Autómatos Finitos

Linguagens Formais e Autómatos

Francisco Coelho
fc@di.uevora.pt

Departamento de Informática
Escola de Ciências e Tecnologia
Universidade de Évora

UNIVERSIDADE DE ÉVORA

Um autómato de estados finitos (abrev. AFD) é um tuplo

$$A = (Q, \Sigma, \delta, q_I, F)$$

em que

- estados de controlo Q é um conjunto finito;
- alfabeto de entrada Σ é um alfabeto;
- transição $\delta : Q \times \Sigma \to Q$;
- estado inicial $q_I \in Q$;
- estados finais (ou de aceitação) $F \subseteq Q$;

Um AFD processa palavras letra-a-letra. Se o processamento de uma palavra terminar num estado final essa palavra é "aceite" caso contrário é "rejeitada".

Configuração e Computação

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFD.

Uma configuração de A é um par $[q, w] \in Q \times \Sigma^*$ onde q é o "estado corrente" do autómato e w é a parte da palavra que ainda falta processar.

A computação da palavra $w = a_1 a_2 \cdots a_n \in \Sigma^*$ pelo AFD A é a sequência de configurações

$$[s_0, a_1 a_2 \cdots a_n] \vdash_A [s_1, a_2 \cdots a_n] \vdash_A \cdots \vdash_A [s_n, \varepsilon]$$

em que

base $s_0 = q_I$

passo $s_i = \delta(s_{i-1}, a_i)$ para $i \geq 1$

Aceita $w = 000$?

$$A = \begin{cases} Q = \{q_0, q_1, q_2, q_3\}, \\ \Sigma = \{0, 1\}, \\ \delta, \\ q_I = q_0, \\ F = \{q_1, q_2\} \end{cases}$$

onde δ é

δ	0	1
q_0	q_1	q_3
q_1	q_2	q_3
q_2	q_2	q_2
q_3	q_3	q_3

$$\begin{array}{ccc|c} q_0 & & & \\ 0 & 0 & 0 & \delta(q_0, 0) = q_1 \\ \hline \rightarrow & q_1 & & \\ & 0 & 0 & \delta(q_1, 0) = q_2 \\ \hline & \rightarrow & q_2 & \\ & & 0 & \delta(q_2, 0) = q_2 \\ \hline & & q_2 & \in F \\ & & & \text{aceita} \end{array}$$

$$A = \begin{cases} Q = \{q_0, q_1, q_2, q_3\}, \\ \Sigma = \{0, 1\}, \\ \delta, \\ q_I = q_0, \\ F = \{q_1, q_2\} \end{cases}$$

onde δ é

δ	0	1
q_0	q_1	q_3
q_1	q_2	q_3
q_2	q_2	q_2
q_3	q_3	q_3

Aceita $w = 010$?

q_0 0	1	0	$\delta(q_0, 0) = q_1$
	q_1 1	0	$\delta(q_1, 1) = q_3$
		q_3 0	$\delta(q_3, 0) = q_3$
		q_3	$\notin F$ não aceita

Diagrama de um AFD

UNIVERSIDADE DE ÉVORA

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFD.

O diagrama de A é o digrafo $\mathcal{G}(A)$ definido por

- a cada transição $q_j = \delta(q_i, a)$ corresponde a arco

$$\textcircled{i} \xrightarrow{a} \textcircled{j}$$

- o estado inicial é indicado por uma seta:

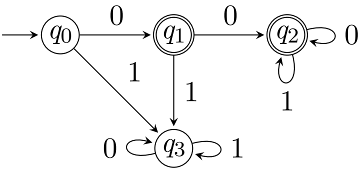
$$\longrightarrow \textcircled{i}$$

- os estados finais por círculos duplos:

$$\textcircled{\textcircled{j}}$$

Exemplo

δ	0	1
q_0	q_1	q_3
q_1	q_2	q_3
q_2	q_2	q_2
q_3	q_3	q_3



Transição Estendida

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFD.

A função de transição estendida de A é

$$\hat{\delta} : Q \times \Sigma^* \to Q$$

em que

$$\begin{aligned} \text{base} \quad & \hat{\delta}(q, \varepsilon) = q \text{ e } \hat{\delta}(q, a) = \delta(q, a) \text{ para } a \in \Sigma \\ \text{passo} \quad & \hat{\delta}(q, wa) = \delta\left(\hat{\delta}(q, w)\, , a\right) \end{aligned}$$

Além disso

$$\hat{\delta}(q, w) = q' \iff [q, w] \vdash_A^* [q', \varepsilon]$$

Exemplo de Transições Estendidas

$$A = \begin{cases} Q = \{q_0, q_1, q_2, q_3\}, \\ \Sigma = \{0, 1\}, \\ \delta, \\ q_I = q_0, \\ F = \{q_1, q_2\} \end{cases}$$

onde δ é

δ	0	1
q_0	q_1	q_3
q_1	q_2	q_3
q_2	q_2	q_2
q_3	q_3	q_3

$$\begin{aligned}
\hat{\delta}(q_0, 000) &= \delta\Big(\hat{\delta}(q_0, 00), 0\Big) \\
&= \delta\Big(\delta\Big(\hat{\delta}(q_0, 0), 0\Big), 0\Big) \\
&= \delta\Big(\delta(\delta(q_0, 0), 0), 0\Big) \\
&= \delta(\delta(q_1, 0), 0) \\
&= \delta(q_2, 0) \\
&= q_2 \\
\hat{\delta}(q_0, 010) &= \delta\Big(\hat{\delta}(q_0, 01), 0\Big) \\
&= \delta\Big(\delta\Big(\hat{\delta}(q_0, 0), 1\Big), 0\Big) \\
&= \delta\Big(\delta(\delta(q_0, 0), 1), 0\Big) \\
&= \delta(\delta(q_1, 1), 0) \\
&= \delta(q_2, 0) \\
&= q_3
\end{aligned}$$

UNIVERSIDADE DE ÉVORA

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFD.

Uma palavra $w \in \Sigma^*$ é aceite por A se $\hat{\delta}(q_I, w) \in F$.

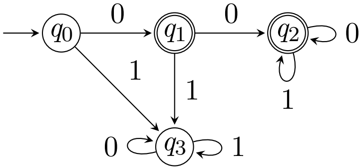
A linguagem reconhecida (ou aceite) por A é o conjunto das palavras aceites por A

$$\mathcal{L}(A) = \left\{ w \ : \ \hat{\delta}(q_I, w) \in F \right\}$$

Dois autómatos finitos são equivalentes se reconhecem a mesma linguagem.

Exemplo

O autómato



reconhece a linguagem $0 + 00\,(0+1)^*$.

Contexto

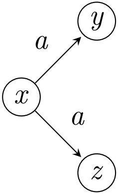
UNIVERSIDADE DE ÉVORA

Num AFD, a transição δ faz corresponder um único estado a cada combinação $(q, a) \in Q \times \Sigma$.

Mas esta condição restringe as possibilidades dos AFDs.

1. não existem estados futuros alternativos (dado (q, a));
2. não há transições "internas", sem processar símbolos da palavra;

Transições Múltiplas



Numa transição múltipla, num certo estado x, dado uma letra, a, é possível "escolher" entre vários estados futuros y, z.

Transições ε (ou Vazias)

$$x \xrightarrow{\varepsilon} y$$

Numa transição ε, é possível passar de um certo estado x para um estado futuro y sem “ler” nenhuma letra da palavra dada.

Um autómato finito não-determinista (com transições ε) (abrev. AFND) é um tuplo

$$A = (Q, \Sigma, \delta, q_I, F)$$

em que

- estados de controlo Q é um conjunto finito;
- alfabeto de entrada Σ é um alfabeto;
- transição $\delta : Q \times \left(\Sigma \cup \{\varepsilon\}\right) \to \mathcal{P}(Q)$;
- estado inicial $q_I \in Q$;
- estados finais (ou de aceitação) $F \subseteq Q$;

  Por vezes são proíbidas as transições ε e a transição fica $\delta : Q \times \Sigma \to \mathcal{P}(Q)$.

Linguagem Reconhecida

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFND.

Uma palavra $w \in \Sigma^*$ é aceite por A se existe uma computação de w por A que termina num estado final depois de terem sido lidos todos os símbolos de w:

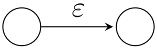
$$[q_I, w] \vdash_A^* [q, \varepsilon] \text{ onde } q \in F$$

A linguagem reconhecida (ou aceite) por A é o conjunto das palavras aceites por A

$$\mathcal{L}(A) = \left\{w \ : \ \text{existe } [q_I, w] \vdash_A^* [q, \varepsilon] \text{ onde } q \in F\right\}$$

Diagrama de um AFND

Dado um AFND, A, o grafo $\mathcal{G}(A)$ representa-se como se fosse dum AFD, atendendo que algumas arcos podem ter etiqueta ε, correspondentes a transições vazias:



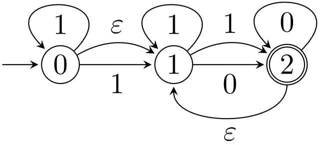
Também pode acontecer que $\delta(q, a) = \emptyset$. Neste caso não há estado seguinte. A computação termina e a palavra dada é rejeitada.

Exemplo: Construção de um Diagrama

Dado um AFND por

q	0	1	ε
0	$\emptyset$	$\{0,1\}$	$\{1\}$
1	$\{2\}$	$\{1,2\}$	$\emptyset$
2	$\{2\}$	$\emptyset$	$\{1\}$

obtemos o diagrama

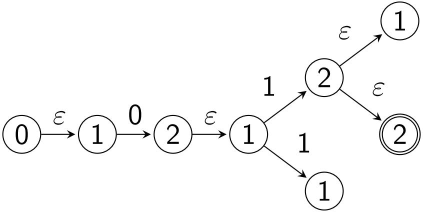


A palavra 01 tem três computações possíveis:

- $0 \xrightarrow{\varepsilon} 1 \xrightarrow{0} 2 \xrightarrow{\varepsilon} 1 \xrightarrow{1} 1$ (rejeita);
- $0 \xrightarrow{\varepsilon} 1 \xrightarrow{0} 2 \xrightarrow{\varepsilon} 1 \xrightarrow{1} 2$ (aceita);
- $0 \xrightarrow{\varepsilon} 1 \xrightarrow{0} 2 \xrightarrow{\varepsilon} 1 \xrightarrow{1} 2 \xrightarrow{\varepsilon} 1$ (rejeita);

Árvore de Computação

As computações de uma palavra podem ser representadas por uma árvore:



O arco $2 \xrightarrow{\varepsilon} 2$ indica o caminho que termina em 2

Fecho Vazio

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFND e $q \in Q$ um estado de controlo.

O $\text{fecho-}\varepsilon\,(q)$ (ou fecho vazio de q) é o conjunto de todos os estados alcançáveis através de zero ou mais transições vazias a partir de q:

base $q \in \text{fecho-}\varepsilon\,(q)$

passo se $p \in \text{fecho-}\varepsilon\,(q)$ e $p' \in \delta(p, \varepsilon)$ então $p' \in \text{fecho-}\varepsilon\,(q)$

A função de transição de entrada do AFND A é a função $\delta^\varepsilon : Q \times \Sigma \to \mathcal{P}(Q)$ definida por

$$\delta^\varepsilon(q, a) = \bigcup_{p \in \text{fecho-}\varepsilon(q)} \text{fecho-}\varepsilon\,\big(\delta(p, a)\big)$$

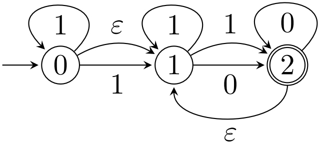
Exemplo — fecho-ε () e transição de entrada

UNIVERSIDADE DE ÉVORA

Dado um AFND por

q	0	1	ε
0	$\emptyset$	$\{0,1\}$	$\{1\}$
1	$\{2\}$	$\{1,2\}$	$\emptyset$
2	$\{2\}$	$\emptyset$	$\{1\}$

com o diagrama



q	fecho-$\varepsilon\,(q)$
0	$\{0,1\}$
1	$\{1\}$
2	$\{2,1\}$

$$\delta^{\varepsilon}(0,1) = \{0,1,2\}$$
$$\delta^{\varepsilon}(0,0) = \{2,1\}$$

Simulação de AFNDs

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFND.

O AFD equivalente a A é o autómato $A' = \left(Q', \Sigma, \delta', q'_I, F'\right)$ em que

- alfabeto de entrada Σ
- estado inicial $q'_I = \text{fecho-}\varepsilon\,(q_I)$
- transição

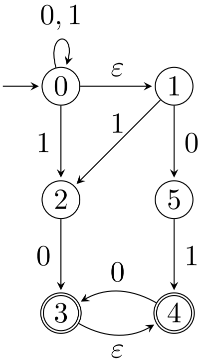
$$\delta'(q, a) = \bigcup_{p \in q} \delta^{\varepsilon}(p, a)$$

- estados de controlo Q' é definido por

base	$q'_I \in Q'$
passo	se $q \in Q'$ então, para cada $a \in \Sigma, \delta'(q, a) \in Q'$
fecho	mais nenhum estado está em Q'

- estados finais $F' = \left\{q \in Q' \;:\; q \cap F \neq \emptyset\right\}$

Exemplo de Simulação



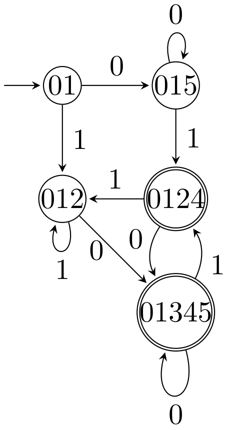
Cálculo de δ' e Q'

δ'	$a = 0$	$a = 1$
$q'_I = \{0, 1\}$	$\{0, 1, 5\}$	$\{0, 1, 2\}$
$q'_1 = \{0, 1, 5\}$	$\{0, 1, 5\} = q'_1$	$\{0, 1, 2, 4\}$
$q'_2 = \{0, 1, 2\}$	$\{0, 1, 3, 4, 5\}$	$\{0, 1, 2\} = q'_2$
$q'_3 = \{0, 1, 2, 4\}$	$\{0, 1, 3, 4, 5\} = q'_4$	$\{0, 1, 2\} = q'_2$
$q'_4 = \{0, 1, 3, 4, 5\}$	$\{0, 1, 3, 4, 5\} = q'_4$	$\{0, 1, 2, 4\} = q'_3$

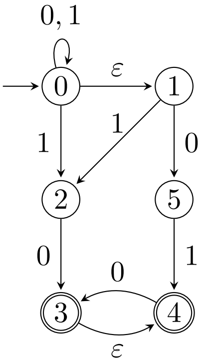
Portanto

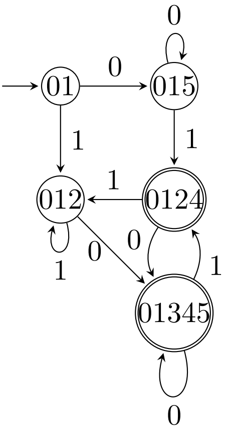
$$
\begin{aligned}
&Q' : \\
&\quad q'_0 = q'_I && = 01, \\
&\quad q'_1 && = 015, \\
&\quad q'_2 && = 012, \\
&\quad q'_3 && = 0124, \\
&\quad q'_4 && = 01345 \\
&F' = \{q'_3, q'_4\}
\end{aligned}
$$

δ'	0	1
q'_0	q'_1	q'_2
q'_1	q'_1	q'_3
q'_2	q'_4	q'_2
q'_3	q'_4	q'_2
q'_4	q'_4	q'_3



Exemplo de Simulação — Continuação



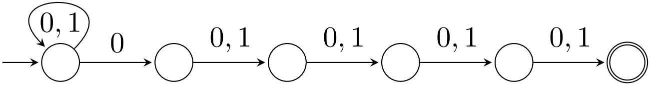


Teorema (Equivalência entre AFD e AFND)

Uma linguagem é reconhecida por um AFND se e só se for reconhecida por um AFD.

Um pequeno AFND simulado por um AFD gigante

Para construir um AFD que aceite o conjunto das palavras binárias cujo quinto símbolo da direita é 0... simula-se o autómato seguinte com um AFD



Mas a conversão (directa) deste AFND para um AFD produz um monstro com 32 estados de controlo...

Em geral, de um AFND com n estados, podemos obter um AFD com 2^n estados.

Estados Indistinguíveis

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFD.

Dois estados $p, q \in Q$ são indistinguíveis se, para cada $w \in \Sigma^*$,

$$\hat{\delta}(p, w) \in F \iff \hat{\delta}(q, w) \in F$$

Se $p \in F$ e $q \in Q \setminus F$ não são indistinguíveis. (porquê?)

Cálculo de Estados Indistinguíveis

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFD.

1. iniciar $P = \{Q \setminus F, F\}$
2. enquanto existirem $p, p' \in P, a \in \Sigma, q, q' \in p$ tais que

$$\delta(q, a) \in p', \delta(q', a) \notin p'$$

   2.1 retirar p de P: $P \leftarrow P \setminus \{p\}$
   2.2 acrescentar $P \leftarrow P \cup \{s \in p \ : \ \delta(s, a) \in p'\}$
   2.3 acrescentar $P \leftarrow P \cup \{s \in p \ : \ \delta(s, a) \notin p'\}$

Este algoritmo faz uma partição P dos estados Q de forma que

- se dois estados estão no mesmo subconjunto são indistinguíveis;
- se dois estados estão em subconjuntos distintos então não são indistinguíveis;

Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFD.

1. Seja P a partição dos estados indistinguíveis
2. Para cada subconjunto $p \in P$ e para cada $a \in \Sigma$
   2.1 seja $q \in p$ (um estado qualquer)
   2.2 seja p' o subconjunto de P que contém $\delta(q, a)$
   2.3 definir $\delta'(p, a) = p'$
3. O AFD mínimo (ou reduzido) de A é

$$A' = \left(P, \Sigma, \delta', q_I', F'\right)$$

   em que

   3.1 estado inicial q_I' é o elemento de P que contém q_I
   3.2 estados finais $F' = \{p \in P \; : \; p \subseteq F\}$

Exemplo

Minimizar o AFD $A = (\{A, B, C, D, E, F\}, \{a, b\}, \delta, A, \{B, D\})$
em que a transição é

δ	a	b
A	B	E
B	E	C
C	D	F
D	F	C
E	F	F
F	E	F

Exemplo – Resolução

δ	a	b
A	B	E
B	E	C
C	D	F
D	F	C
E	F	F
F	E	F

$\rightarrow$

p	δ	a	b
p_1	A	$B \in p_2$	$E \in p_1$
p_1	C	$D \in p_2$	$F \in p_1$
p_1	E	$F \in p_1$	$F \in p_1$
p_1	F	$E \in p_1$	$F \in p_1$
p_2	B	$E \in p_1$	$C \in p_1$
p_2	D	$F \in p_1$	$C \in p_1$

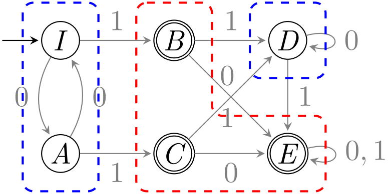
$\rightarrow$

p	δ	a	b
p_1	A	$B \in p_2$	$E \in p_3$
p_1	C	$D \in p_2$	$F \in p_3$
p_3	E	$F \in p_3$	$F \in p_3$
p_3	F	$E \in p_3$	$F \in p_3$
p_2	B	$E \in p_3$	$C \in p_1$
p_2	D	$F \in p_3$	$C \in p_1$

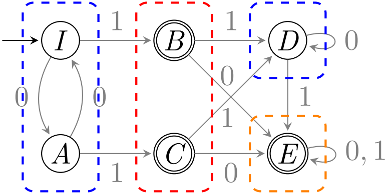
O autómato mínimo é $A' = \left(\{p_1, p_2, p_3\}, \{a, b\}, \delta', p_1, \{p_3\}\right)$ em que

δ'	a	b
p_1	p_2	p_3
p_3	p_3	p_3
p_2	p_3	p_1

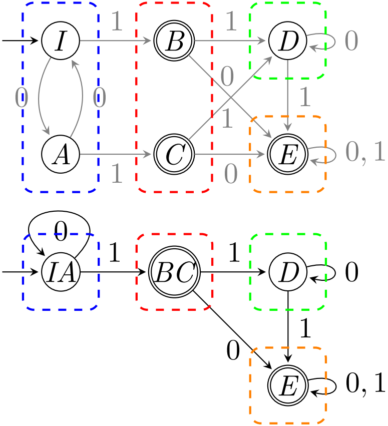
Exemplo com Diagramas



Exemplo com Diagramas



Exemplo com Diagramas



Seja $A = (Q, \Sigma, \delta, q_I, F)$ um AFND.

Existe um AFND $A' = \left(Q \cup \{q'_I, q_f\}, \Sigma, \delta', q'_I, \{q_f\}\right)$ equivalente a A e que

- $q'_I \neq q_f$ e $\{q'_I, q_f\} \cap Q = \emptyset$
- não há transições para o estado inicial q'_I
- o único estado de aceitação é q_f
- não há transições a partir de q_f

A equivalência é obtida estendendo δ com a transição δ' por

- $\delta'(q'_I, \varepsilon) = \{q_I\}$
- $\delta'(p, \varepsilon) = \{q_f\}$ para cada $p \in F$

Sejam $A = \left(Q_A, \Sigma, \delta_A, q_{I_A}, \left\{q_{f_A}\right\}\right)$ e $B = \left(Q_B, \Sigma, \delta_B, q_{I_B}, \left\{q_{f_B}\right\}\right)$ dois AFND nas condições anteriores e com $Q_A \cap Q_B = \emptyset$.

Seja $P = \left(Q_A \cup Q_B, \Sigma, \delta., q_{I_A}, \left\{q_{f_B}\right\}\right)$ o autómato com

$$\begin{aligned} \delta. &= \delta_A \cup \delta_B \\ &\cup \left\{\left(q_{f_A}, \varepsilon, \left\{q_{I_B}\right\}\right)\right\} \end{aligned}$$

Então $\mathcal{L}(P) = \mathcal{L}(A)\,\mathcal{L}(B)$.

Sejam $A = \left(Q_A, \Sigma, \delta_A, q_{I_A}, \left\{q_{f_A}\right\}\right)$ e $B = \left(Q_B, \Sigma, \delta_B, q_{I_B}, \left\{q_{f_B}\right\}\right)$ dois AFND nas condições anteriores e com $Q_A \cap Q_B = \emptyset$.

Seja $U = \left(Q_A \cup Q_B \cup \left\{q_I, q_f\right\}, \Sigma, \delta_\cup, q_I, \left\{q_f\right\}\right)$ o autómato com

$$
\begin{aligned}
\delta_\cup = {} & \delta_A \cup \delta_B \\
& \cup \left\{\left(q_I, \varepsilon, \left\{q_{I_A}, q_{I_B}\right\}\right)\right\} \\
& \cup \left\{\left(q_{f_A}, \varepsilon, \left\{q_f\right\}\right), \left(q_{f_B}, \varepsilon, \left\{q_f\right\}\right)\right\}
\end{aligned}
$$

Então $\mathcal{L}(U) = \mathcal{L}(A) \cup \mathcal{L}(B)$.

N.B. $q_I \neq q_f$ e $(Q_A \cup Q_B) \cap \left\{q_I, q_f\right\} = \emptyset$

Seja $A = \left(Q_A, \Sigma, \delta_A, q_{I_A}, \left\{q_{f_A}\right\}\right)$ um AFND nas condições anteriores.

Seja $I = \left(Q_A \cup \left\{q_I, q_f\right\}, \Sigma, \delta_*, q_I, \left\{q_f\right\}\right)$ o autómato com

$$\begin{aligned}\delta_* = {} & \delta_A \\ & \cup \left\{\left(q_I, \varepsilon, \left\{q_{I_A}, q_f\right\}\right)\right\} \\ & \cup \left\{\left(q_{f_A}, \varepsilon, \left\{q_{I_A}, q_f\right\}\right)\right\}\end{aligned}$$

Então $\mathcal{L}(I) = \mathcal{L}(A)^*$.

N.B. $q_I \neq q_f$ e $Q_A \cap \left\{q_I, q_f\right\} = \emptyset$

Seja $A = \left(Q_A, \Sigma, \delta_A, q_{I_A}, F_A\right)$ um AFD qualquer.

Seja $C = \left(Q_A, \Sigma, \delta_A, q_{I_A}, Q_A \setminus F_A\right)$ o autómato que se obtém de A trocando os estados finais com os não finais.

Então $\mathcal{L}(C) = \Sigma^* \setminus \mathcal{L}(A)$ é a linguagem complementar de $\mathcal{L}(A)$. Isto é

$$w \in \mathcal{L}(C) \Leftrightarrow w \notin \mathcal{L}(A).$$

N.B. O autómato A é um AF determinista, ao contrário das operações anteriores, que usam AFNDs.

Exemplo

Obtenha um autómato finito (não determinista) que reconheça a linguagem $\mathcal{L}\left((aaa)^*\right) \cup \mathcal{L}\left((aa)^*\right)$.

Um AFND para reconhecer $\mathcal{L}\left(aa\right)$ é

δ_1	a	ε
q_0	$\{q_1\}$	$\emptyset$
q_1	$\{q_2\}$	$\emptyset$
q_2	$\emptyset$	$\emptyset$

Um AFND para reconhecer $\mathcal{L}\left((aa)^*\right)$ é

δ_2	a	ε
q_0'	$\emptyset$	$\left\{q_0, q_f'\right\}$
q_0	$\{q_1\}$	$\emptyset$
q_1	$\{q_2\}$	$\emptyset$
q_2	$\emptyset$	$\left\{q_0', q_f'\right\}$
q_f'	$\emptyset$	$\emptyset$

Exemplo (continuação)

Obtém-se um AFND para reconhecer $\mathcal{L}\left((aaa)^*\right)$ de forma semelhante:

δ_4	a	ε
p_0'	$\emptyset$	$\left\{p_0, p_f'\right\}$
p_0	$\{p_1\}$	$\emptyset$
p_1	$\{p_2\}$	$\emptyset$
p_2	$\{p_3\}$	$\emptyset$
p_3	$\emptyset$	$\left\{p_0', p_f'\right\}$
p_f'	$\emptyset$	$\emptyset$

Exemplo (continuação)

Um AFND para reconhecer a união $\mathcal{L}\left((aaa)^*\right) \cup \mathcal{L}\left((aa)^*\right)$ é:

δ_4	a	ε
r_0	$\emptyset$	$\left\{q_0', p_0'\right\}$
q_0'	$\emptyset$	$\left\{q_0, q_f'\right\}$
q_0	$\{q_1\}$	$\emptyset$
q_1	$\{q_2\}$	$\emptyset$
q_2	$\emptyset$	$\left\{q_0', q_f'\right\}$
q_f'	$\emptyset$	$\{r_f\}$
p_0'	$\emptyset$	$\left\{p_0, p_f'\right\}$
p_0	$\{p_1\}$	$\emptyset$
p_1	$\{p_2\}$	$\emptyset$
p_2	$\{p_3\}$	$\emptyset$
p_3	$\emptyset$	$\left\{p_0', p_f'\right\}$
p_f'	$\emptyset$	$\{r_f\}$
r_f	$\emptyset$	$\emptyset$

Exemplo (continuação)

Um AFND "feito à mão" para reconhecer $\mathcal{L}\left((aaa)^*\right) \cup \mathcal{L}\left((aa)^*\right)$ é

δ	a	λ
1	$\emptyset$	$\{2,4\}$
2	$\{3\}$	$\emptyset$
3	$\{2\}$	$\emptyset$
4	$\{5\}$	$\emptyset$
5	$\{6\}$	$\emptyset$
6	$\{4\}$	$\emptyset$

mas não está "bem preparado"... porquê?

Seja L uma linguagem regular e k o número de estados de um AFD que a reconheça.

Então qualquer palavra $p \in L$ com $|p| > k$ pode ser escrita como

$$p = uvw, \qquad \text{com } |uv| \leq k, |v| > 0$$

e $uv^n w \in L$ para cada $n \geq 0$.

Seja $L = \{a^n b^n \ : \ n \geq 0\}$.

Se L for uma linguagem regular existe um AFD que reconhece L. Seja k o número de estados desse autómato e $p = a^k b^k \in L$.

Qualquer decomposição de p nas condições do *pumping lemma* será

$$\begin{array}{ccc} u & v & w \\ a^j & a^m & a^{k-j-m} b^k \end{array}$$

com $j + m < k$ e $m > 0$.

Fazendo $n = 0$ (no *pumping lemma*) obtemos

$$uv^0 w = a^j \, (a^m)^0 \, a^{k-j-m} b^k = a^{k-m} b^k$$

Como $m > 0$, $k - m \neq k$. Mas as palavras de L têm igual número de as e bs portanto $a^{k-m} b^k \notin L$.

Conclui-se que L não é regular.