January 30, 2024 10:01 AM

Nondeterministic Finite Automata (NFA)

- allow choices (and ε-transitions)

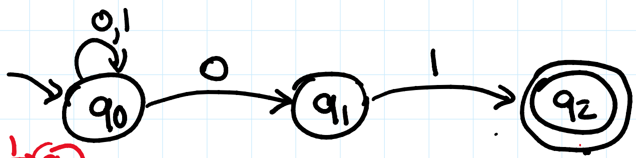
Ex1 all strings ending with 01

$(0+1)^*01$

e.g. 1001

$q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2$

$\xrightarrow{0} q_1$



not valid DFA !!

$\rightarrow \delta(q_0, 0) = q_0$ or q_1 ??

$\delta(q_1, 0)$

$\delta^*(q_0, 1001) = \{q_2, q_0\}$

but valid NFA

but what does it mean for an NFA to accept an input?

$\delta^*(q_0, 100) = \{q_0, q_1\}$

Idea - accept iff $\exists$ path from start state to some accepting state

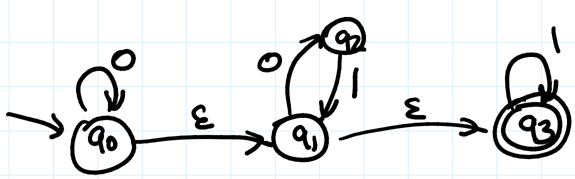
$\neg\exists \equiv \forall\neg$

reject iff $\forall$ path from start state, end in non-accept state

not realistic machine!

ability to guess

Ex2 $0^*(01)^*1^*$



ε-transitions don't consume input

Formal Def'n An NFA is $M = (Q, \Sigma, s, \delta, A)$

states

accept states $A \subseteq Q$

like before

except $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$

except $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \to \underbrace{2^Q}_{\text{power set}}$ $\quad \mathcal{P}(Q)$

= set of all subsets of Q

Ex1: $\delta(q_0, 0) = \{q_0, q_1\}$
$\delta(q_1, 0) = \emptyset$

Ex2: $\delta(q_0, \varepsilon) = \{q_1\}$

Def For $q \in Q$, define $\underline{\varepsilon\text{-reach}(q)}$ recursively:

(i) q is in ε-reach(q)
(ii) if q' is in ε-reach(q) & $q'' \in \delta(q', \varepsilon)$,
then q'' is in ε-reach(q)
(iii) nothing else is in.

Ex2: $\varepsilon\text{-reach}(q_0) = \{q_0, q_1, q_3\}$
$\varepsilon\text{-reach}(q_1) = \{q_1, q_3\}$.

Def Define extended transition fn $\delta^*: Q \times \Sigma^* \to 2^Q$ recursively:

(i) $\delta^*(q, \varepsilon) = \varepsilon\text{-reach}(q)$

(ii) if $x = ay$ $\quad (a \in \Sigma, y \in \Sigma^*)$, $\quad \forall q \in Q, x \in \Sigma^*$

$$\delta^*(q, x) = \bigcup_{q' \in \varepsilon\text{-reach}(q)} \ \bigcup_{q'' \in \delta(q', a)} \underline{\delta^*(q'', y)}$$

$\delta(q_0, 0) = \{q_0, q_1\}$

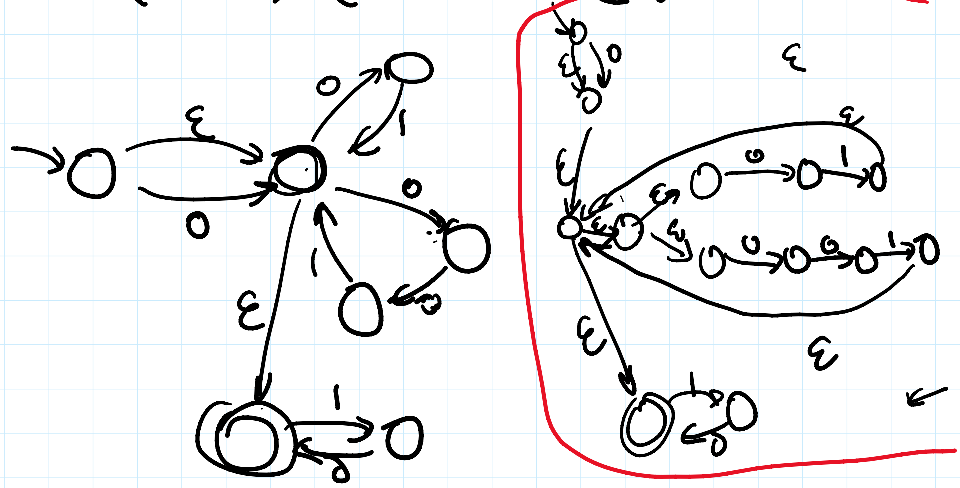
Ex1: $\delta^*(q_0, 100) = \delta^*(q_0, 00) = \delta^*(q_0, 0) \cup \delta^*(q_1, 0)$
$= \{q_0, q_1\} \cup \emptyset$
$= \{q_0, q_1\}$

Def M $\underline{\text{accepts}}$ x iff $\delta^*(s, x) \cap A \neq \emptyset$.

$L(M) = \{x \in \Sigma^* : M \text{ accepts } x\}$.

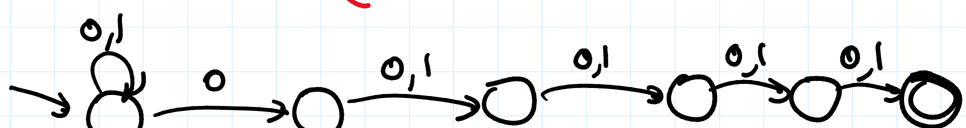
Ex a)

Ex a) $(\varepsilon + 0)(01 + 001)^*(10)^*$

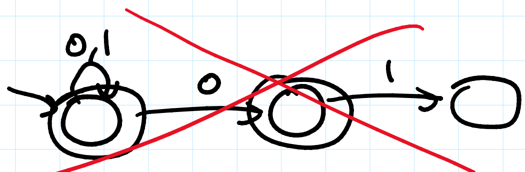


b) all strings whose 5^{th} last symbol is 0

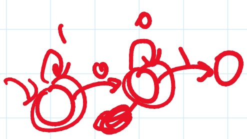
(DFA with 32 states)



c) all strings not ending with 01



WRONG.



Regular
↓
NFA
↓
DFA

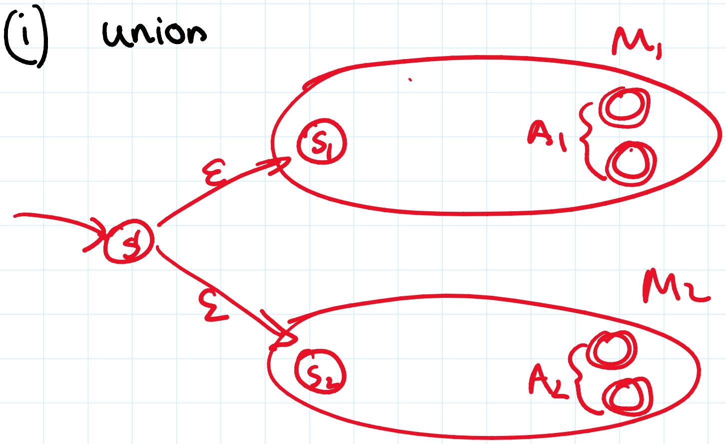
Regular → NFA

Thm If L_1 is accepted by some NFA M_1,
L_2 " " " " " M_2,

Thm If L_1 is accepted by some NFA M_1,
L_2 " " " " " M_2,
then (i) $L_1 \cup L_2$ is accepted by some NFA M'
(ii) $L_1 \cdot L_2$ " " " " "
(iii) L_1^* " " " " "

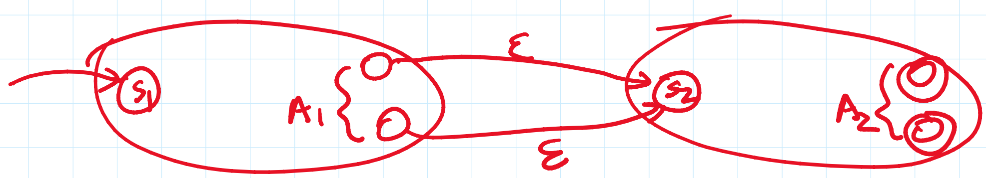
Pf: Given $M_1 = (Q_1, \Sigma, s_1, \delta_1, A_1)$
$M_2 = (Q_2, \Sigma, s_2, \delta_2, A_2)$,

(i) union

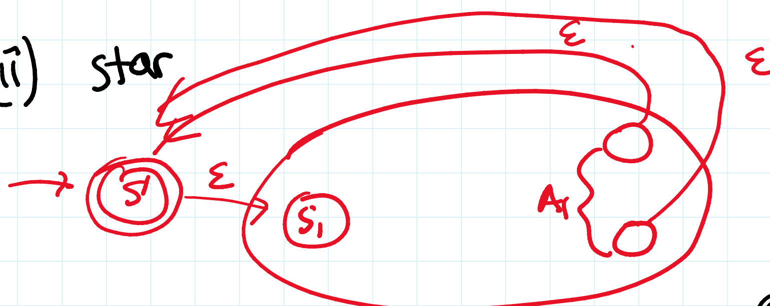


$Q' = Q_1 \cup Q_2 \cup \{s\}$
⋮

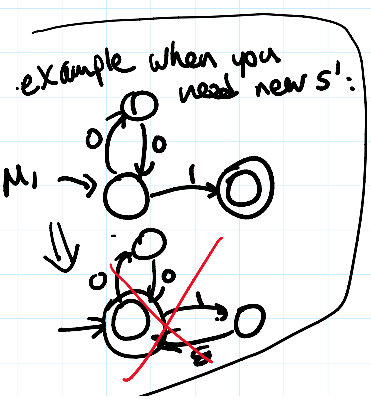
(ii) concat



(iii) star

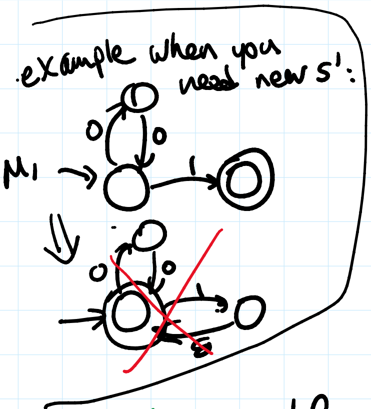


$Q' = Q \cup \{s'\}$
$\delta' = \ldots$



$Q' = Q \cup \{s'\}$

$\delta' = ---.$



Thm If L is regular,
then L is accepted by some NFA.

Pf: By recursion / induction.

Base cases: ϕ, $\{\varepsilon\}$, $\{a\}$