times Q_2 = \{(r_1, r_2) \mid r_1 \in Q_1 \text{ e } r_2 \in Q_2\}$
- Para $(r_1, r_2) \in Q$ e $x \in \Sigma$, $\quad \delta((r_1, r_2), x) = (\ \delta_1(r_1, x),\ \delta_2(r_2, x)\ )$
- $q_0 = (q_1, q_2)$
- $F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ ou } r_2 \in F_2\}$

Note que $L(M) = A_1 \cup A_2$ porque

$$\hat{\delta}(q_0, \omega) = (r_i, r_j) \ \Leftrightarrow\ \hat{\delta_1}(q_1, \omega) = r_i \text{ e } \hat{\delta_2}(q_2, \omega) = r_j\,.$$

$\square$

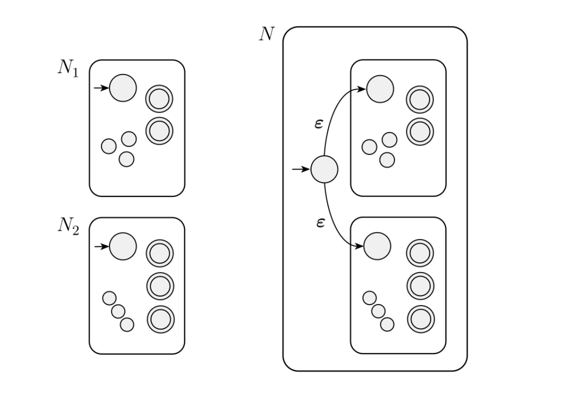
União - Outra demonstração

Teorema

A união de duas linguagens regulares é regular.

Ideia da demonstração.

Criar um único AFN que reconhece $A_1 \cup A_2$ se aproveitando da onisciência do não determinismo:



União - Outra demonstração

Teorema

A união de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares. Por definição, existem AFNs $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que as reconhecem, respectivamente.

União - Outra demonstração

Teorema

A união de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares. Por definição, existem AFNs $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que as reconhecem, respectivamente.
Construa $N = (Q, \Sigma, \delta, q_0, F)$ em que

- $Q = \{q_0\} \cup Q_1 \cup Q_2$
- Para $q \in Q$ e $x \in \Sigma \cup \{\varepsilon\}$,

$$\delta(q, x) = \begin{cases} \delta_1(q, x) & \text{se } q \in Q_1 \\ \delta_2(q, x) & \text{se } q \in Q_2 \\ \{q_1, q_2\} & \text{se } q = q_0 \text{ e } x = \varepsilon \\ \emptyset & \text{se } q = q_0 \text{ e } x \neq \varepsilon \end{cases}$$

- $F = F_1 \cup F_2$

Note que $L(N) = A_1 \cup A_2$ porque ... $\square$

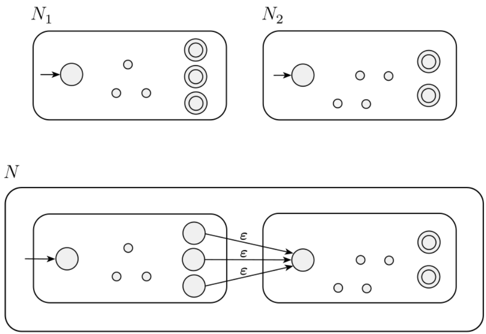
Concatenação

Teorema

A concatenação de duas linguagens regulares é regular.

Ideia da demonstração.

Criar um único AFN que reconhece $A_1 A_2$ se aproveitando da onisciência do não determinismo:



Concatenação

Teorema

A concatenação de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares. Por definição, existem AFNs $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que as reconhecem, respectivamente.

Concatenação

Teorema

A concatenação de duas linguagens regulares é regular.

Demonstração.

Sejam A_1 e A_2 duas linguagens regulares. Por definição, existem AFNs $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que as reconhecem, respectivamente.
Seja $N = (Q, \Sigma, \delta, q_0, F)$ em que

- $Q = Q_1 \cup Q_2$
- Para $q \in Q$ e $x \in \Sigma \cup \{\varepsilon\}$,

$$\delta(q, x) = \begin{cases} \delta_1(q, x) & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q, x) & \text{se } q \in F_1 \text{ e } x \neq \varepsilon \\ \delta_1(q, x) \cup \{q_2\} & \text{se } q \in F_1 \text{ e } x = \varepsilon \\ \delta_2(q, x) & \text{se } q \in Q_2 \end{cases}$$

- $q_0 = q_1$
- $F = F_2$

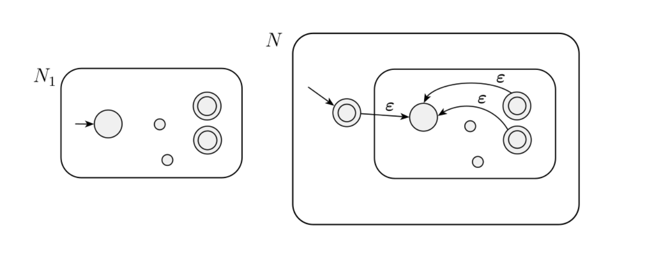
Note que $L(N) = A_1 A_2$ porque ... □

Teorema

A estrela de uma linguagem regular é regular.

Ideia da demonstração.

Criar um único AFN que reconhece A^* se aproveitando da onisciência do não determinismo:



Estrela

Teorema

A estrela de uma linguagem regular é regular.

Demonstração.

Sejam A uma linguagem regular. Por definição, existe um AFN $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ que a reconhece.

Estrela

Teorema

A estrela de uma linguagem regular é regular.

Demonstração.

Sejam A uma linguagem regular. Por definição, existe um AFN $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ que a reconhece.
Seja $N = (Q, \Sigma, \delta, q_0, F)$ em que

- $Q = \{q_0\} \cup Q_1$
- Para $q \in Q$ e $x \in \Sigma \cup \{\varepsilon\}$,

$$\delta(q,x) = \begin{cases} \delta_1(q,x) & \text{se } q \in Q_1 \text{ e } q \notin F_1 \\ \delta_1(q,x) & \text{se } q \in Q_1 \text{ e } x \neq \varepsilon \\ \delta_1(q,x) \cup \{q_1\} & \text{se } q \in F_1 \text{ e } x = \varepsilon \\ \{q_1\} & \text{se } q = q_0 \text{ e } x = \varepsilon \\ \emptyset & \text{se } q = q_0 \text{ e } x \neq \varepsilon \end{cases}$$

- $F = \{q_0\} \cup F_1$

Note que $L(N) = A^*$ porque ... □

Estrela

Por que a demonstração do fechamento sob estrela precisou criar um estado inicial novo?