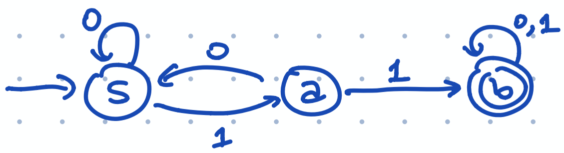
Q – states
$s \in Q$ – start
$A \subseteq Q$ – accepting
Σ – alphabet
$\delta: Q \times \Sigma \to Q$

$\delta^*: Q \times \Sigma^* \to Q$

$$\delta^*(q,w) = \begin{cases} q & \text{if } w=\varepsilon \\ \delta^*(\delta(q,a),x) & \text{if } w=ax \end{cases}$$

M accepts $w \iff \delta^*(s,w) \in A$

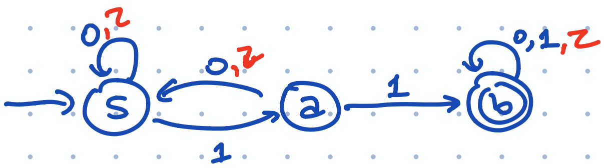
$$\boxed{L(M) = \{w \mid \delta^*(s,w) \in A\}}$$



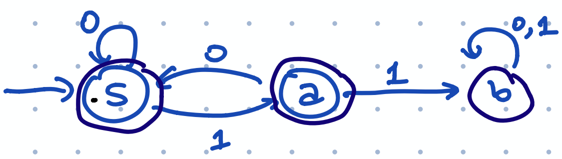
$(0+1)^*11(0+1)^*$

$\delta(s, 0010110)$

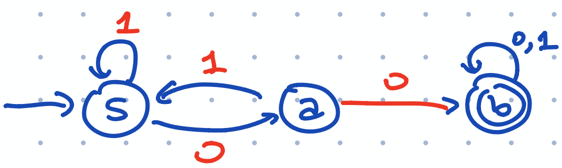
Deterministic finite-state automata



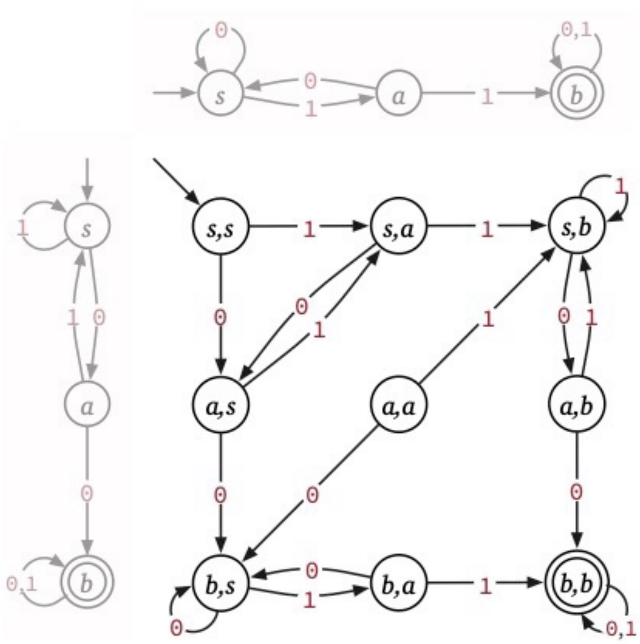
strings containing 11



strings not containing 11



strings containing 00



???
All strings that contain
either 00
$\boxed{\text{or}}$ 11
or both
???

strings containing 00 and 11

"Product construction"

Given $M_1 = (Q_1, s_1, A_1, \delta_1)$ and $M_2 = (Q_2, s_2, A_2, \delta_2)$ } over same Σ

Define $M = (Q, s, A, \delta)$

$$Q = Q_1 \times Q_2 \quad = \{(q_1, q_2) \mid q_1 \in Q_1 \text{ and } q_2 \in Q_2\}$$

$$s = (s_1, s_2)$$

$$A = \{(q_1, q_2) \mid q_1 \in A_1 \text{ and } q_2 \in A_2\}$$

$$\delta((q_1, q_2), a) = (\delta_1(q_1, a), \; \delta_2(q_2, a))$$

Theorem: $L(M) = L(M_1) \cap L(M_2)$

Key Lemma: $\delta^*((p,q), w) = (\delta_1^*(p, w), \delta_2^*(q, w))$ for all w, for all states $p \in Q_1$, $q \in Q_2$

Proof: Let w be an arb. string, p, q be arb. states

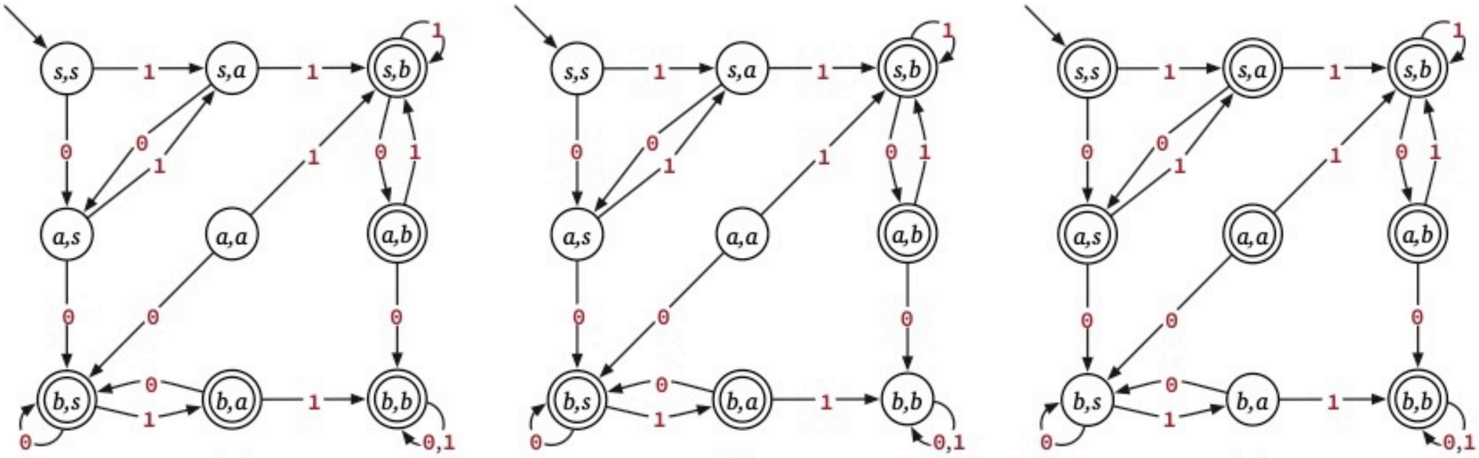
Assume $\delta^*((p', q'), x) = (\delta_1^*(p', x), \delta_2^*(q', x))$ for all strings x shorter than w and all states p', q'

Two cases:

- $w = \varepsilon \Rightarrow \delta^*((p,q), w) = \delta^*((p,q), \varepsilon) =$
$$= (p, q)$$
$$= (\delta_1^*(p, \varepsilon), \delta_2^*(q, \varepsilon))$$

- $w = ax = \delta^*((p,q), w) = \delta^*((p,q), ax)$
$$= \delta^*(\delta((p,q), a), x)$$
$$= \delta^*((\delta_1(p,a), \delta_2(p,a)), x)$$
$$= (\delta_1^*(\delta_1(p,a), x), \delta_2^*(\delta_2(q,a), x)) \quad \text{IH}$$
$$= (\delta_1^*(p, w) \quad \delta_2^*(q, w))$$

Therefore, $\delta^*((p,q), w) = (\delta_1^*(p, w), \delta_2^*(q, w))$



(a) OR (b) XOR (c) $00 \Rightarrow 11$

Closure properties of regular/automatic languages

If L_1 and L_2 are automatic, then so are

$L_1 \cap L_2$

$L_1 \cup L_2$

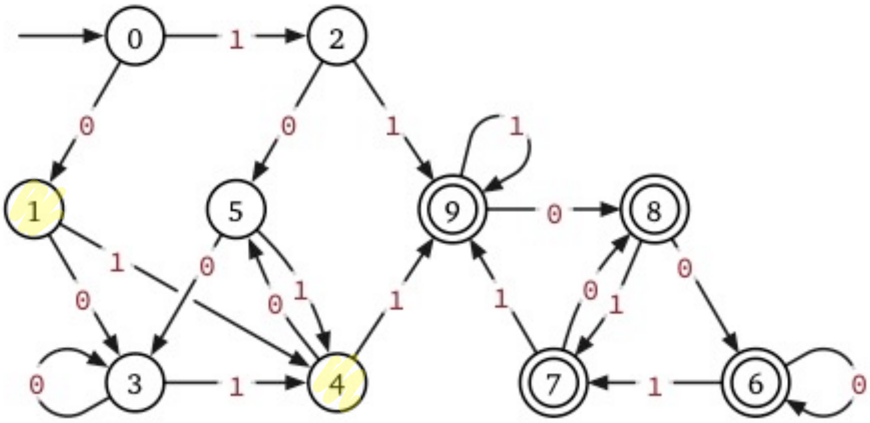
$L_1 \setminus L_2$ $\overline{L_2} = \Sigma^* \setminus L_2$

$L_1 \oplus L_2$

Kleene's Theorem: regular = automatic

If L_1 and L_2 are regular, then so are

$L_1 \cup L_2$ $L_1 \cdot L_2$ L_1^*



p and q are distinguishable: some further input $\nearrow$ acc, $\searrow$ not acc

$0 \xrightarrow{0} 1 \xrightarrow{0} 3$, $1 \xrightarrow{1} 4$

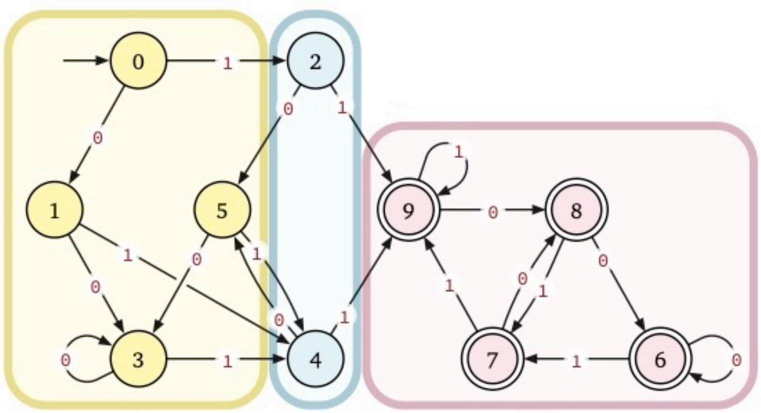
$0 \xrightarrow{1} 2 \xrightarrow{0} 5$, $2 \xrightarrow{1} 9$

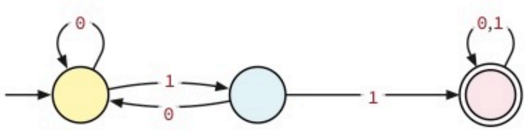
$3 \rightarrow 3 \rightarrow 3$, $3 \rightarrow 4$

$3 \rightarrow 4 \xrightarrow{0} 5$, $4 \xrightarrow{1} 9$

$0 \equiv 3$

$1 \xrightarrow{1} 4$ $\qquad$ $4 \xrightarrow{1} 9$





$L = \{0^n 1^n \mid n \geq 0\} = \{\varepsilon, 01, 0011, 000111, 00001111, \ldots$

$x = 0^i 1$ Let $p = \delta^*(s, x)$

$y = 0^j 1$ $q = \delta^*(s, y)$

$z = 1^{i-1}$ $\delta^*(p, z) =$ accepting!

$\delta^*(q, z) =$ <u>not</u> accepting!

Every string in <u>0^*1</u> leads to a different state of our DFA

infinite

<u>finite</u>

<u>Fooling set</u> for L is a language F s.t.

$$\boxed{\begin{array}{l}\text{for all } x, y \in F \text{ where } x \neq y \\ \quad \text{there is } z \in \Sigma^* \\ \quad\quad \text{s.t. } xz \in L \ \underline{\underline{\text{xor}}} \ yz \in L\end{array}}$$