P.T. $P(n) = 1+2+3+ \cdots + n = \frac{n(n+1)}{2}$ —— (1)

<u>Basis</u>: $P(1) = 1 = \frac{1(1+1)}{2}$

<u>Hypothesis</u>: Let, assume $P(n)$ is true for $n=m$

$$\underbrace{P(m) = 1+2+ \cdots + m = \frac{m(m+1)}{2}}$$ —— (2)

<u>Induction</u>: $n = m+1$

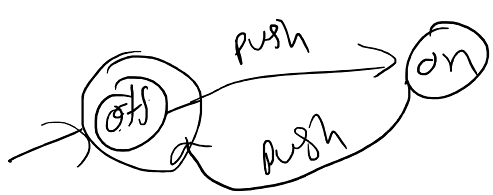
$$P(m+1) = \underbrace{1+2+3+ \cdots + m}_{(1)} + \underbrace{(m+1)} = \frac{m(m+1)}{2} + (m+1)$$

$$= \frac{(m+1)[(m+1)+1]}{2}$$

$P(n) \Rightarrow n=1$ true
$\Rightarrow n=m$ (assumption)
$\Rightarrow n=m+1$

$n = 1, m, m+1$
$\underbrace{1+1}=2, 2+1=3, 3+1=4, \cdots$

$$n(A \cup B \cup C) = n(A) + n(B) + n(C) - n(A \cap B) - n(A \cap C) - n(B \cap C) + n(A \cap B \cap C)$$



state relation

1) Input : push

2) output :

3) states : off, on

4) state relation : $\delta(\text{off}, \text{push}) = \text{on}$
$\delta(\text{on}, \text{push}) = \text{off}$

5) output relation :

$\lambda(\text{off}, \text{push}) = 0$

$\lambda(\text{on}, \text{push}) = 1$

$\lambda(q, a) = 0$ ← Mealy m/c

o/p function → $\lambda(q) = 0$ ← Moore m/c

state o/p

Alphabet: A nonempty finite set of symbols and represented by Σ.

Example: 1) $\Sigma = \{a, b, c, \ldots z\}$

2) $\Sigma = \{0, 1\}$ ← Binary alphabet

3) $\Sigma = \{a, b\}$

4) $\Sigma = \{0, 1, \ldots, 9\}$ ← decimal

5) $\Sigma = \{0, 1, \ldots 7\}$ ← octal

6) $\Sigma = \{0, 1, \ldots, 9, A, B, C, D, E, F\}$

× 7). $\Sigma = \phi$

8) $\Sigma = \{1, 2, 3 \ldots\}$

$\times$ 8) $\Sigma = \{1, 2, 3, ---\}$ ✓

Powers on Alphabet :

$\underbrace{\Lambda, \epsilon, \varepsilon, \lambda}_{\text{null string}}$

$\Sigma = \{a, b\}$

$\Sigma^0 = \{\epsilon\}$

$w = a\underline{bb}ba$ $|\epsilon| = 0$

$|w| = \cancel{5}$

$\Sigma^1 = \{a, b\}$

$\Sigma^2 = \{aa, ab, ba, bb\}$

$|a| = 1$, $|b| = 1$

$\Sigma^3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

Kleene's

$n \Rightarrow 2^n \leftarrow$

Kleene's star closure Σ^*

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup ---$

$= \{\epsilon\} \cup \{a, b\} \cup \{aa, ab, ba, bb\} \cup \{ ---\} \cup ---$

$= \{\epsilon\} \cup \{a, b\} \cup \{aa, ab, ba, bb, aaa, aab, ---\}$

$\Sigma^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, aab, ---\}$

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

$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, ---\}$

closure

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup ---$

$= \{a, b\} \cup \{aa, ab, ba, bb\} \cup ---$

$= \{a, b, aa, ab, ba, bb, ---\}$

$\Sigma^* = \Sigma^+ \cup \{\epsilon\}$

$\Sigma^+ = \Sigma^* - \{\epsilon\}$

String A string is a finite sequence of symbols chosen from a alphabet.

$\Sigma = \{a, b, c\}$

$\Sigma = \{a, \ldots$

$\Sigma = \{a, b\}$

$x = abbab$ ✓ , $y = bba$ ✓ $z = bbac$

$|x| = 5$ $|y| = 3$

Length of a string : No of characters present in the string

Empty string (null string) : A string with length zero.

$\underbrace{\epsilon, \varepsilon, \lambda, \Lambda}$ $|\varepsilon| = 0$

ϵ λ $\vee$

$\wedge$ $\wedge$

Concatenation of strings

Let x, y are any two strings then

xy is the concatenation of x & y.

e.g. $x = abb$, $y = babb$

$xy = \underbrace{abb}\underbrace{babb}$ $|xy| = |x| + |y|$

$= 3 + 4 = 7$

concatenation is not a commutative but associative

$\epsilon x = x\epsilon = x$

$\epsilon \leftarrow$ identity element for concatenation

$yx = babbabb$

$xy \neq yx$

$|xy| = |yx|$

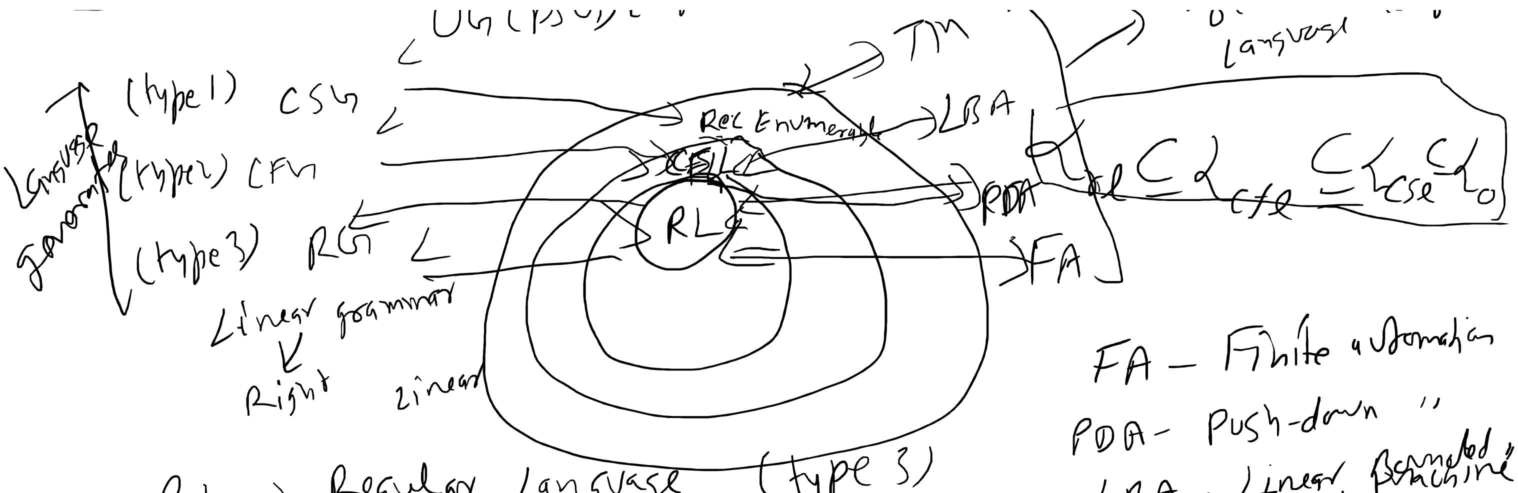
$2 + 0 = 0 + 2 = 2$

$5 \times 1 = 1 \times 5 = 5$

Chomsky Hierarchy of Languages

UG (PSG) (type 0) TM Language recognizers language acceptors

(type 1) CSG LBA



RL → Regular Language (type 3)

CFL → Context-free Language (type 2)

CSL → Context - sensitive Language (type 1)

Recursively Enumerable (type 0)

FA – Finite automation

PDA – Push-down "

LBA – Linear Bounded Machine

TM – Turing Machine

UG – Unrestricted grammar

PSG – Phrase Structured grammar

CSG – Context Sensitive grammar

CFG – Context-free grammar

RG – Regular grammar

Grammar: Past Simple Tense

S + V-II + Ob + other

I ate two mangoes yesterday.
(I = S, ate = V-II, two mangoes = Ob, yesterday = other)

$L = \mathcal{L}$

$L(G)$

Grammar: $G = (V_H, \Sigma, P, S)$ (T above Σ)

$S \in V_H$ -

V_H – Non empty finite set of variables / non-terminals

Σ – Nonempty finite set of terminals

Sentence → Noun Verb Adverb

Noun → Raman | Sita | Gita

P: $\left[\begin{array}{l} \text{Sentence} \rightarrow \text{Noun Verb Adverb} \\ \text{Noun} \rightarrow \text{Mohan} \mid \text{Sita} \mid \text{Gita} \\ \text{Verb} \rightarrow \text{sang} \mid \text{ate} \mid \text{ran} \\ \text{Adverb} \rightarrow \text{fast} \mid \text{well} \end{array} \right]$

$\alpha \rightarrow \beta$

S = Sentence ⇒ Noun Verb Adverb
⇒ Mohan Verb Adverb
⇒ Mohan sang Adverb
⇒ Mohan sang well = w

$L(G) = \{ w \in \Sigma^* \mid S \overset{*}{\underset{G}{\Rightarrow}} w \}$

V_N = { Sentence, Noun, Verb, Adverb }

Σ = { Mohan, sita, Gita, sang, ate, ran, fast, well }

P: set of productions

S: Sentence

$L \subseteq \Sigma^*$ ⇒ Any subset of Σ^* is a language.

$xy \neq yx$ (not commutative)

Σ^* is also a language

$x(yz) = (xy)z$

$A \subseteq A \Rightarrow \Sigma^* \subseteq \Sigma^*$
↳ Σ^* is also a language.

$\boxed{x = x\varepsilon = \varepsilon x}$

Prefix, suffix & substring

Let w = abc is a string.

1) Prefix of w=abc are abc, ab, a, ε ✓
2) suffix of w=abc are abc, bc, c, ε ✓
3) substring of w=abc are w - (prefix) - (suffix)

abc - (ε) - (ε) = abc ✓

✓

$5+3$

$5(1+\frac{3}{5})$

5×1

$\underline{ab} + \underline{abb}$

$ab($ $)$

$ab\varepsilon + abb$

$ab(\varepsilon + b)$

$abc - (\varepsilon) - (\varepsilon) = abc$ ✓

$abc - (\varepsilon) - (c) = \underline{ab}$ ✓

$abc - (a) - (c) = b$ ✓

$ab \neq ba$

$\Rightarrow ab\varepsilon + ab$

$ab(\varepsilon + b)$

$\underbrace{5\times2} + \underbrace{2\times3}$

$\underline{a}\underline{bb} \neq b\underline{ab}$

$ab(b+b)$ ✗

$(ab+ba)b$

$\underline{ab} + \underline{ba}$ ✓

$2ab$ ✗

$3\times2 + 2\times3$

$2(2\times3)$

$L \subseteq \underbrace{\Sigma^{*}}$

Operations on Languages

1. Union : If L_1 & L_2 are the languages
   $L_1 \cup L_2 = \{w \mid w \in L_1 \text{ or } w \in L_2\}$
2. Intersection: $L_1 \cap L_2 = \{w \mid w \in L_1 \text{ and } w \in L_2\}$
3. Concatation : $L_1 L_2 = \{uv \mid u \in L_1 \text{ and } v \in L_2\}$
4. Complement : $\overline{L}$ or $L^c = \Sigma^{*} - L$

Powers on Languages

i) $L \Rightarrow L^0 = \{\varepsilon\}$

ii) $L^1 = L$

iii) $L^2 = L \cdot L$

(iv) $L^i = L^{i-1} L$

$\Downarrow$

L^4

$\Sigma^0 = \{\varepsilon\}$

$L_1^0 = \{\varepsilon\}$ $L_1 = \{ab, \underline{bb}, \underline{ba}\}$

$L_2 = \{ba, a\}$

$L_1 \cup L_2 = \{ab, bb, ba, a\}$

$L_1 \cap L_2 = \{ba\}$

$L_1 L_2 = \{abba, aba, bbba, bba, baba, baa\}$

$\overline{L_1} = \Sigma^{*} - L_1$

$\overline{L_2} = \Sigma^{*} - L_2$

(v) $L^{*} = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots$

$= \{\varepsilon\} \cup L \cup L \cdot L \cup L^2 \cdot L \cup \ldots$

$L_1^2 = L_1 \cdot L_1$

$= \{ab, \underline{bb}, \underline{ba}\}\{ab, bb, ba\}$

$= \{abab, abbb, abba, bbab, bbbb, bbba, \ldots$

baab, b abb, baba
{

$A = \{a, b, c\}$

$A \times B = \{(x, y) \mid x \in A \text{ and } y \in B\}$

$= \{(a,1), (a,2), (a,3), (b,1), (b,2), (b,3),$
$(c,1), (c,2), (c,3)\}$

$|A| = m, \ |B| = n$

$|A \times B| = mn$

$R \subseteq \underline{A \times B}$

$R = \{(x, y) \mid x \in A, y \in B \text{ and } xRy\}$

$= \{(1,2), (1,4), (1,8), (1,16),$
$(2,2), (2,4), (2,8), (2,16)$
$(4,4), (4,8), (4,16)\}$

$R \subseteq A \times A$

$R = \{(1,1), (1,2), (1,3), (1,4)$
$(2,2), (2,4), (3,3), (4,4)\}$

Reflexive relation $R = \{(a,a) \ \forall a \in A\}$

Symmetric " $R = \{(a,b) \in R \Rightarrow (b,a) \in R\}$

Asymmetric $R = \{(a,b) \in R \Rightarrow (b,a) \notin R\}$

Anti symmetric $R = \{(a,b) \in R, (b,a) \in R \Rightarrow a = b\}$

Transitive $R = \{(a,b) \in R, (b,c) \in R \Rightarrow (a,c) \in R\}$
$aRb, bRc \Rightarrow aRc$

Equivalence relation = reflexive, symmetric, transitive
partial order relation = reflexive, antisymmetric, transitive

Finite automation: $M = (Q, \Sigma, \delta, q_0, F)$

Q - Nonempty finite set of states

Σ - " " " of input symbol

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

$\delta: Q \times \Sigma \rightarrow Q$

Σ =

$\delta : \underbrace{Q \times \Sigma}{} \rightarrow Q$

↳ Transition function

$q_0 \in Q$: start state (initial)

F = (set of accepting states / final states)

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

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

$Q \times \Sigma = \{(\underline{q_0, 0}), (q_0, 1), (q_1, 0), (q_1, 1), (q_2, 0)$

$\delta(q_0, 0) = q_1$ ← mathematical

$(q_0) \rightarrow (q_1)$ ← diagram

Finite Automation

- FA without O/P
  - DFA
  - NFA (Nondeterminity)
  - ϵ-NFA (NFA with ϵ-moves)
- FA with O/P
  - Moore
  - Mealy

DFA (Deterministic Finite Automation)

(Deterministic finite automation) $\delta : \underbrace{Q \times \Sigma}{} \rightarrow \underbrace{Q}{}$

1) All inputs must be defined on all states
2) only single transition is allowed for every input on a state.

NFA (Non deterministic Finite Automation)

$\delta : Q \times \Sigma \rightarrow \underbrace{2^Q}{}$

ϵ-NFA (NFA with ϵ-moves)

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

$\xrightarrow{0} (q_2)$

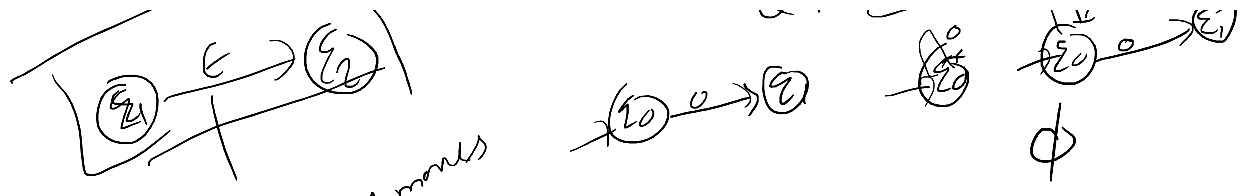
$(q_0) \xrightarrow{0} (q_1)$

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

$2^3 = 8$

$(q_0) \xrightarrow{0} (q_1)$

$\xrightarrow{0} (q_1)$



ε-moves

Moore m/c : $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ F

Δ – output alphabet

$\lambda : Q \to \Delta$

Mealy m/c : $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

$\lambda : Q \times \Sigma \to \Delta$

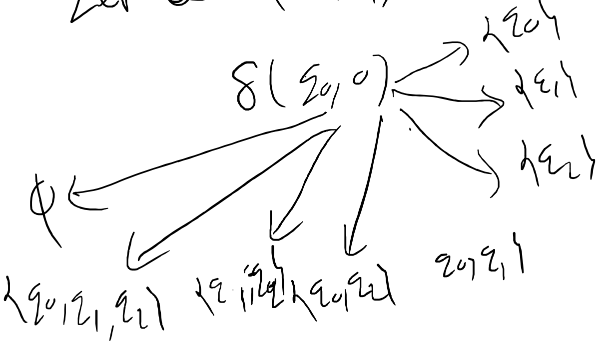
$\lambda(t) = Z(q(t), x(t))$ ← Mealy m/c

$\lambda(t) = Z(q(t))$ ← Moore m/c

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

$|Q| = n$

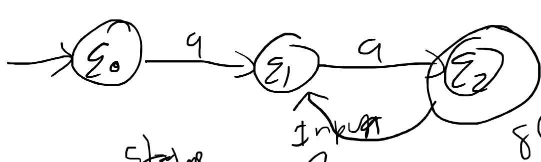
2^n



Q. Design a DFA to accept $L = \{a^{2n} \mid n \geq 1\}$

solution As $L = \{a^{2n} \mid n \geq 1\}$ → 0 $a^3 = aaa$

$= \{aa, aaaa, aaaaaa, \ldots\}$



← Transition diagram

$\delta(q_0, a) = q_1$, $\delta(q_1, a) = q_2$, $\delta(q_2, a) = q_1$

$|Q| = n = 3$
$= 1$

States	a
$\to q_0$	q_1

← Transition Table

$|Q| = n = 3$
$= 1$

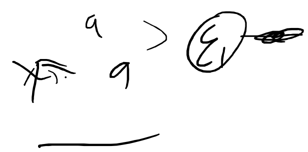
States	a
$\rightarrow q_0$	q_1
q_1	q_2
q_2	q_1

$\leftarrow$ Transition Table

$M = (\underbrace{\{q_0, q_1, q_2\}}_{Q}, \underbrace{\{a\}}_{\Sigma}, \delta, q_0, \{q_2\})$

Q. Design a DFA to accept $L = \{a^{2n} \mid n \geq 0\}$
$= \{\varepsilon, aa, aaaa, aaaaaa, \ldots\}$

Sol^n



aaa

δ	I/P a
$\rightarrow q_0$	q_1
q_1	q_0

$w = aaa$

$q_0 \longrightarrow$

$M = (\{q_0, q_1\}, \Sigma, \delta, q_0, \{q_0\})$

$\xrightarrow{a} q_0 \xrightarrow{a}$ or $\hat{\delta}(q_0, aaa) = \delta(\hat{\delta}(q_0, aa), a)$

$= \delta(\delta(\hat{\delta}(q_0, a), a), a)$

$= \delta(\delta(\delta(\underbrace{\hat{\delta}(q_0, \varepsilon)}, a), a), a)$

$= \delta(\delta(\underbrace{\delta(q_0, a)}, a), a)$

$= \delta(\delta(q_1, a), a)$

$= \delta(q_0, a)$

$\hat{\delta}(q_0, \underset{\uparrow w}{aaa}) \notin F \Rightarrow w = aaa$ is not accepted by M

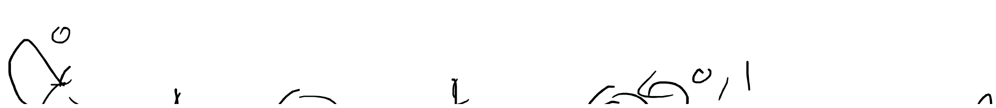
$w = 011101$

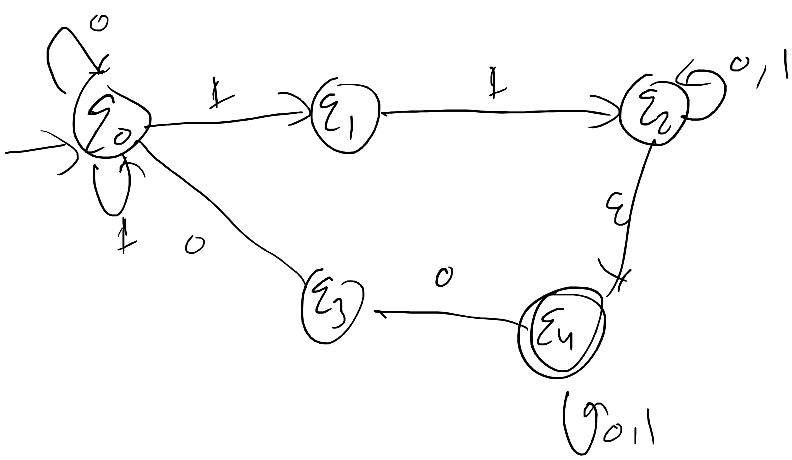
$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{1} q_2$

as $\hat{\delta}(q_0, 011101) \in F$
$\therefore$ 011101 accepted by M

NFA

$x = aw$





$x = aw$

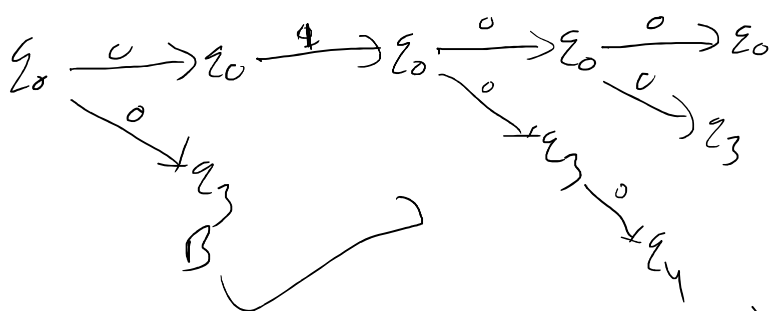
DFA

1) $\delta(q, \varepsilon) = q$

2) $\delta(q, wa) = \delta(\hat{\delta}(q, w), a)$

ii) $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$

$\hat{\delta}(q_0, 0100)$



$\hat{\delta}(q_0, 0100) = \{q_0, q_3, q_4\}$

acceptance of w: $\hat{\delta}(q_0, w) \cap F \neq \phi$ (NFA)

$\hat{\delta}(q_0, w) \in F$ (DFA)

The Language of FA

The Language of a ~~DFA~~ NFA

$L_{DFA} = \{w \mid \hat{\delta}(q_0, w) \in F\}$

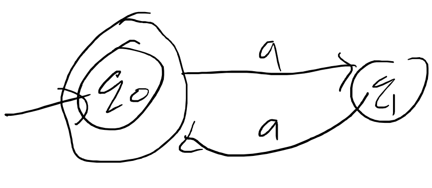
The Language of a NFA

$L_{NFA} = \{ w \mid \hat{\delta}(q_0, w) \cap F \neq \phi \}$

$\hat{\delta}(q_0, aaaa)$

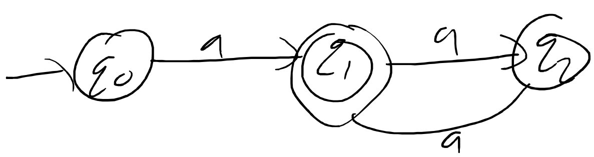
$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \in F$

Q: Design a DFA to accept $L = \{ a^{2n} \mid n \geq 0 \}$
$= \{ \varepsilon, aa, aaaa, \text{---} \}$



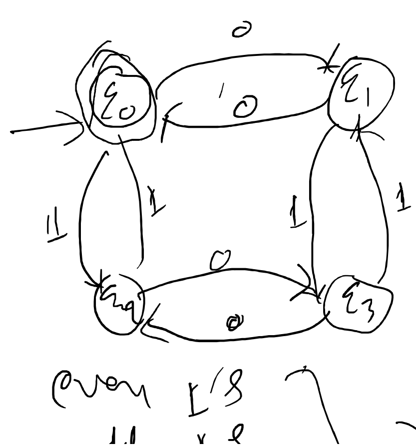
states	Input a
→ q_0	q_1
q_1	q_0

Q: Design a DFA to accept $L = \{ a^{2n-1} \mid n \geq 1 \}$
$= \{ a, aaa, aaaaa, \text{---} \}$



Q: Design a DFA to accept all strings of 0's & 1's which contain even 0's and even 1's.

Solution



0011 (even, even)

010 ✓ (even, odd)
011 ✓ (odd, even)
01 ✓ (odd, odd)

1) even 0's even 1's } only change in final state

1) even 0's even 1's
2) even 0's odd 1's
3) odd 0's even 1's
4) odd 0's odd 1's

→ only change in final s

$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_3 \xrightarrow{1} q_1$

$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_3 \xrightarrow{0} q_2$

$$\begin{aligned}\hat{\delta}(q_0, 1001) &= \hat{\delta}(q_1, 001) \\ &= \hat{\delta}(q_3, 01) \\ &= \hat{\delta}(q_2, 1) \\ &= \hat{\delta}(q_0, \epsilon) \\ &= q_0 \in F\end{aligned}$$

1001 accepted by M.

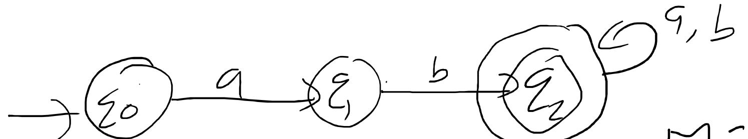
q_0 as final state for even, even

q_2 as final state for even, odd

q_1 as final state for odd, even

q_3 as final state for odd, odd

Q. Design a DFA to accept all the strings over {a, b} that starts with ab



$M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

state	Input a	b
$\rightarrow q_0$	q_1	-
q_1	-	q_2
$* q_2$	q_2	q_2

← Transition table for NFA

state		b
		ϕ
$\rightarrow [q_0]$		

$\leftarrow \delta'$

$M \leftarrow$ NFA

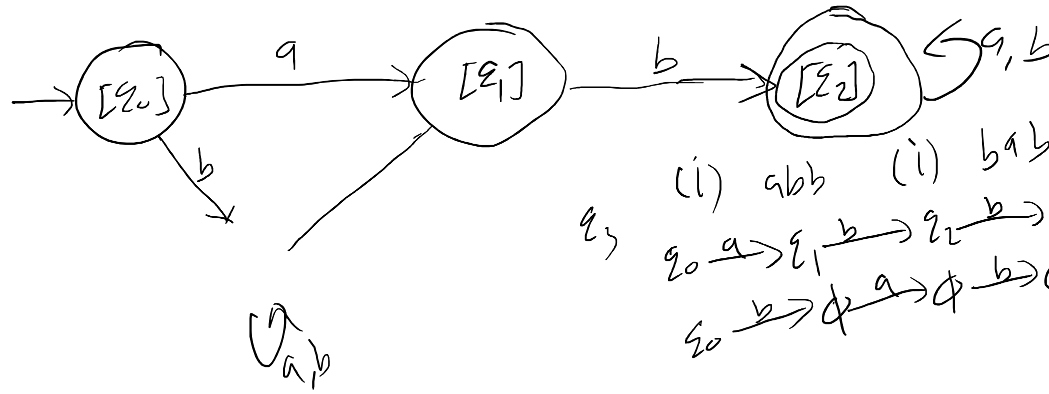
$M' \leftarrow$ DFA

→[q_0]
[q_1]   [q_2]
M→NFA *[q_2]   [q_2]   [q_2]
ϕ ϕ

← δ'

NFA DFA

$M' = (\{[q_0], [q_1], [q_2], \phi\}, \{a,b\}, \delta', [q_0], \{[q_2]\})$
$\quad Q \qquad \Sigma \qquad F$



(i) abb   (i) bab

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} q_2$

$q_0 \xrightarrow{b} \phi \xrightarrow{a} \phi \xrightarrow{b} \phi$

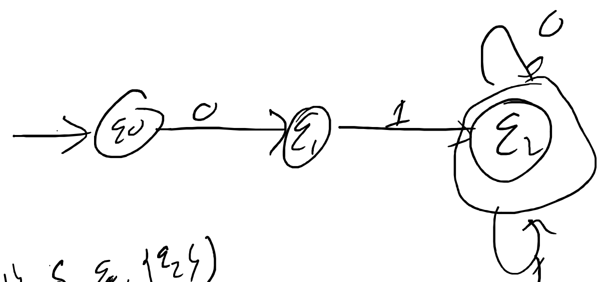
Q: Design a DFA to accept the language $L = \{x01y \mid x, y \in \{0,1\}^*\}$

OR

Design a DFA to accept all strings over $\{0,1\}$ which contain 01 as substring.

$\epsilon 01 \epsilon = 01$

$\boxed{\epsilon x = x\epsilon = x}$



$M = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$ ← NFA
$\quad Q \qquad \Sigma \qquad F$

Convert to DFA

States	Input 0	1
→ q_0	q_0, q_1	
q_1	–	
* q_2	q_2	

$A \cup \phi = A$

Transition table for DFA

States	Input 0	1
→ [q_0]	[q_0, q_1]	[q_0]

$\delta'([q_0], 0) = \delta(q_0, 0)$

States	0	1
$\rightarrow [q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_2]$
$* [q_0, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_2]$
$* [q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_2]$

$q_0' = [q_0]$

$\delta'([q_0], 0) = \delta(q_0, 0) = [q_0, q_1]$

$\delta'([q_0], 1) = \delta(q_0, 1) = [q_0]$

$\delta'([q_0, q_1], 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \phi = [q_0, q_1]$

$\delta'([q_0, q_1], 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = [q_0, q_2]$

$\delta'([q_0, q_2], 0) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \{q_2\} = [q_0, q_1, q_2]$

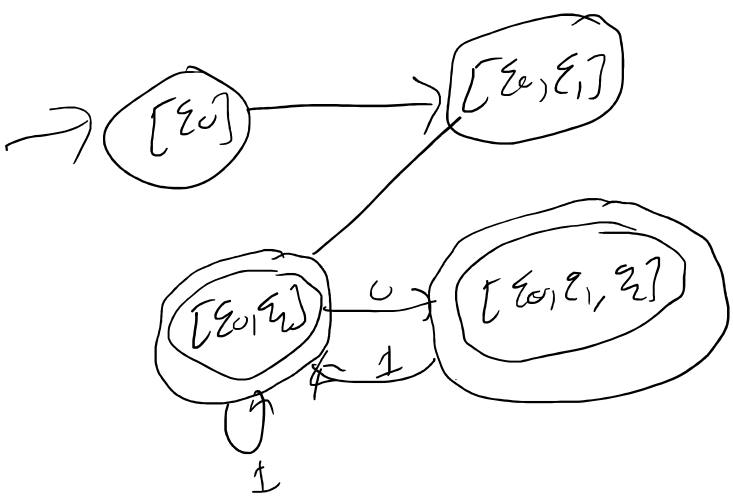
$\delta'([q_0, q_2], 1) = \delta(q_0, 1) \cup \delta(q_2, 1) = [q_0, q_2]$

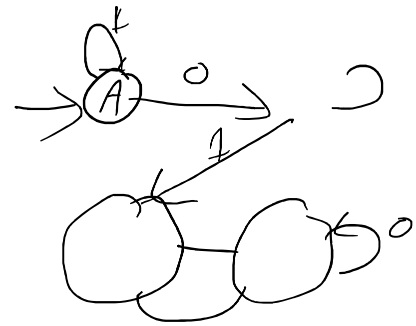
$[q_0] \rightarrow A$

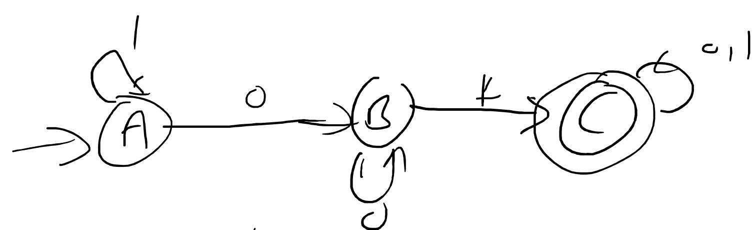
$[q_0, q_1] \rightarrow B$

$[q_0, q_2] \rightarrow C$

$[q_0, q_1, q_2] \rightarrow D$







$M' = (\{A, B, C\}, \{0, 1\}, \delta', A, \{C\})$

$[q_0] = A$

$\delta'(A, 0) =$

State 1

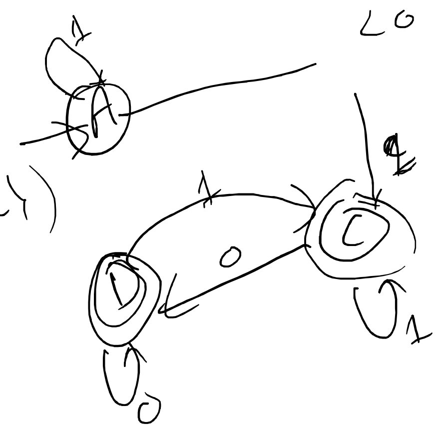
State	1
$\rightarrow A$	
ϕ	

$\delta'(A,0) =$

$\delta'([q_0],0) = [q_0, q_1$

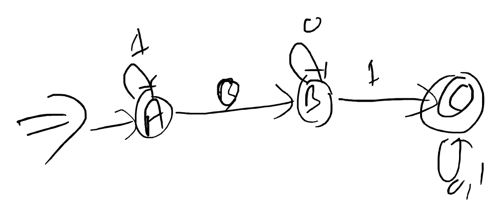
$\delta'([q_0, q_1],1) = [q_0, q_2]$ ↓ C

$D = [q_0, q_1, q_2]$



$M' = (\{A,B,C\},\{0,1\},\delta',A,\{C\})$

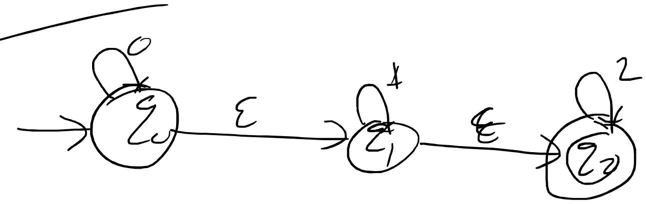
states	Input 0	1
$\rightarrow A$	B	A
B	B	C
$*C$	C	C



$\underline{100}$

$q_0 \quad A \xrightarrow{1} A \xrightarrow{0} B \xrightarrow{0} B \notin F'$

ε-NFA to NFA (NFA with ε-moves to NFA without ε-moves)



$\varepsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\} = \varepsilon^c(q_0)$

$\varepsilon\text{-closure}(q_1) = \{q_1, q_2\} = \varepsilon^c(q_1)$

$\varepsilon\text{-closure}(q_2) = \{q_2\} = \varepsilon^c(q_2)$

states	Input 0	1	2	ε
q_0	q_0	–	–	q_1
q_1	–	q_1	–	q_2

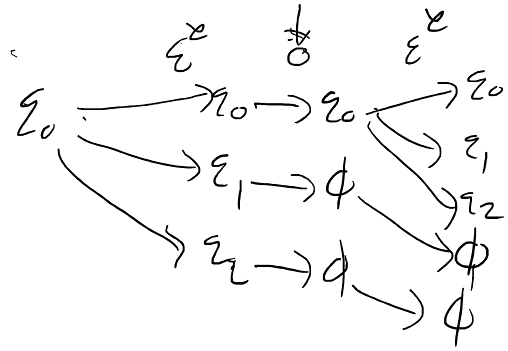
states	Input 0	1	2

states	Input 0	1	2
→ q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\{\ \}$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\{\ \}$	$\{\ \}$	$\{q_2\}$

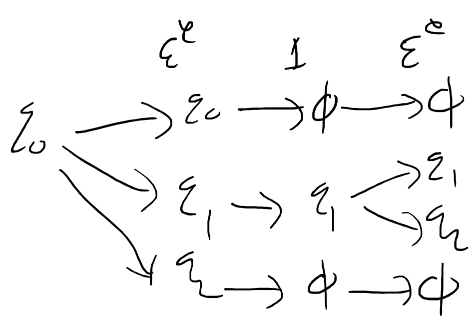
q_0	q_0	–	–	q_1
q_1	–	q_1	–	q_2
q_2	–	–	q_2	–

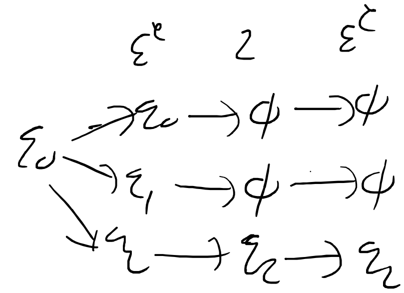
$\underbrace{\varepsilon\text{-closure}} = \varepsilon^e$

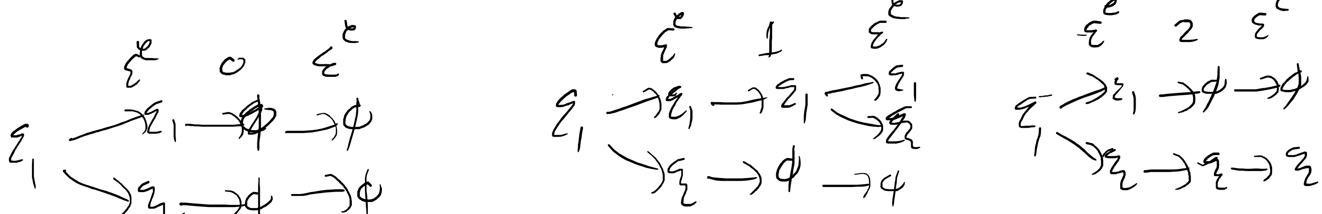
$\hat{\delta}(q_0, 0) = \varepsilon^e(\delta(\varepsilon^e(q_0), 0))$

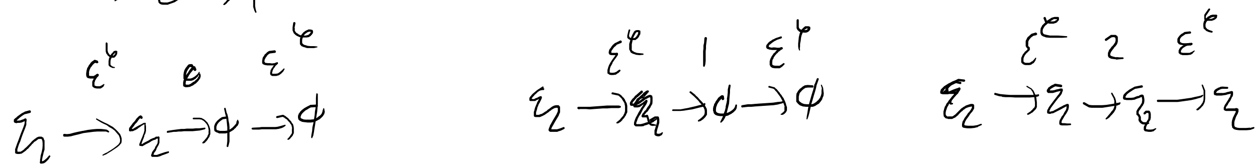


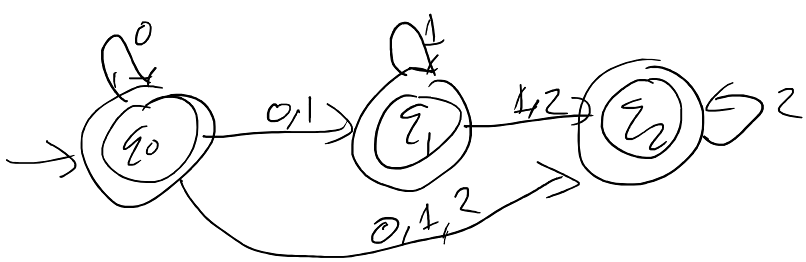
$A \cup \phi = A$











↑ NFA without ε-moves