LECTURE - 4

Recap

DFA M : Q - states

$s \in Q$ - start state

$A \subseteq Q$ - accepting states

Σ - alphabet

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

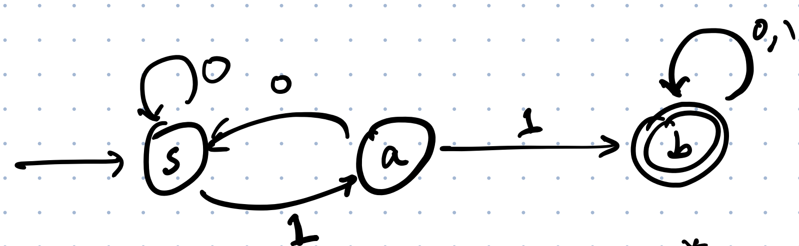
EXTENDED $\delta^* : Q \times \Sigma^* \to Q$

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

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

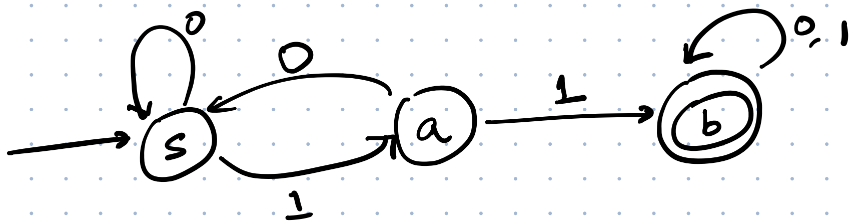
$L(M) = \{ w \mid \delta(s,w) \in A \}$

EXAMPLE :

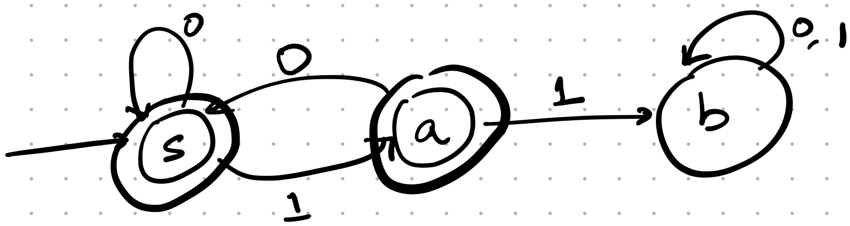


$(0+1)^* \, 11 \, (0+1)^*$

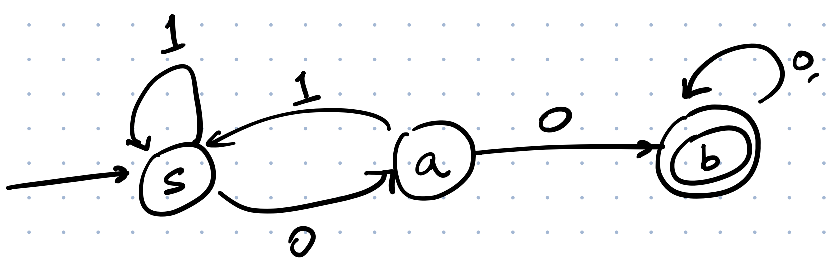
Eg string : 010110010



strings containing 11



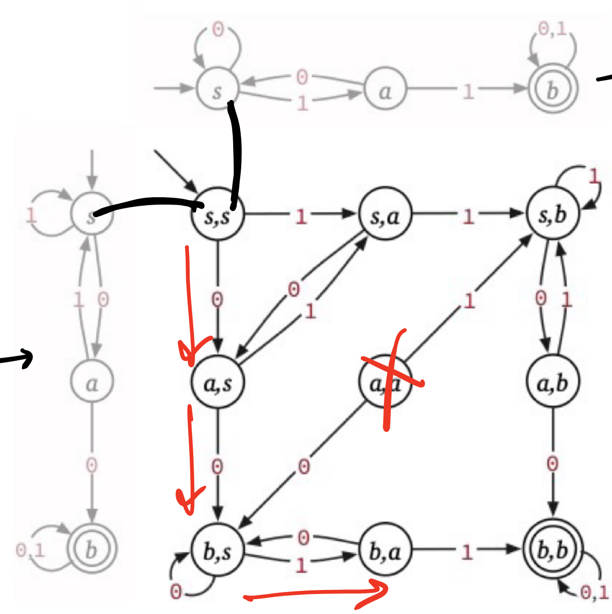
strings not containing 11



strings containing 00

$\delta^*((s,s), 001) \to (b,a)$

strings containing 11



strings containing 00

Product construction strings containing 00 and 11

$$M_1 = (Q_1, s_1, A_1, \delta_1)$$
$$M_2 = (Q_2, s_2, A_2, \delta_2)$$

over the same Σ

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

$$Q = Q_1 \times Q_2 = \{(q_1, q_2) : q_1 \in Q_1, q_2 \in Q_2\}$$

$$s = (s_1, s_2)$$

$$A = \{(q_1, q_2) \mid q_1 \in A_1 \ \boxed{\quad}\ q_2 \in A_2\}.$$

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

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 arbit. string, p, q be arbit. states.

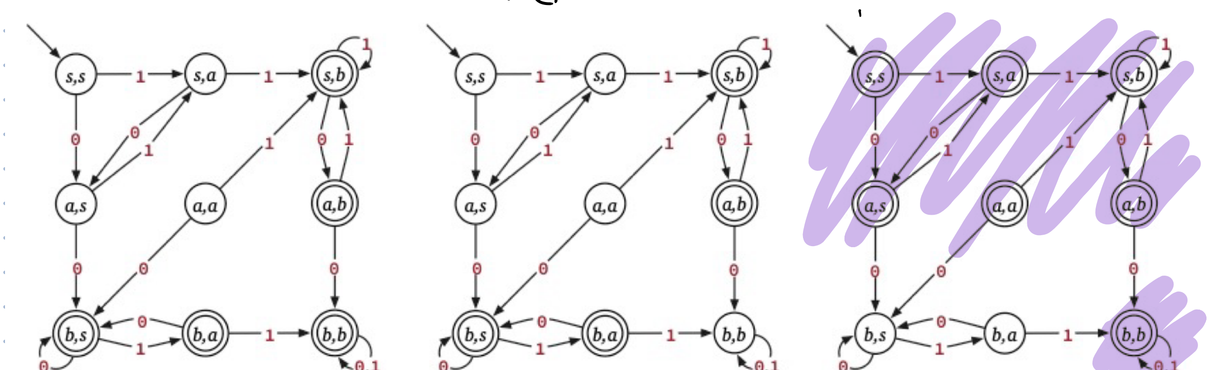
Assume: $\delta^*((p',q'),x) = \delta_1^*(p',x), \delta_2^*(q',x)$
for all strings x shorter than w,
all $p' \in Q_1$, $q' \in Q_2$.

Case 1: $w = \varepsilon$.

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

Case 2: $w = a \cdot x$.

$$\delta^*((p,q),w)$$
$$= \delta^*((p,q), a \cdot x)$$
$$= \delta^*(\delta((p,q),a), x) = \delta^*((\delta_1(p,a), \delta_2(q,a)), x)$$
$$= (\delta_1^*(\delta_1(p,a),x), \delta_2^*(\delta_2(q,a),x))$$
IH where $p' = \delta_1(p,a)$, $q' = \delta_2(q,a)$
$$= (\delta_1^*(p,w), \delta_2^*(q,w))$$



(a) (b) (c)

00 OR 11 OR BOTH

XOR (00 or 11 but not both)

$00 \Rightarrow 11$ (if 00, then 11) (+ accept all strings without 00)

Kleene's Theorem :

Every regular language is automatic, and every automatic language is regular.

CLOSURE PROPERTIES

By def. of regular,

automatic

If L_1 & L_2 are = regular

so are $L_1 \cup L_2$ $L_1 \circ L_2$ L_1^*

If L_1 and L_2 are automatic, so are

$L_1 \cap L_2$ [Product construction with AND accepting]

$L_1 \cup L_2$ [" OR "]

$L_1 \setminus L_2$ [" \ "]

$L_1 \oplus L_2$ [" $\oplus$ "]

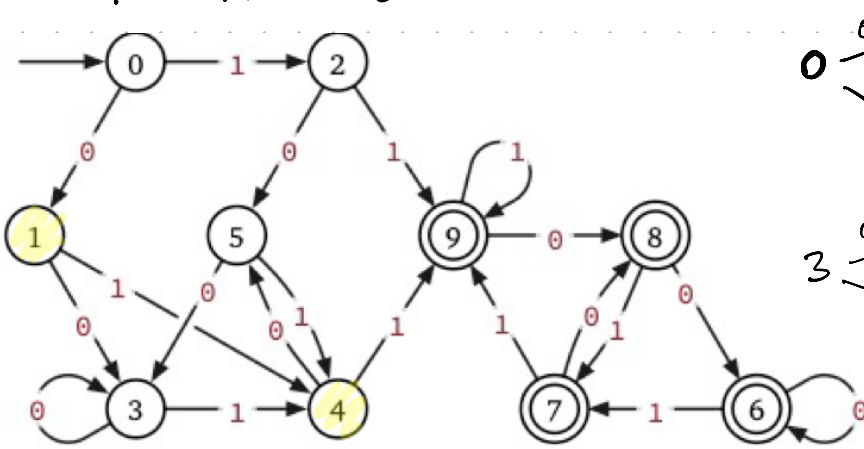
$\Sigma^* \setminus L_1$ [" with accept and reject states swapped]

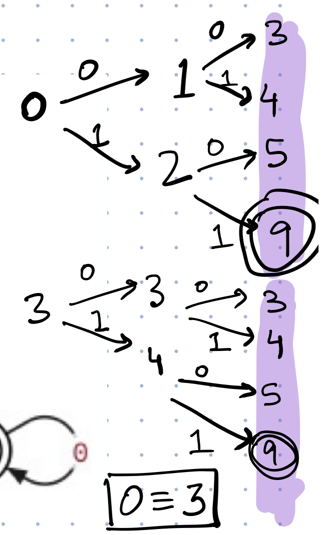
EXAMPLE OF A LANGUAGE THAT IS <u>NOT</u> REGULAR.

$$L = \{0^n 1^n \mid n \geq 0\}.$$

How should we prove that a language is NOT regular?

DISTINGUISHABLE STATES.





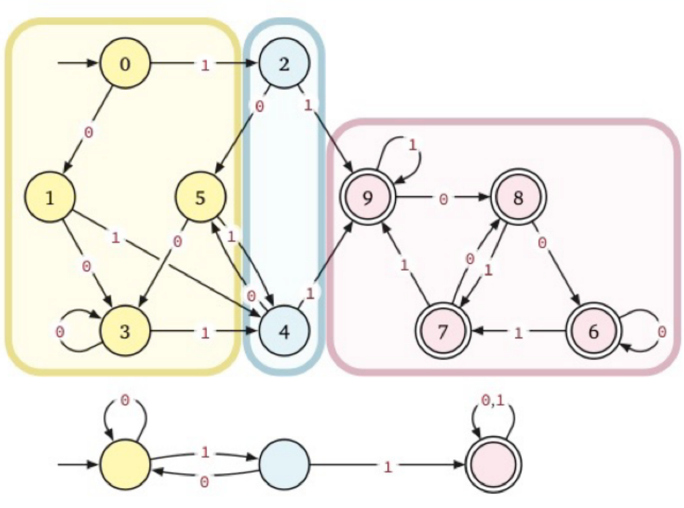
$\boxed{0 \equiv 3}$

(p, q) are distinguishable if there is a string w

s.t. $\delta^*(p, w) \notin$ ACCEPT

BUT $\delta^*(q, w) \in$ ACCEPT

or vice-versa.



FOOLING SETS.