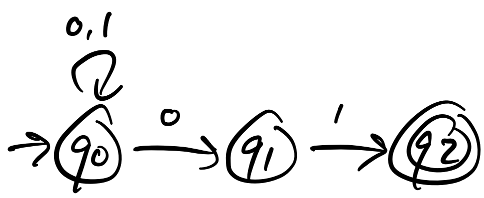
Comp 330 - Lecture 5 - September 14^{th} 2023

Def An NFA N is a 5-tuple $(Q, \Sigma, \Delta, S_0, F)$ where :

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

- Q : finite set of states
- Σ : input alphabet
- Δ : transition function
  $\Delta : Q \times \Sigma \to \underline{\underline{2^Q}} \to$ set of states
- Q_0 : set of start states
  $Q_0 \subseteq Q$
- F : set of accept states

Ex Suppose we have following NFA N



$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1\}$
$S_0 = \{q_0\}$
$F = \{q_2\}$
$\Delta(q_0, 0) = \{q_0, q_1\}$
$\Delta(q_2, 0) = \emptyset$

Extended transition function Δ^* for NFA

NFA $N = (Q, \Sigma, \Delta, S_0, F)$

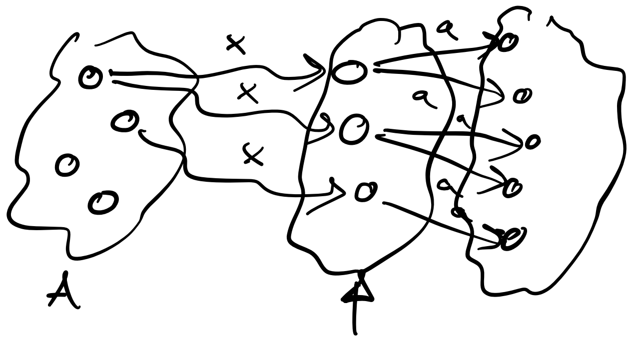
Def $\Delta^* : 2^Q \times \Sigma^* \to 2^Q$

$$A \subseteq Q$$

$$\Delta^*(A, \varepsilon) = A$$

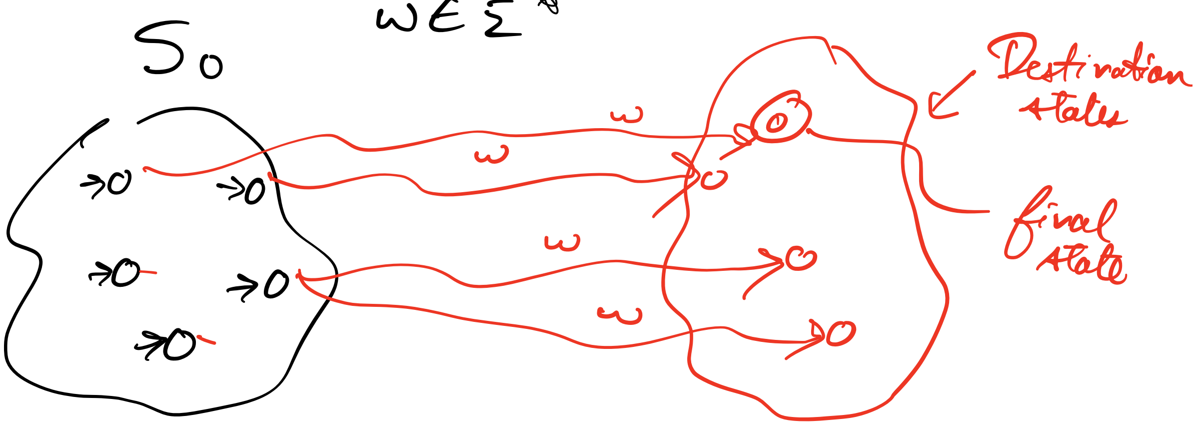
$$w \in \Sigma^*, \quad w = xa, \quad x \in \Sigma^*, \quad a \in \Sigma$$

$$\Delta^*(A, xa) = \bigcup_{q \in \Delta^*(A, x)} \Delta(q, a)$$



Language acceptance for NFA

$w \in \Sigma^*$

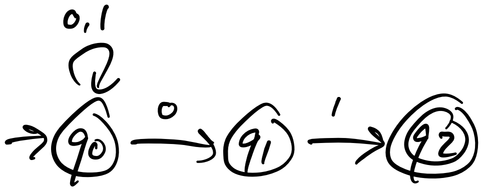


NFA N, $w \in \Sigma^*$

w is accepted by N $\iff$ $\underbrace{\Delta^*(S_0, w)}_{\text{destination states}} \cap \underbrace{F}_{\text{accept states}} \neq \emptyset$

$$L(N) = \{ w \in \Sigma^* : \Delta^*(S_0, w) \cap F \neq \emptyset \}$$

Ex



$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$

$S_0 = \{q_0\}$

$F = \{q_2\}$

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

$\Delta(q_2, 0) = \emptyset$

$\Delta^*(\{q_0\}, 001) = \{q_0, \underline{q_2}\}$ ↳ accept state

$\Rightarrow$ 001 is accepted

$\Delta^*(\{q_0\}, 00) = \{q_0, q_1\} \Rightarrow$ 00 is rejected.

Facts about Δ^* N, $A \subseteq Q$, $B \subseteq Q$, $x, y \in \Sigma^*$

① $\Delta^*(A, xy) = \Delta^*(\Delta^*(A, x), y) \rightarrow$ Proof in extra note

② $\Delta^*(A \cup B, x) = \Delta^*(A, x) \cup \Delta^*(B, x)$
↳ A2

Thm The family of languages accepted by DFA, $\mathcal{L}_{DFA}$, is exactly the same as the family of languages accepted by NFA, $\mathcal{L}_{NFA}$.

$$\mathcal{L}_{DFA} = \{ L(M) : M \text{ DFA} \}$$

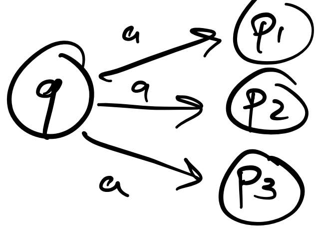
$$= \{ L(N) : N \text{ NFA} \} = \mathcal{L}_{NFA}$$

Proof (sketch)
① $\mathcal{L}_{DFA} \subseteq \mathcal{L}_{NFA}$.
② $\mathcal{L}_{NFA} \subseteq \mathcal{L}_{DFA}$.

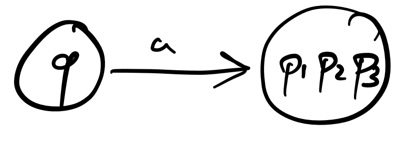
① $\mathcal{L}_{DFA} \subseteq \mathcal{L}_{NFA}$: A DFA is an NFA which does not use non-determinism.

② $\mathcal{L}_{NFA} \subseteq \mathcal{L}_{DFA}$: Given arbitrary NFA N, create an equivalent DFA M s.t. $L(N) = L(M)$

NFA N



DFA M



$\Delta(q, a) = \{p_1, p_2, p_3\}$ $\delta(\{q\}, a) = \{p_1, p_2, p_3\}$

Formal construction (Subset construction)

Given NFA $N = (Q_N, \Sigma_N, \Delta_N, S_N, F_N)$

Construct an equivalent DFA $M = (Q_M, \Sigma_M, \delta_M, s_M, F_M)$

$$Q_M := 2^{Q_N} \rightarrow \text{Powerset of } Q_N$$

$$\Sigma_M := \Sigma_N$$

$$s_M := S_N$$

$$F_M := \{ B \subseteq Q_N : B \cap F_N \neq \emptyset \}$$

$$\delta_M := Q_M \times \Sigma \rightarrow Q_M$$

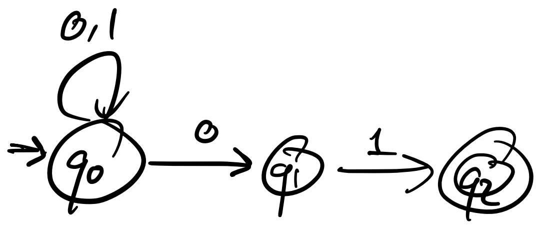
$$B \in Q_M \iff B \subseteq Q_N, \quad \sigma \in \Sigma$$

$$\delta_M(B, \sigma) := \bigcup_{q \in B} \Delta(q, \sigma)$$

Next step: $L(M) = L(N)$, post extra note.

Example of subset construction

N



$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$

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

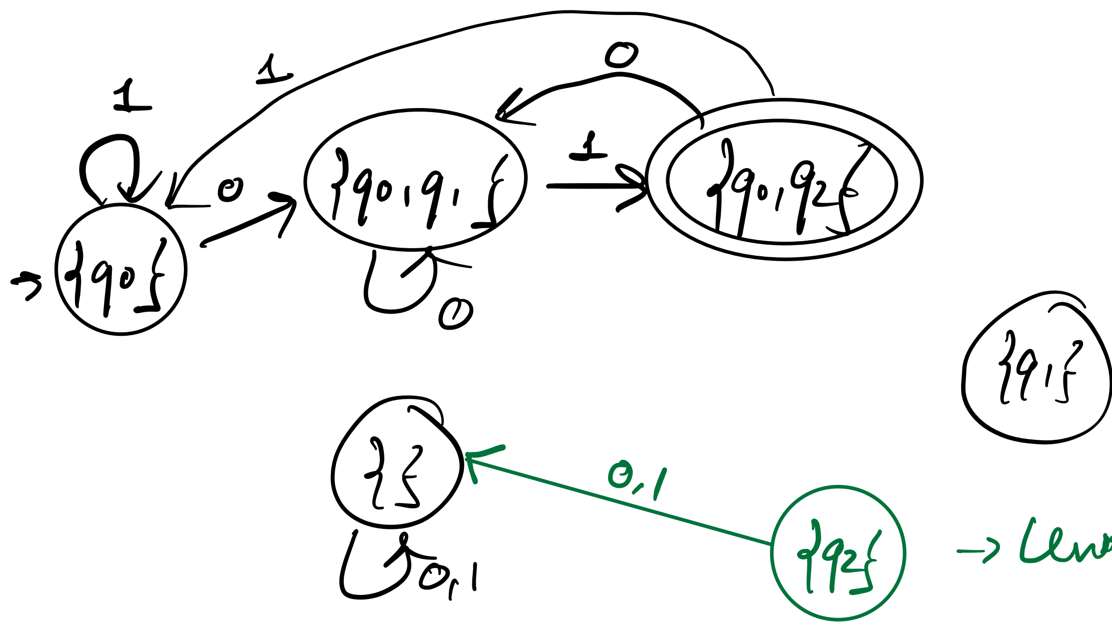
$\Delta(q_2, 0) = \phi$

$Q_0 = \{q_0\}$

$F = \{q_2\}$

$$\Delta^*(\{q_0, q_1\}, 0) = \{q_0, q_1\}$$

M via subset construction will have $2^3 = 8$ states. Unreachable states removed.



→ Unreachable

Ex : Design a DFA that accepts

$\{w \in \{0,1\}^* : w \text{ end in } 01\}$

Result $L_{DFA} = L_{NFA} = L_{REG}$

Implication: To prove L is regular
① Construct DFA M s.t. $L(M) = L$
NEW ② Construct NFA N s.t. $L(N) = L$

Exercise Σ, $L \subseteq \Sigma^*$, if L is finite then L is regular, AZ.

NFA + ε

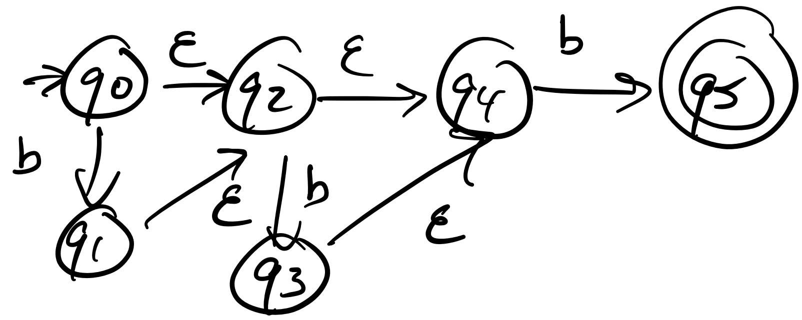
Def (Informal) An NFA + ε is an NFA which allows ε-transitions.

ε-transitions are transitions labeled with ε that allow the machine to change state without reading any letter from the input tape.

Ex (Dexter) $\Sigma = \{b\}$

NFA + ε

N



Input $w = b$

$\to q_0 \xrightarrow{b} q_1$, Fail

$\to q_0 \xrightarrow{b} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{\varepsilon} q_4$, Fail

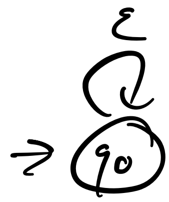
$\to q_0 \xrightarrow{\varepsilon} q_2 \xrightarrow{\varepsilon} q_4 \xrightarrow{b} q_5$, Success

b is accepted.

The acceptance conditions for NFA + ε are the same as NFA.

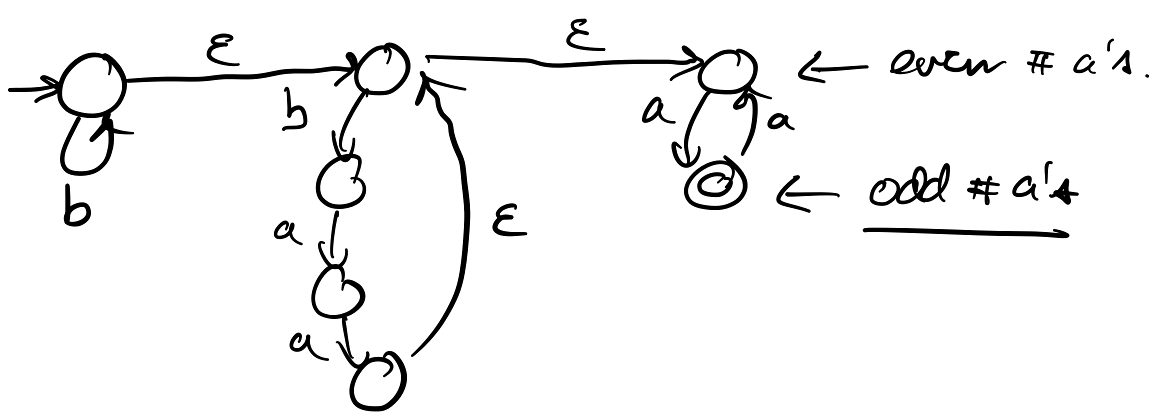
Exercise Check $L(N) = \{b, b^2, b^3\}$.

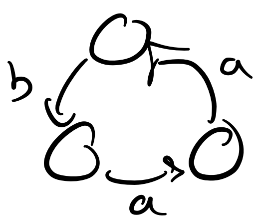
NFA + ε



Ex Design an NFA + ε which accepts

$$L = \{ b^n (baa)^m a^p : n, m, p \in \mathbb{N},\ p \text{ is odd} \}$$

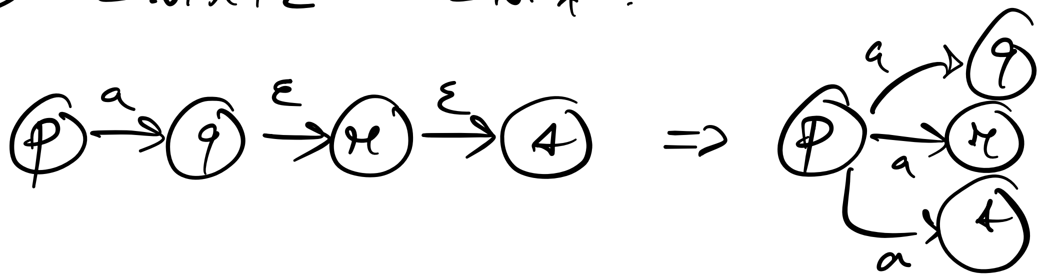




Thm $L_{NFA} = L_{NFA+\varepsilon}$

① $L_{NFA} \subseteq L_{NFA+\varepsilon}$: NFA are NFA + ε without any ε-transitions

② $L_{NFA+\varepsilon} \subseteq L_{NFA}$:



ε-closures / homomorphisms.

Result $L_{DFA} = L_{NFA} = L_{NFA+\varepsilon} = L_{REG}$

To prove that L is regular you

1) DFA
2) NFA
3) NFA+ε

NEW!

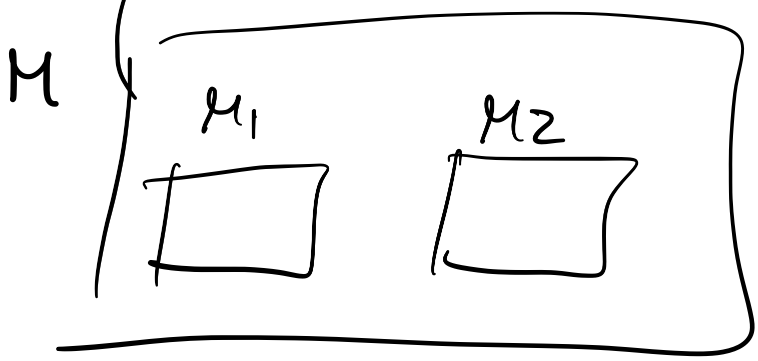
Ex Prove that the language is reg.

$$L = \{ (ab)^n (ba)^m : n, m \geq 0 \} \cup \{ (aba)^n : n \geq 0, \ n \text{ is odd} \}$$

Closure Properties of Reg Languages

Ex L_1, L_2 are regular then

$L_1 \cup L_2$ is also regular.



$L_1 \cup L_2$

Ex $L_1 \circ L_2$, L_1^*, $\overline{L_1}$, $L_1 \cap L_2$, ...