Automata Theory

Theory of Computation

Theoretical Computer Science

- Computability Theory
- Complexity Theory
- Automata Theory

$\Sigma \rightarrow$ symbol for alphabet

$\Sigma = \{0,1\} \rightarrow$ binary alphabet set

$= \{0,1,\ldots,9\} \rightarrow$ decimal alphabet set

$= \{a,b\} \rightarrow$ Only a and b can be used in the language

Language

$L = \{a^n b^n \mid n \geq 0\}$ $\qquad \lambda$ = null string

for $n=0$ $L = \lambda$.

λ, ab, aabb, ...

$n=0$ $\quad n=1$ $\quad n=2$

Hence L is an infinite set, since there is no upper bound.

$L = \{0,1,2\} \rightarrow$ finite language

$L_1 = \{\underbrace{x \in \{0,1\}^*}_{} \mid \underbrace{x \text{ ends in } 1}_{1,11,01,001,0001,\ldots}\}$

x consists of 0s and 1s.

the * means all possible combinations/strings that can be constructed from the alphabets 0,1.

λ, 0, 1, 00, 11, 01, 10, ...

$L_2 = \{x \in \{0,1\}^* \mid x \text{ ends in } 0\}$

0, 10, 100, 110, ...

L_1 and L_2 are two <u>disjoint languages</u>.

Scanned by CamScanner

Operations Between Two Languages

1) $L = L_1 \cup L_2$

$= \{x \in \{0,1\}^* \mid x \text{ ends in } 0 \text{ and } 1\}$??

$\hookrightarrow \lambda$ is excluded here from $\{0,1\}^*$

$= \{x \in \{0,1\}^* \mid |x| \geq 1\}$

2) $L = L_1 \cap L_2$

$= \phi$

- If $L_3 = \{x \in \{0,1\}^* \mid x \text{ ends in } 01\} \rightarrow \overline{L_1}$

$\therefore L = L_1 \cap L_3$

$= L_3$

3) $L = L_1 - L_3$

$= L_1 \cap \overline{L_3}$

- $\overline{L_1} \cup \lambda = L_2$

4) $L = \{a^n b \mid n \geq 0\} \rightarrow b, ab, aab, aaab, \ldots$

$L^R = \{b a^n \mid n \geq 0\}$

Reverse of a Language

Grammer, $G = (V, \Sigma, P, S)$

Variable Set, Alphabets, Production, Start Symbol

- Using Capital letters, we denote the Variables.
- Using small letters, we denote the alphabets

Rules $\begin{cases} S \rightarrow ab \\ S \rightarrow aSb \end{cases}$

- First String Term $\rightarrow ab$

- $\Rightarrow aabb$

$\Rightarrow aaSbb$

a^3b^3

- Variables are for expanding
- Productions are the Rules
- $S \in V$
- There is only 1 start symbol

$L = \{a^n b^n \mid n \geq 1\}$ [$\because \lambda$ is not in my language]

Scanned by CamScanner

1) $G = (\{S\}, \{a\}, \{S \to aS, S \to \lambda\}, S)$

$G = (V, \Sigma, P, S)$

Find $L = \{\lambda, a, aa, aaa, \ldots\} = \{a^n \mid n \geq 0\}$

① $S \to aS$

$\Rightarrow a \quad [S \to \lambda]$

② $S \to aS$

$\Rightarrow aaS \quad [S \to aS$

$\Rightarrow aa \quad [S \to \lambda]$

2) $G = (\{S, A\}, \{a, b\}, \{S \to Ab, A \to aAb, A \to \lambda\}, S)$

Find $L = \{b, abb, aabbb, \ldots\}$

$= \{a^{n-1} b^n \mid n \geq 1\}$

$= \{a^n b^{n+1} \mid n \geq 0\}$

① $S \to Ab$

$\Rightarrow b \quad [A \to \lambda$

② $S \to Ab$

$\Rightarrow aAbb \quad [S \to aAb$

$\Rightarrow abb \quad [A \to \lambda$

③ $\Rightarrow aaAbbb \quad [A \to aAb$

$\Rightarrow aabbb \quad [A \to \lambda$

3) $G = (\{S, A\}, \{a, b\}, \{S \to aA, A \to bS, S \to \lambda\}, S)$

Find $L = \{\lambda, ab, (ab)^2, \ldots, (ab)^n\}$

① $S \to aA$

$\Rightarrow abS \quad [A \to bS$

$\Rightarrow ab \quad [S \to \lambda$

② $\Rightarrow abaA$

$\Rightarrow ababS \quad [A \to bS$

$\Rightarrow abab \quad [$

4) $G = (\{S, A, B\}, \{a\}, \{S \to Aa, A \to B, B \to Aa\}, S)$
Find L.

5) $G = \{ S \to abS \mid Sab \mid \lambda \}$, $L(G) = ?$

6) $S \to abS \mid Sba \mid \lambda$. Verify whether abbaabba can be generated or not?

7) $G = (\{S\}, \{a, b\}, \{S \to aSbS, S \to bSaS, S \to \lambda\}, S)$
Generate abbaba.

<u>Answers</u>

4.
$$S \to Aa$$
$$\Rightarrow Ba \quad [A \to B$$
$$\Rightarrow Aaa \quad [B \to Aa$$
$$\Rightarrow \ldots$$

$\therefore L = \phi$ (Ans)

5. $L = \{\lambda, ab, (ab)^2, \ldots\ldots, (ab)^n\} = \{(ab)^n \mid n \geq 0\}$

~~$S \to Sab$~~

(1) $S \to abS$

$\Rightarrow ab \quad [S \to \lambda$

(3)

$\Rightarrow ababS$

$\Rightarrow abab$

Scanned by CamScanner

6.

$S \to abS$

$\Rightarrow abSba$ [$S \to Sba$

~~$\Rightarrow abba$~~

$\Rightarrow abSbaba$

$\Rightarrow ababSbaba$? Rule

No Answer

7.

$S \to aSbS$

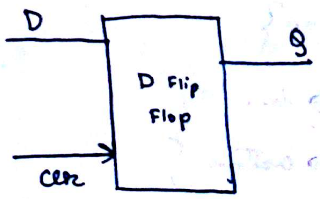
~~$\Rightarrow abSaSbb$~~ $\Rightarrow ab(bSaS)$ [$S \to \lambda$, $S \to bSaS$

~~$\Rightarrow abbaSbS \Rightarrow abb$~~

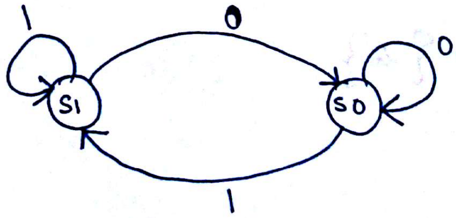
$\Rightarrow abbaSbSa$ [$S \to aSbS$; $S \to \lambda$

$\Rightarrow abbaba$ [$S \to \lambda$, $S \to \lambda$

Sequential Circuit



- Transition function is a mapping.



State Diagram

$f(Q \to Q)$

$Q = \{S0, S1\}$

$I = \{0, 1\}$

- For combinational circuits, state diagrams are not possible, since the output depends only on the inputs, not on the

• FSM (Finite State Machine)

Q → finite set of states

F → finite set of final states

S → a single start ~~start~~ state

δ → transition function

Σ → finite set of alphabet

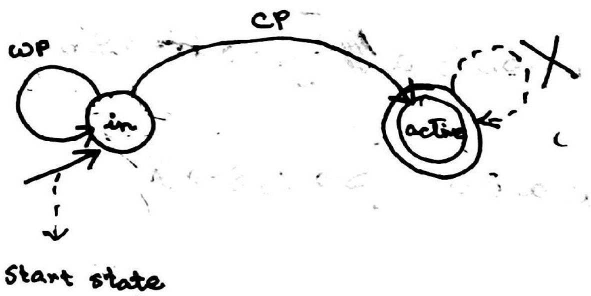
5 tuple → $(Q, \Sigma, \delta, F, S)$ → Finite Automata; Q is finite

Cell Phone

wp → wrong Password

cp → correct Password

in → inactive



Formal Representation of above State Diagram:

$Q = \{in, active\}$

$\Sigma = \{wp, cp\}$

$\delta = \left. \begin{matrix} in \xrightarrow{wp} in. \\ in \xrightarrow{cp} active \end{matrix} \right\}$

$F = \{active\}$

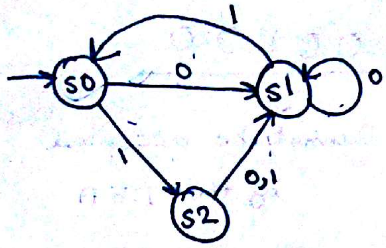
$S = \{in\}$

Scanned by CamScanner

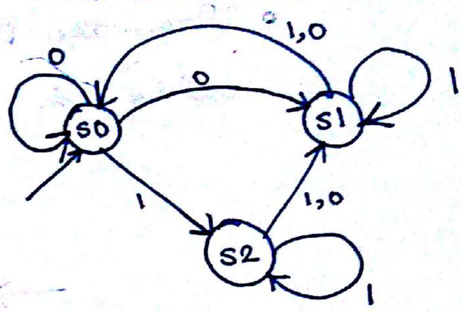
Finite Automata

Deterministic FA (DFA)

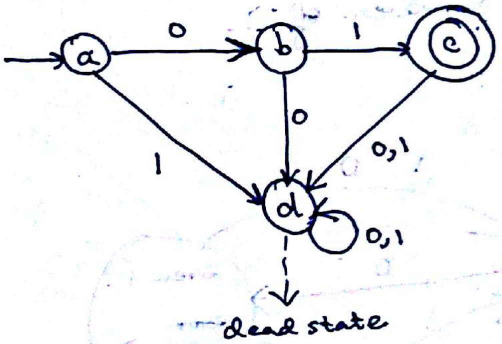
[for any particular δ, there will be only one O/P state.]



Non-Deterministic FA (NFA)



- Draw a DFA with input 01.

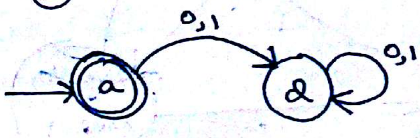


State Table

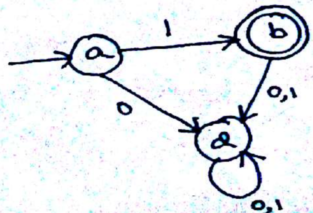
	0	1
a	b	d
b	d	c
c	d	d
d	d	d

- Draw the DFA for accepting ~~with input~~ λ.

λ

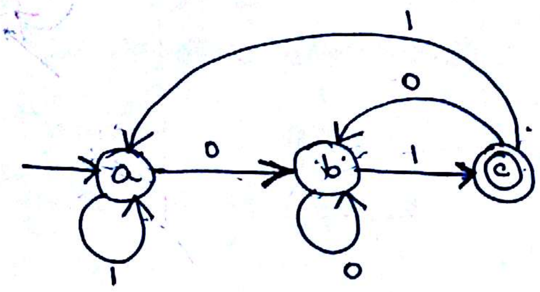


- Draw the DFA for accepting 1.



Scanned by CamScanner

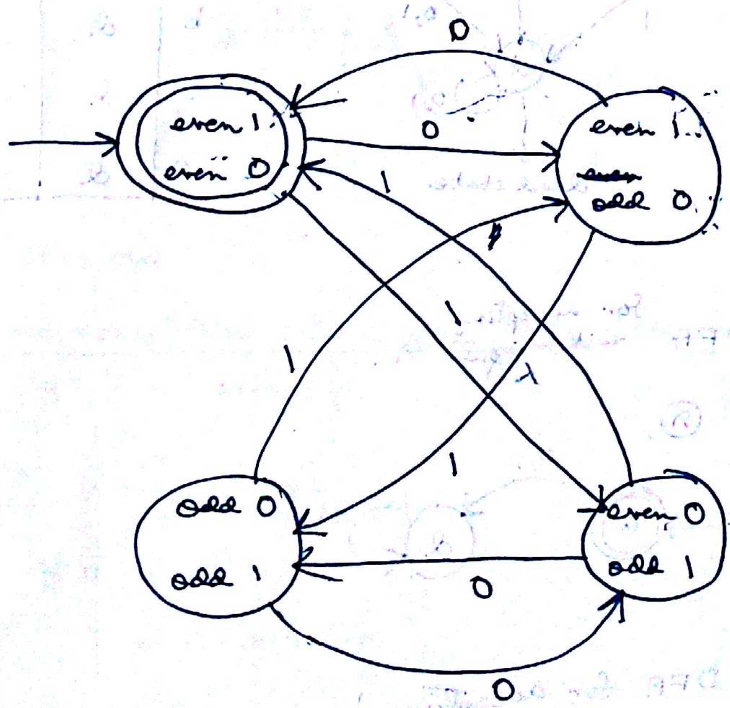
- Draw the DFA for accepting all strings ~~on for~~ ending in 01.



1100100

Cannot be accepted by the DFA

- Draw the DFA for accepting all strings with even number of 0 and even number of 1.



- Draw DFAs for the following —

1) $a(ab)^*aa$ $* \rightarrow$ either 0 or greater than 0

2) $(ab+bb)^*$ $+ \rightarrow$ OR

3) $a(bba+baa)^*bb$

4) $b^* + b^*a(ba)^*$

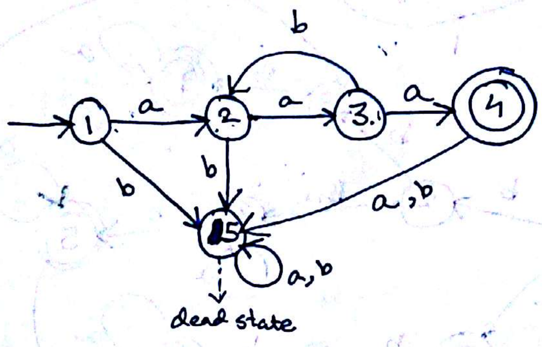
5) Having 01 as substring

6) Exactly two bs and at least one a

7) The sum of the input is divisible by 5.

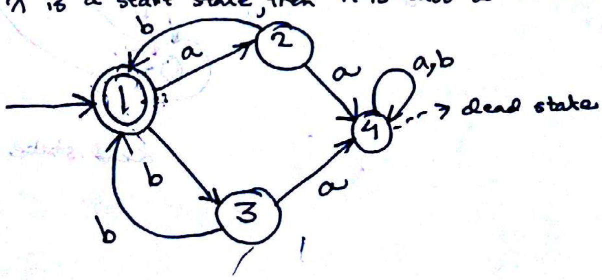
Solutions:

1.

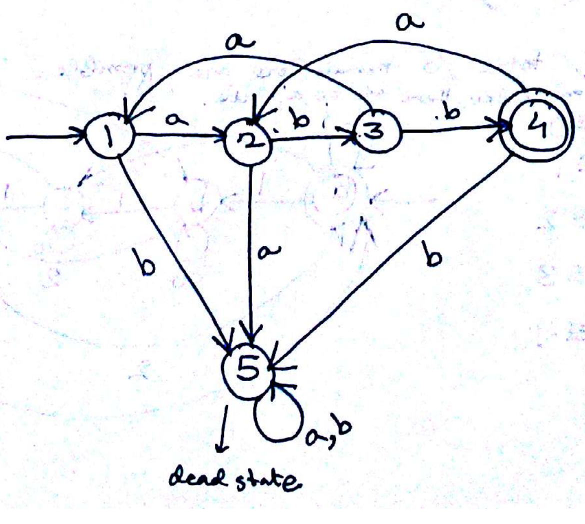


2.

If λ is a start state, then it is also a final state.

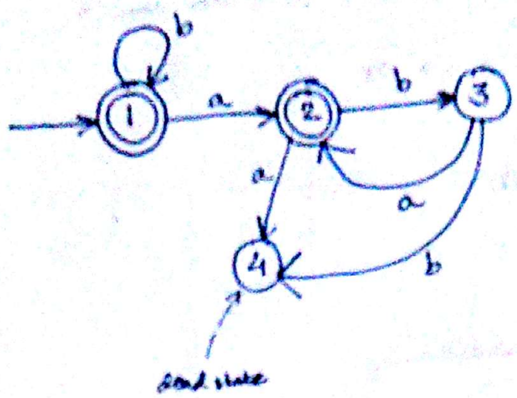


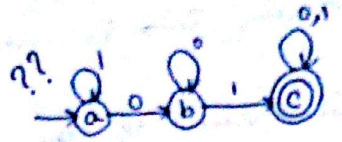
3.



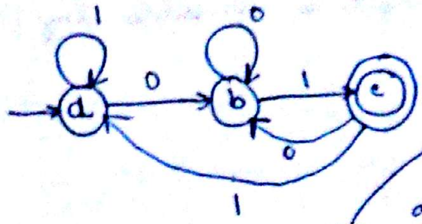
Scanned by CamScanner

4.

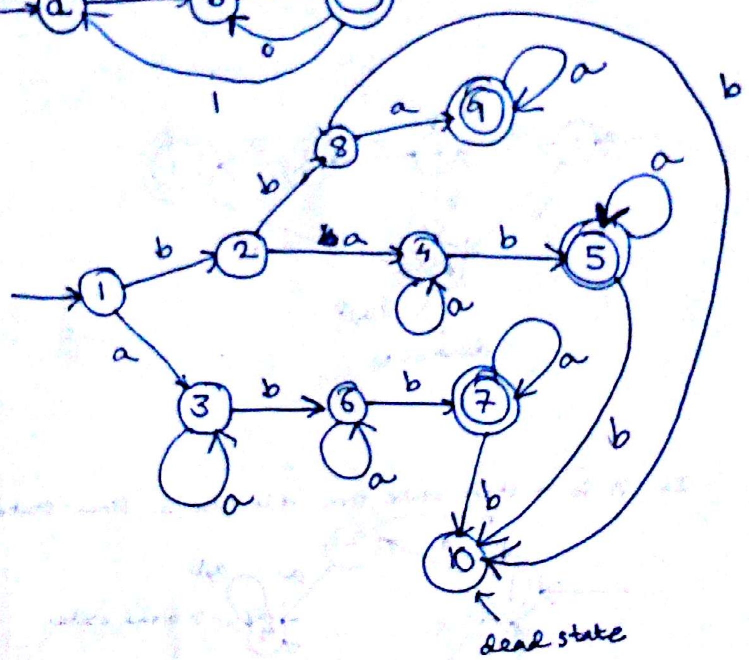




5.



6.



7. $\Sigma = (0,1,2) \rightarrow$ given

For 5, total 5 remainders are possible.
Here for each remainder, there will be a state.

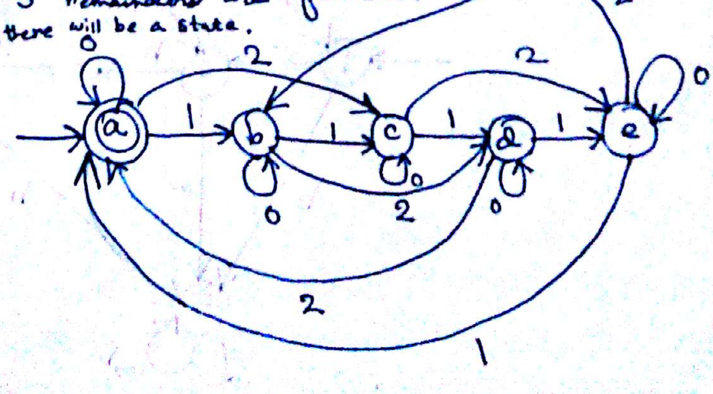
$a \equiv 0$

$b \equiv 1$

$c \equiv 2$

$d \equiv 3$

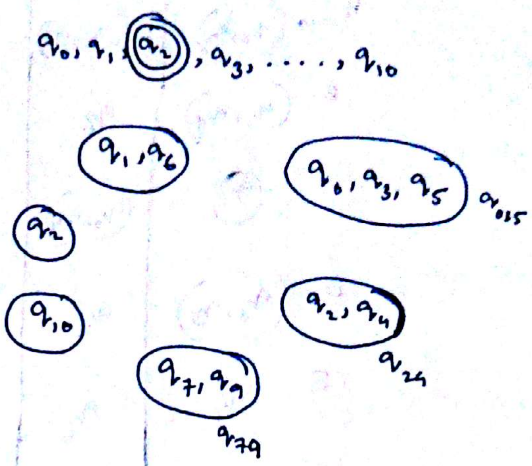
$e \equiv 4$



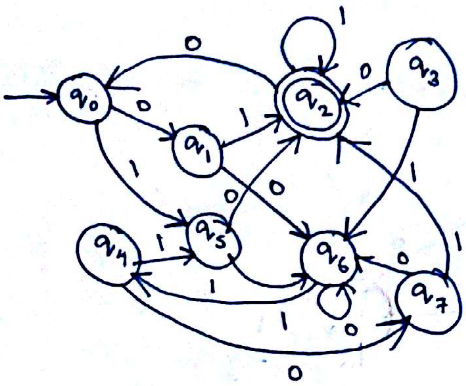
Scanned by CamScanner

Minimization of DFA

- Partitioning



1.



State Table

State	0	1
$\rightarrow q_0$	$q_1^{2,3}$	$q_5^{2,4}$
q_1	$q_6^{1,2}$	$q_2^{3,5}$
q_2	$q_0^{1,1}$	$q_2^{3,5}$
q_3	$q_2^{3,5}$	$q_6^{1,2}$
q_4	$q_7^{2,3}$	$q_5^{2,4}$
q_5	$q_2^{3,5}$	$q_6^{1,2}$
q_6	$q_6^{1,2}$	$q_4^{1,1}$
q_7	$q_6^{1,2}$	$q_2^{3,5}$

Partition — final state — all rest

$\Pi_0 = \{\{q_2\}, \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}\}$

- In Π_0, there will always be two sets within the main set

$\Pi_1 = \{\{q_0, q_4, q_6\},$
$\{q_1, q_3, q_5, q_7\},$
$\{q_2\}\}$

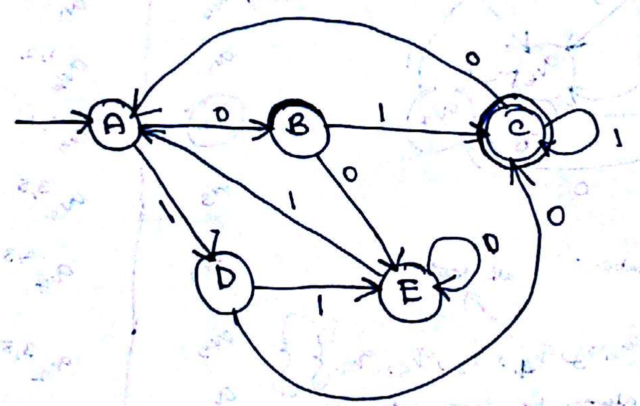
$\Pi_2 = \{\{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\}, \{q_2\}\}$

$\Pi_3 = \Pi_2$

Scanned by CamScanner

State Table for Minimized DFA —

	State	0	1
A	→ (q_0, q_4)	(q_1, q_7)	(q_3, q_5)
B	(q_1, q_7)	q_6	q_2
C	q_2	(q_0, q_4)	q_2
D	(q_3, q_5)	q_2	q_6
E	q_6	q_6	(q_0, q_4)



2.

State	a	b
→ q_0	q_1 2,3,4	q_2 2,3,4
q_1	q_4 1,1,1	q_3 1,1,1
q_2	q_4 1,1,1	q_3 1,1,1
q_3	q_5 2,4,5	q_6 2,2,3
q_4	q_7 2,4,5	q_6 2,2,3
q_5	q_3 1,1,1	q_6 2,2,3
q_6	q_6 2,2,3	q_6 2,2,3
q_7	q_4 1,1,1	q_6 2,2,3

Scanned by CamScanner

$\Pi_0 = \{\{q_3, q_4\}, \{q_0, q_1, q_2, q_5, q_6, q_7\}\}$

$\Pi_1 = \{\{q_3, q_4\}, \{q_0, q_6\}, \{q_1, q_2\}, \{q_5, q_7\}\}$

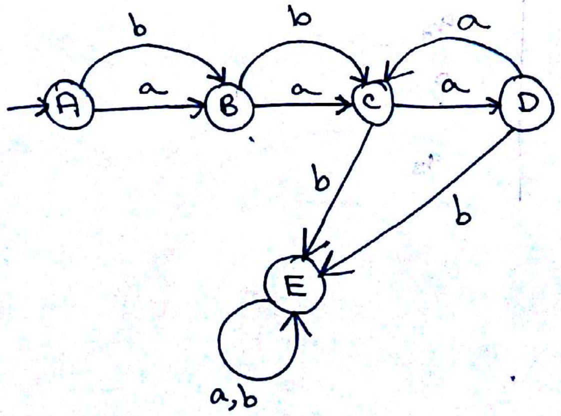
$\Pi_2 = \{\{q_3, q_4\}, \{q_0\}, \{q_6\}, \{q_1, q_2\}, \{q_5, q_7\}\}$

$\Pi_3 = \{\{q_3, q_4\}, \{q_0\}, \{q_6\}, \{q_1, q_2\}, \{q_5, q_7\}\}$

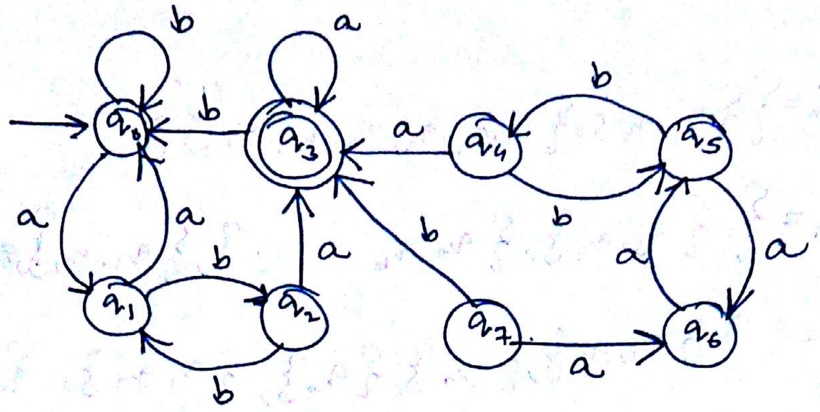
$= \Pi_2$

State Table for Minimized DFA

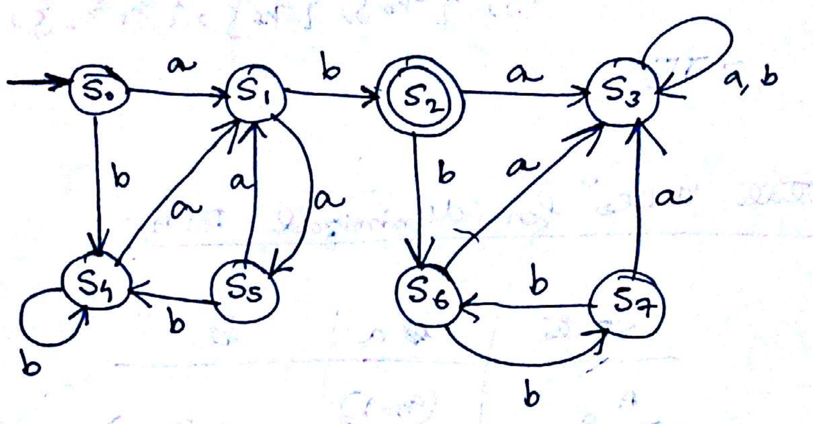
State	a	b
→ A q_0	(q_1, q_2)	(q_1, q_2)
B : (q_1, q_2)	(q_3, q_4)	(q_3, q_4)
C : (q_3, q_4)	(q_5, q_7)	q_6
D (q_5, q_7)	(q_3, q_4)	q_6
E q_6	q_6	q_6



3.



4.



3. State Table

State	a	b
q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	?
q_7	q_6	q_3

4. State Table

State	a	b
S_0	$S_1^{3,4}$	$S_4^{2,2}$
S_1	$S_5^{2,2}$	$S_2^{1,1}$
S_2	$S_3^{2,3}$	$S_6^{2,3}$
S_3	$S_3^{2,3}$	$S_3^{2,3}$
S_4	$S_1^{3,4}$	$S_4^{2,2}$
S_5	$S_1^{3,4}$	$S_4^{2,2}$
S_6	$S_3^{2,3}$	$S_7^{2,3}$
S_7	$S_3^{2,3}$	$S_6^{2,3}$

State Table for Minimized DFA

	State	a	b
A	$\rightarrow S_2$	(S_3, S_6, S_7)	(S_3, S_6, S_7)
B	(S_0, S_4, S_5)	S_1	(S_0, S_4, S_5)
C	(S_3, S_6, S_7)	(S_3, S_6, S_7)	(S_3, S_6, S_7)
D	S_1	(S_0, S_4, S_5)	S_2

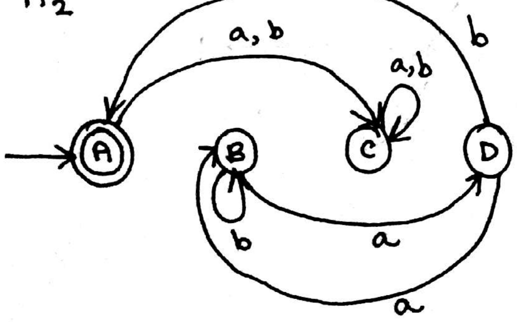
$$\Pi_0 = \{\{S_2\}, \{S_0, S_1, S_3, S_4, S_5, S_6, S_7\}\}$$

$$\Pi_1 = \{\{S_2\}, \{S_0, S_3, S_4, S_5, S_6, S_7\}, \{S_1\}\}$$

$$\Pi_2 = \{\{S_2\}, \{S_0, S_4, S_5\}, \{S_3, S_6, S_7\}, \{S_1\}\}$$

$$\Pi_3 = \{\{S_2\}, \{S_0, S_4, S_5\}, \{S_3, S_6, S_7\}, \{S_1\}\}$$

$$= \Pi_2$$



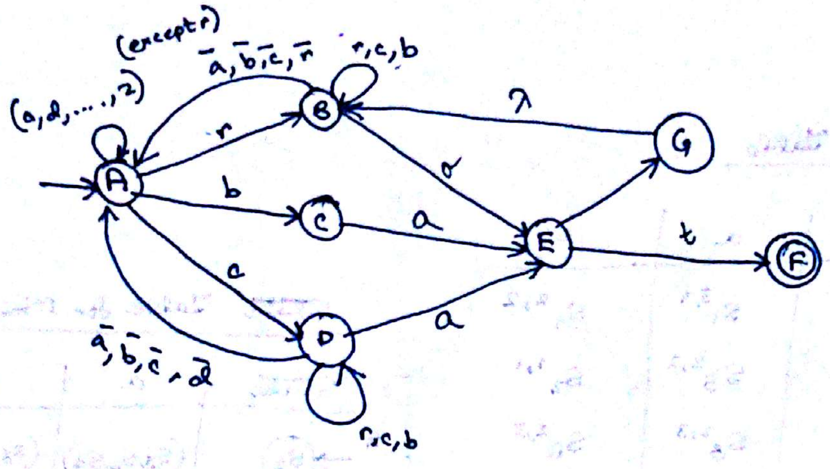
Scanned by CamScanner

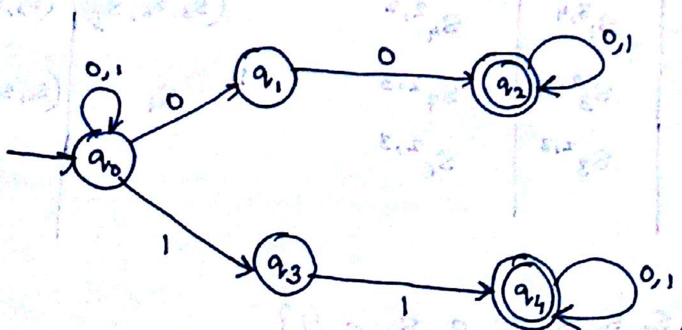
Draw DFA for —

1) $L = \{a^m b^n \mid n \geq 2, m \geq 1\}$

2) $L = \{a^n \mid n \geq 0, n \neq 3\}$

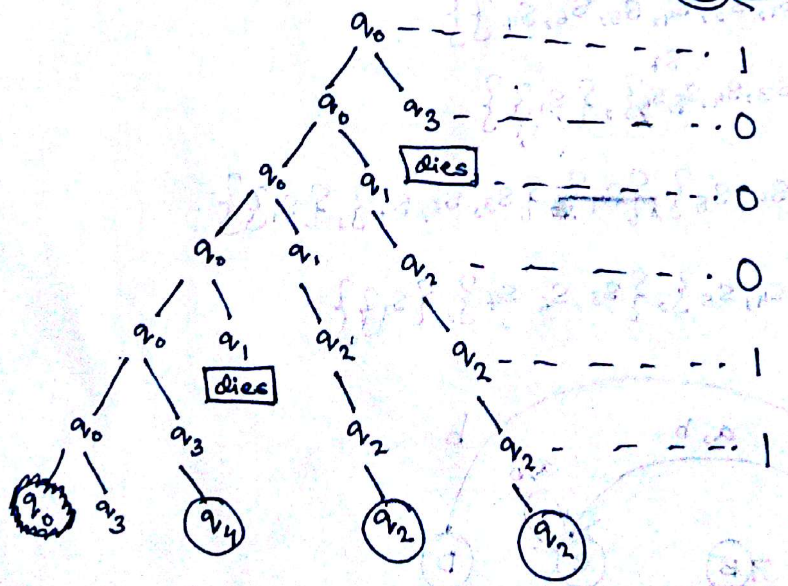
3) Construct a DFA for all strings containing rat, bat or cat and $\Sigma = \{a, b, \ldots, z\}$.





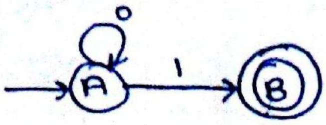
$\delta(q_0, 0) = \{q_0, q_1\}$

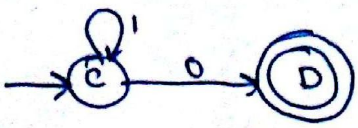
Check if 100011 is ~~p~~ possible or not.

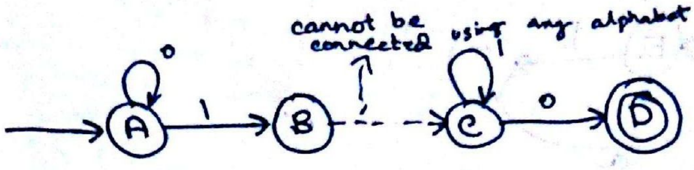


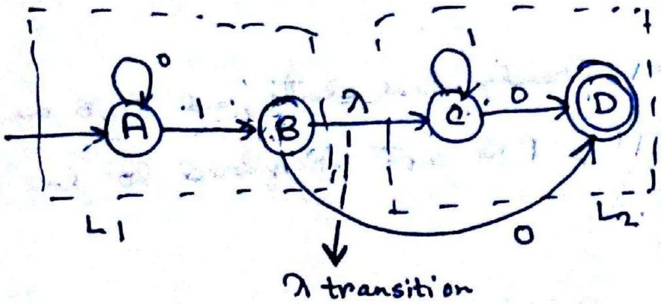
$L = L_1 . L_2$

$= \{ 0^n 1 1^m 0 \mid n \geq 0, m \geq 0 \}$



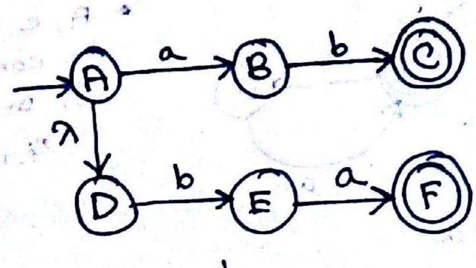


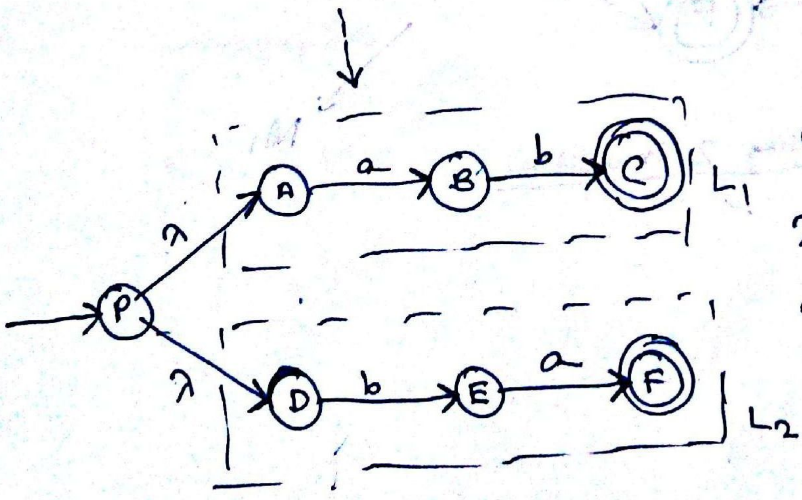




2. $L_1 = \{ab\}$ $\quad L = L_1 \cup L_2$

$L_2 = \{ba\}$





- λ closure of any state, are the state itself, and any state that is reachable from the current

$\lambda(q)$

$\lambda(B) = \{B, C\}$

$\lambda(P) = \{A, D, P\}$

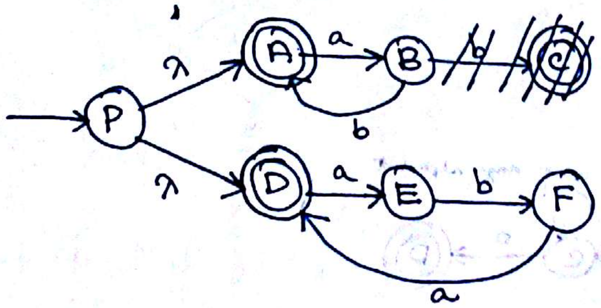
NFA using λ Transition:

1) $L = (ab \cup aba)^*$

2) $L = (ab)^*(ba)^* \cup aa^*$

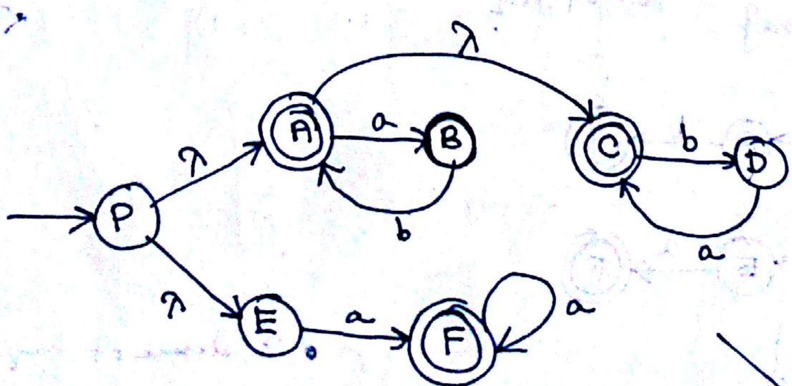
3) $L = \{a^n \mid n \geq 0 \cup b^n a \mid n \geq 1\}$

1. $L = (ab \cup aba)^* = (L_1 \cup L_2)^*$



∵ A and D are already final state, and A, D are the λ closure of P, P is not required to be made a final state.

2. $L = (ab)^*(ba)^* \cup aa^* = L_1.L_2 \cup L_3$

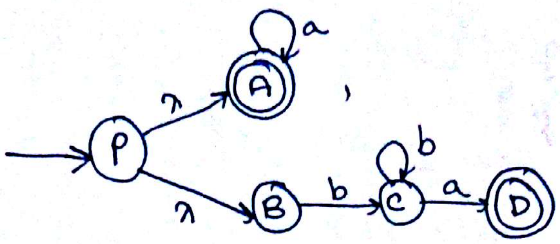


- A, C needs to be connected by λ transition, ∵ A is the final state.

$\rightarrow M_1$

NFA Using λ Transition

3. $L = \{\underbrace{a^n \mid n \geq 0}_{L_1} \cup \underbrace{b^n a \mid n \geq 1}_{L_2}\} = L_1 \cup L_2$



Acceptance of a string x in an NFA :

$\delta^*(q_0, x) = q_1 \in F$ — δ^*: extended transition function; x: String

$\delta(q, a)$ ----> transition function; a: alphabet

$$\delta^*(q_0, x) \qquad x = aba$$

$$= \delta^*(\delta(q_0, a), ba)$$

$$= \delta^*(\underbrace{\delta(\underbrace{\delta(\underbrace{q_0, a}_{q_1}), b}_{q_2}), a}_{q_3})$$

Finding NFA without using λ

Step 1:

$\lambda(P) = \{A, E\} \mid \lambda(P) \cap F \neq \phi$ — F: set of final states $[\{A, C, F\}]$

$\lambda(P) \cap F = \{A\}$

$\therefore$ P should be a final state.

$M_1 \rightarrow M_2$

NFA-λ $\longrightarrow$ NFA (without λ)

~~$F_2 = F_1 \cup (\lambda$~~

$F_2 = F_1 \cup q_1 \; [\text{if } \lambda(q_1) \cap F \neq \phi]$

Scanned by CamScanner

Refer to M_1 in 2

$\delta(A,a) = B$, Check if λ is present from there. Reach the state where λ reaches. If a is present ~~[illegible]~~, add the state it reaches.

$\delta(A,b) = D$

$\delta(B,a) = \emptyset$

$\delta(B,b) = \{A,C\}$

$\delta(C,a) = \emptyset$

$\delta(C,b) = D$

$\delta(D,a) = C$

$\delta(D,b) = \emptyset$

$\delta(E,a) = F$

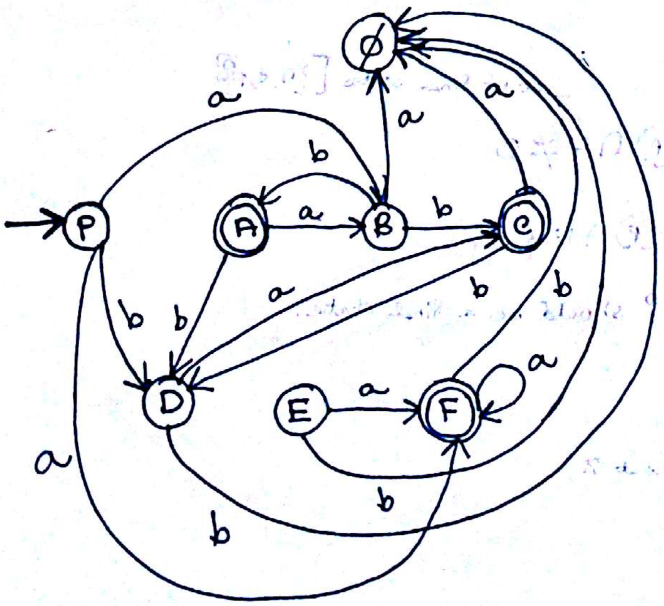
$\delta(E,b) = \emptyset$

$\delta(F,a) = F$

$\delta(F,b) = \emptyset$

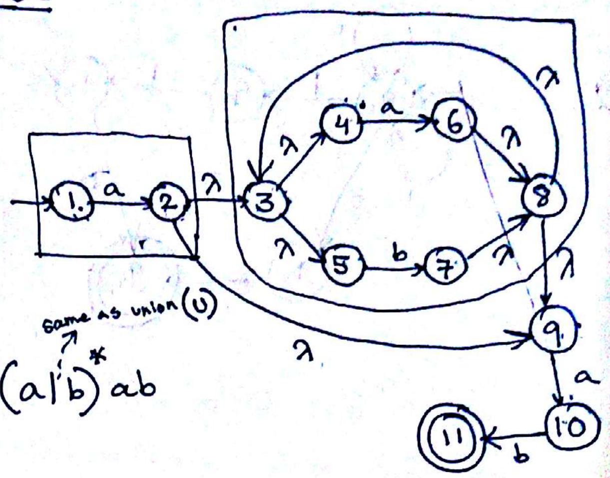
$\delta(P,a) = \{B,F\}$

$\delta(P,b) = D$



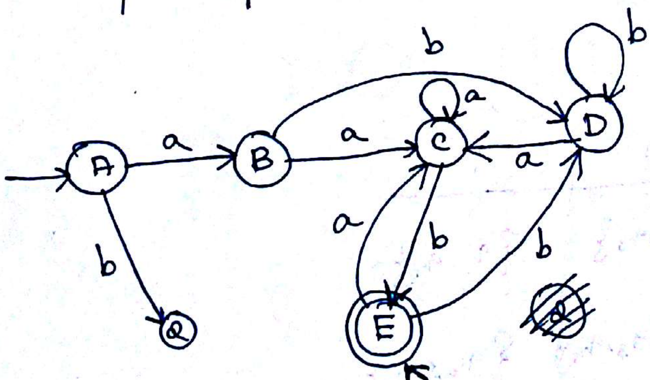
NFA to DFA Construction

1. NFA-λ to NFA-without λ



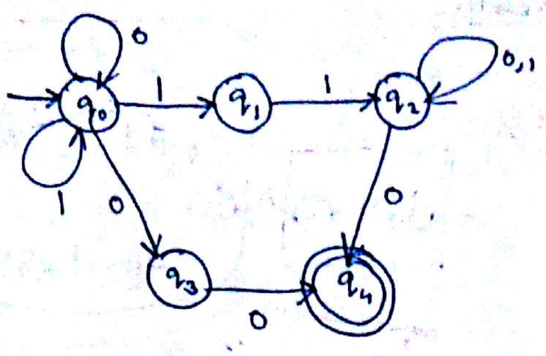
$a(a|b)^*ab$

	a	b
A (1)	{2,3,4,5,9}	∅
B {2,3,4,5,9}	{6,10,8,3,9,4,5} C	{7,8,3,9,4,5} D
C {6,10,8,3,9,4,5}	{10,6,8,3,9,4,5} C	{11,7,8,9,3,4,5} E
D {7,8,3,9,4,5}	{10,6,8,3,4,5,9} C	{7,8,3,9,4,5} D
E {11,7,8,3,9,4,5}	{10,6,8,3,4,5,9} C	{7,8,3,9,4,5} D

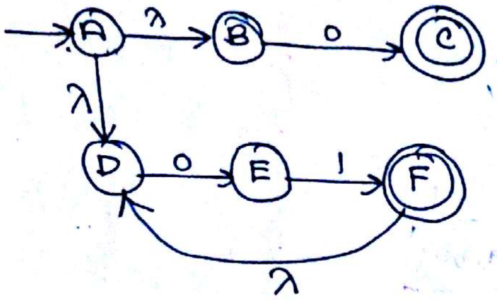


E is the final state ∵ only in E, 11 is present

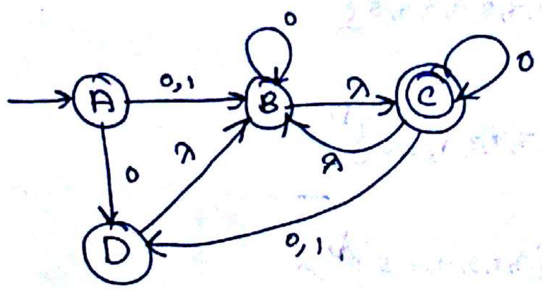
2.



3.



4.



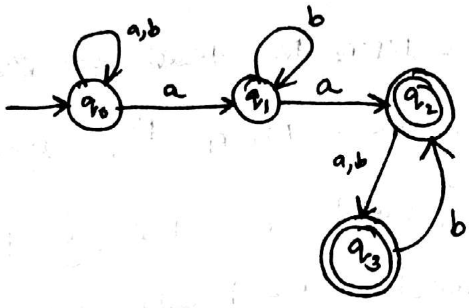
Solutions:

2.

	0	1
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
$\{q_0, q_3\}$	$\{q_0, q_3, q_4\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_3, q_4\}$	$\{q_0, q_3, q_4\}$	$\{q_0, q_1\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_3, q_4, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2, q_3, q_4\}$	$\{q_0, q_3, q_2, q_4\}$	$\{q_0, q_1, q_2\}$

Scanned by CamScanner

5.



6. Design NFA-λ for $\{abab^n \mid n \geq 0\} \cup \{aba^n \mid n \geq 0\}$ Then remove the λ-transactions. Then find the corresponding DFA.

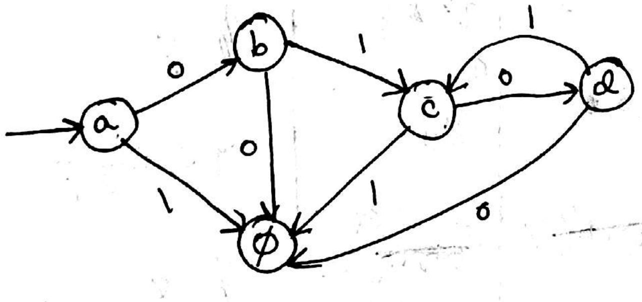
3.

	0	1
a $\rightarrow \{A, B, D\}$	b $\{C, E\}$	$\emptyset$
b $\{C, E\}$	$\emptyset$	c $\{F, D\}$
c $\{F, D\}$	d $\{E\}$	$\emptyset$
d $\{E\}$	$\emptyset$	c $\{F, D\}$

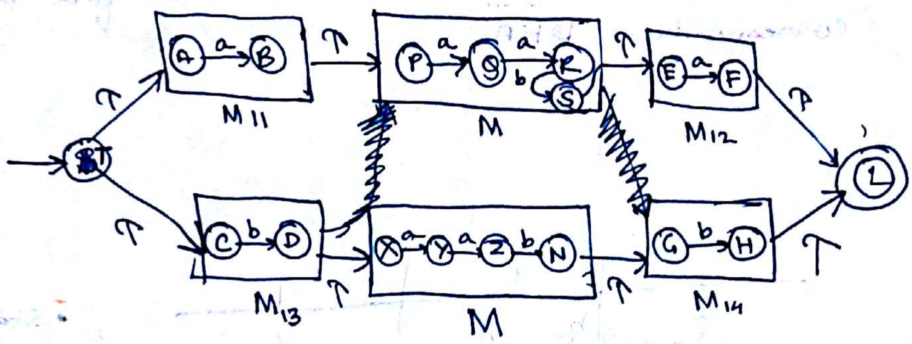
- Start State $= \{A, B, D\}$

$\lambda(A) = \{A, B, D\}$

$\therefore$ if ~~A~~ is the start state, B, D are both start states.



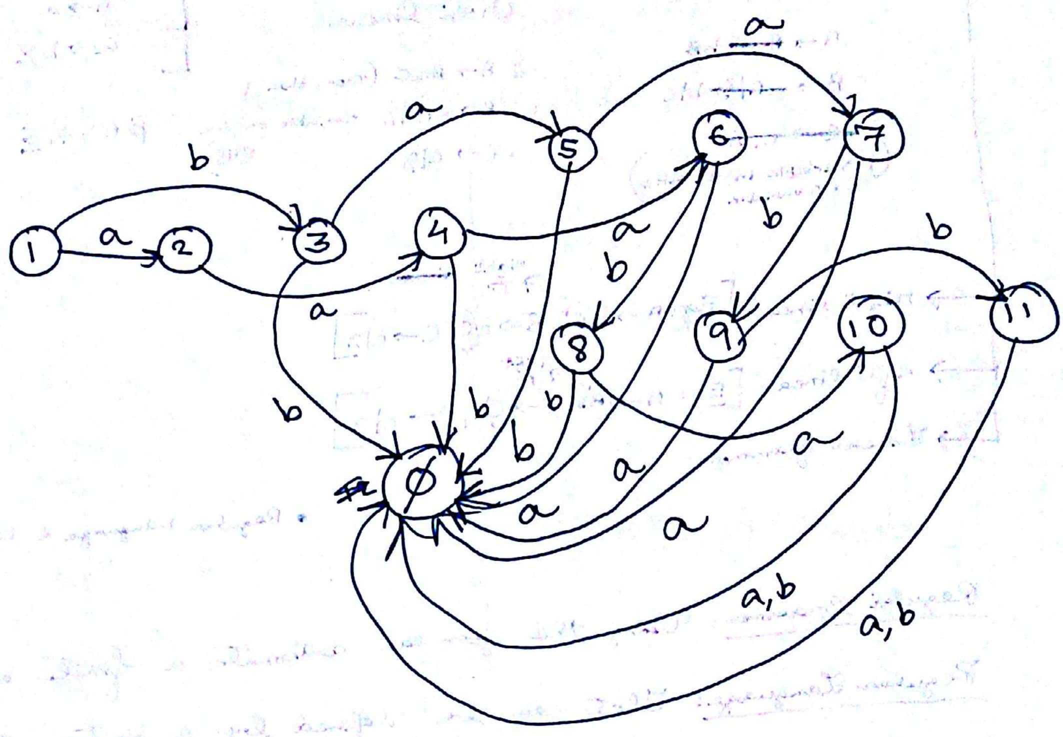
1. $\Sigma = \{a,b\}$. Given an NFA M that accepts the language L, design a new NFA M_1 that accepts the language $L_1 = \{twt \mid w \in L, t \in Z\}$. For eg. if aab is in L, then aaaba or baabb are in L_1.



	a	b
1 → $\{T, A, C\}$	2 $\{B, P\}$	3 $\{D, X\}$
2 $\{B, P\}$	4 $\{Q\}$	~~$\{S, E\}$~~ $\emptyset$
3 $\{D, X\}$	5 $\{Y\}$	$\emptyset$
4 $\{Q\}$	6 $\{R\}$	~~$\{S, E\}$~~ $\emptyset$
5 ~~$\{S, E\}$~~ Y	7 ~~$\{F, L\}$~~ Z	$\emptyset$
6 ~~$\{Y, Z\}$~~ R	$\emptyset$	8 $\{S, E\}$
7 Z	$\emptyset$	9 $\{N, G\}$
8 $\{S, E\}$	10 F	$\emptyset$
9 $\{N, G\}$	$\emptyset$	11 H

Scanned by CamScanner

10 F	ϕ	ϕ
11 H	ϕ	ϕ



$G = (V, \Sigma, P, S)$

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

$S \to aSb \mid ab$

$S \to aSb$

$\Rightarrow a^2Sb^2$

$\Rightarrow a^3Sb^3$

$\Rightarrow a^4b^4$

Sentential Form

$S \overset{*}{\Rightarrow} a^4b^4$

a^4b^4 is derivable from S.

Scanned by CamScanner

$G = (V, \Sigma, P, S)$

Grammer

Linear

$A \to$ ~~Bac~~ bB

$B \to$ ~~aA|a~~ $b|c$

~~C → c|λ~~

(1 variable in the RHS on 0 variable)

Non linear

$A \to BaC$ (more than 1

$B \to aA|\lambda$ variable on the RHS)

$C \to c|\lambda$

• $\alpha \to \beta$

$\alpha \in V \times \Sigma$

[eg: $aA \to baA$

$A \to a$

$a \to bX$]

$\beta \in V \times \Sigma$

→ right linear [Eg: $A \to bB$, $B \to cC$, $C \to c|\lambda$] right ~~linear~~

→ left linear [Eg: $A \to Bb$, $B \to Cc$, $C \to c|\lambda$] left

→ Linear grammer

• Regular Language to Linear Grammar.

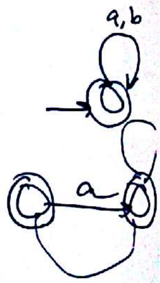
<u>Regular grammer</u>: Using this you can automate a finite automata.

<u>Regular Language</u>: That can be defined by a finite automata.

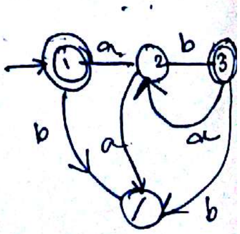
$L = \{a^n b^n \mid n \geq 0\} \to$ Can't Draw

$L = \{a^n b^n \mid 2 \geq n \geq 0\}$

$= \{\lambda, ab, a^2b^2\}$

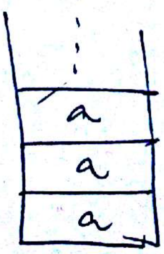


• If our language is finite, then there is guaranteed to be a corresponding finite automata.



Stack or queue can be used to keep track of the number of a's.

$a^P b^P$ $P \geq 0$



- If we cannot draw a finite automata, for a particular language, then it cannot be regular.

Properties:

- ~~alp~~ Σ must be same for L_1 and L_2

L_1 regular, L_2 regular $\Rightarrow M_1, M_2$ exist

1) $L_1 \cup L_2$ regular $\} \rightarrow M$
2) $L_1 \cap L_2$ "
3) $\overline{L_1}$ "
4) $L_1 - L_2$ "
5) $L_1 . L_2$ "
6) L_1^* "
7) L_1^R "

$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$

$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

$M = (Q, \Sigma, \delta, q, F)$

$Q = Q_1 \times Q_2$ ~~$\delta = \delta_1 \times \delta_2$~~ $A \in Q_1$, $B \in Q_2$

$\delta = \delta(AB, a)$ $\delta(A,a), \delta(B,a)$

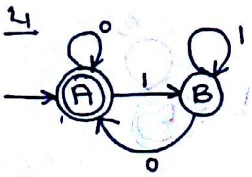
$q = q_1 \times q_2$

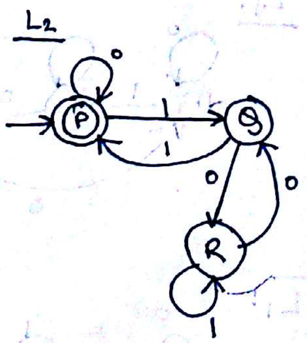
$F = PQ$, $P \in F_1$, $Q \in F_2$

Examples—

1) $L_1 = \{x \in \{0,1\}^* \mid$ binary number is divisible by $2\}$

$L_2 = \{x \in \{0,1\}^* \mid$ binary number is divisible by $3\}$





	0	1
→AP	AP	BQ
BQ	AR	BP
AR	AQ	BR
BP	AP	BQ
AQ	AR	BP
BR	AQ	BR

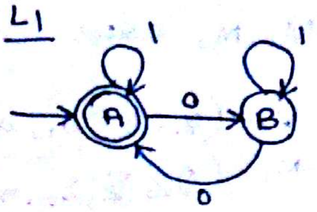
- Draw the DFA

16 8 4 2 1
1 0 0 1 1

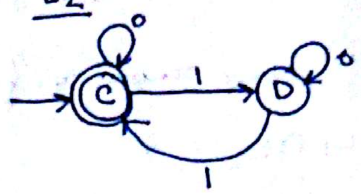
2) $L_1 = \{ x \in \{0,1\}^* \mid$ even number of zero's $\}$

$L_2 = \{ x \in \{0,1\}^* \mid$ even number of one's $\}$

L_1



L_2

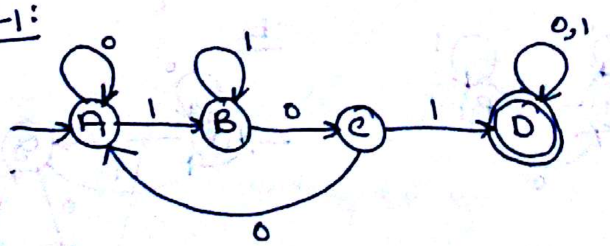


	0	1
→ AC	BC	AD
BC	AC	BD
AD	BD	AC
BD	AD	BC

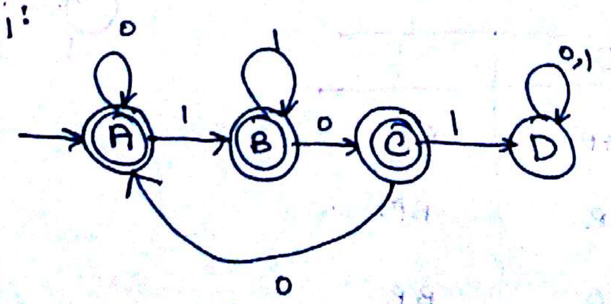
3) $L_1 = \{ x \in \{0,1\}^* \mid$ x has substring ~~ends in~~ 101 $\}$

- whatever be the final state of L_1, it should be a non-final state of $\overline{L_1}$.

L_1:



$\overline{L_1}$:



4) $L_1 - L_2$

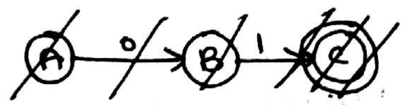
$= \underbrace{L_1}_{\text{regular}} \cap \underbrace{\overline{L_2}}_{\text{regular}}$

Now, $L_1 \cap L_2 \rightarrow$ regular

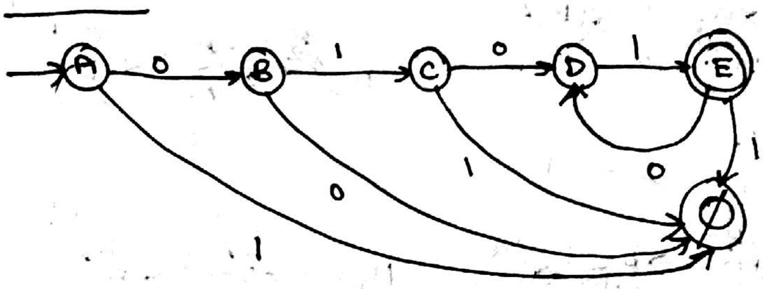
$L_1 = \{(01)^n \mid n \geq 1\}$

$L_2 = \{01\}$

$L_1 - L_2 = ?$ $\quad\therefore L_1 - L_2 = \{(01)^n \mid n \geq 2\}$



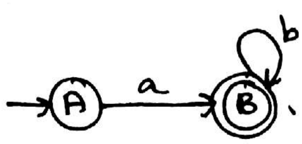
$\underline{L_1 - L_2}$



7) $L_1 = \{ab^n \mid n \geq 0\}$

$L_2 = \{b^n a \mid n \geq 0\}$

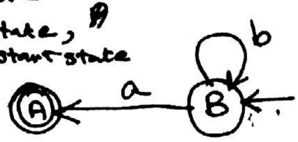
$L_2 = L_1^R$



• If more than 1 final state is present in L_1, then λ transitions need to be used in L_2

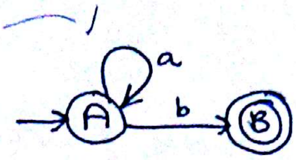
$\Downarrow R$

Make start state the final state, final states the start state

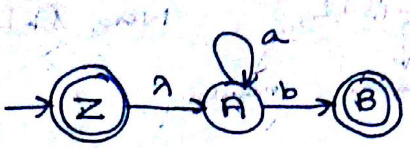


Scanned by CamScanner

6) L_1



$L_2 = L_1^*$

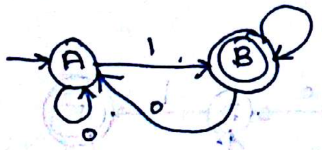


1. Construct automata for

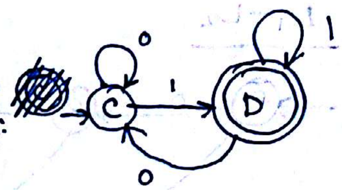
$L = \{x \in \{0,1\}^* \mid x$ contains 01 as substring and ends in $1\}$

2. Verify whether the following DFAs are equivalent or not

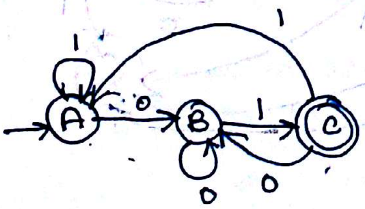
i) M_1:



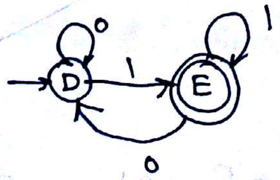
M_2:



ii) M_1:



M_2:



3. Prove that the following language is regular —

$L = \{ababⁿ \mid n \geq 0\} \cup \{abaⁿ \mid n \geq 0\}$

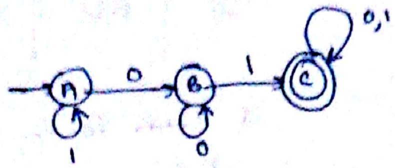
Scanned by CamScanner

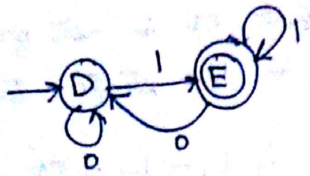
Answers :

1. $L_1 = \{x \in \{0,1\}^* \mid x$ contains 01 as substring$\}$

$L_2 = \{x \in \{0,1\}^* \mid x$ ends in 1$\}$

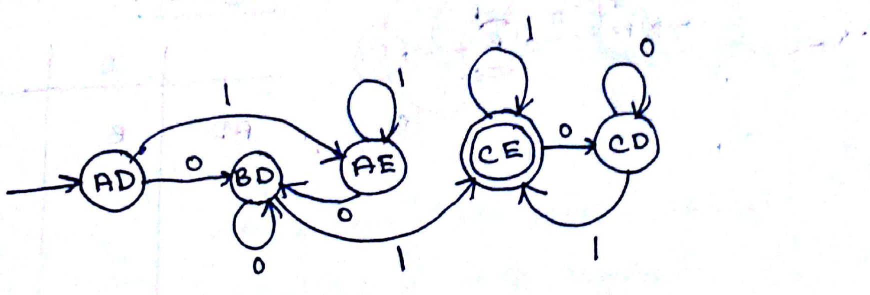
L_1 :





L_2

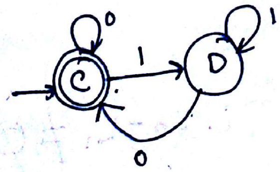
	0	1
→AD	BD	AE
BD	BD	CE
AE	BD	AE
CE	CD	CE
CD	CD	CE



• If $L_1 - L_2 = L_2 - L_1 = \emptyset$, then equivalent.

2. i) $L_1 - L_2 = L_1 \cap \overline{L_2}$

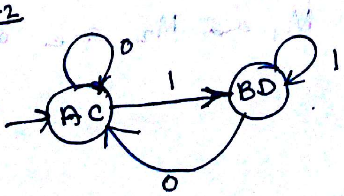
$\overline{L_2}$:



	0	1
→AC	AC	BD
BD	AC	BD

L_1 : Same as M_1

$L_1 \cap \overline{L_2}$



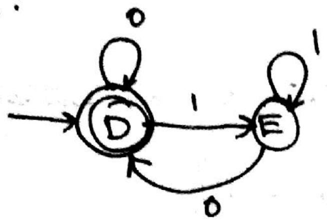
No final state.

$L_1 - L_2 = \emptyset$.

If we find $L_2 - L_1$, then we will see $L_2 - L_1 = \emptyset$. [Proved]

... are equivalent

ii) $\overline{L_2}$:



$L_1 \cap \overline{L_2}$:

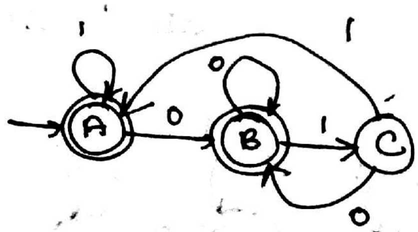
	0	1
→ AD	BD	AE
BD	BD	CE
AE	BD	AE
CE	BD	AE

CD needs to be the final state.

But CD is not present in the state table.

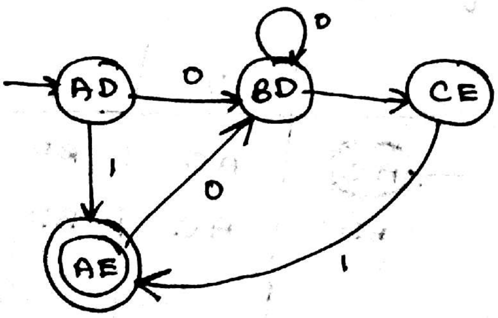
$\therefore L_1 - L_2 = \phi$.

$\overline{L_1}$:



$L_2 \cap \overline{L_1}$:

	0	1
AD	B	



$\therefore L_2 - L_1 \neq \phi$, $\because$ there is a final state.

$\therefore L_1 - L_2 \neq L_2 - L_1$

$\therefore M_1$ and M_2 are not equivalent (Ans)

3.

4. Write

a)

b)

c)

d)

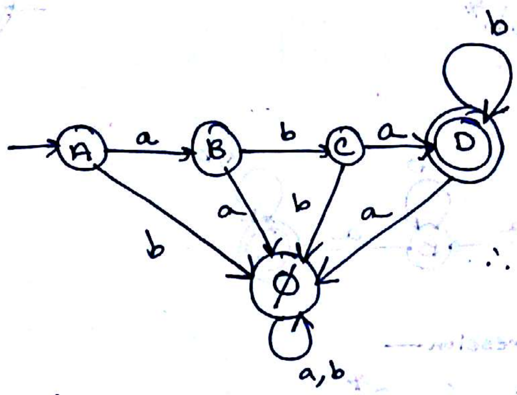
e)

Scanned by CamScanner

3. $L_1 = \{ abab^n \mid n \geq 0 \}$

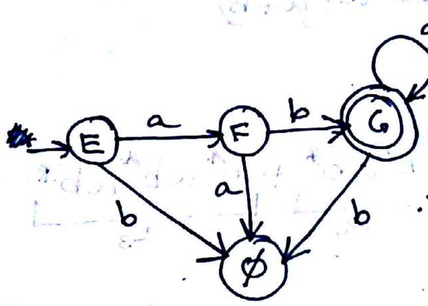
$L_2 = \{ aba^n \mid n \geq 0 \}$

$\underline{L_1}$



$\therefore L_1$ is regular

$\underline{L_2}$



$\therefore L_2$ is regular.

$\because L_1$ and L_2 are both regular, hence

$L = L_1 \cup L_2$ is also regular.

4. Write down the regular expressions for the following—

a) the set of all strings containing exactly 2 a's.

b) " " " " " " atleast 2 a's.

c) " " " " " " atmost 2 a's.

d) " " " " " " the substring aa

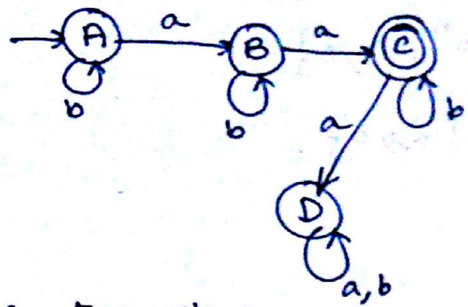
e) $a^m b^n c^p$ where $m, n, p \geq 1$

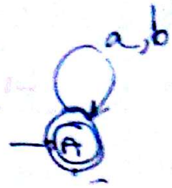
f) $a^m b^{2n} c^{3p}$ where $m, n, p \geq 1$

g) $a^n b a^{2m} b^2 \mid m \geq 0, n \geq 1$

Scanned by CamScanner

4. a)

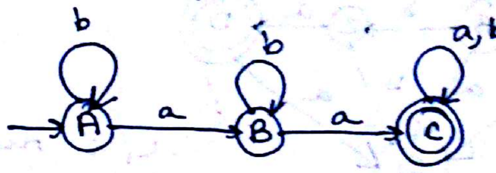




Regular Expression:

$b^*ab^*ab^*$

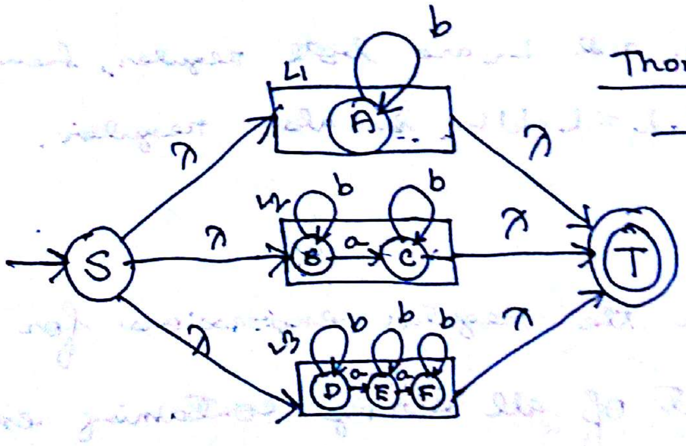
b)



Regular Expression—

$b^*ab^*a(a+b)^*$

c) $\underbrace{b^*}_{L_1} + \underbrace{b^*ab^*}_{L_2} + \underbrace{b^*ab^*ab^*}_{L_3}$



Thomson's Construction: NFA from Regular Expression

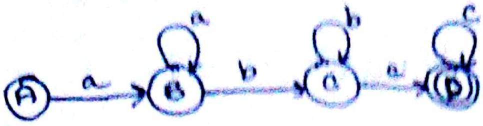
• NFA to DFA: Subset Construction

• Minimization of DFA: Partitioning

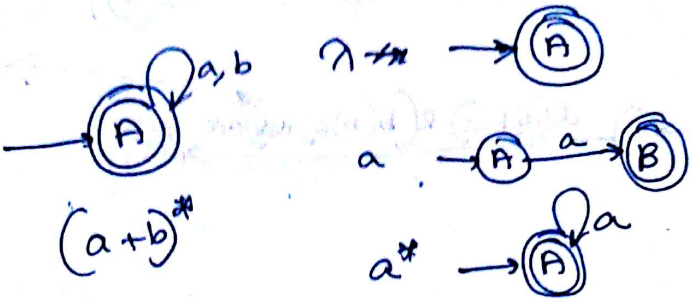
e)

Regular Expression

$aa^* \, bb^* \, cc^*$

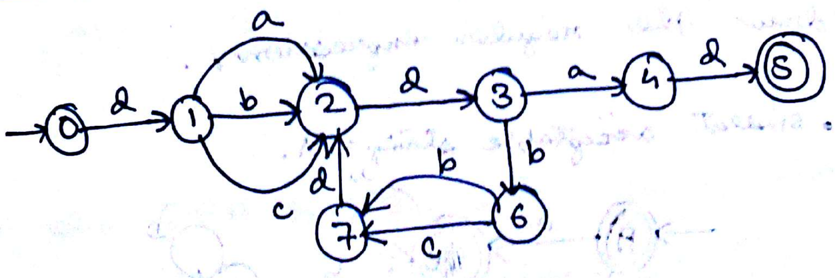


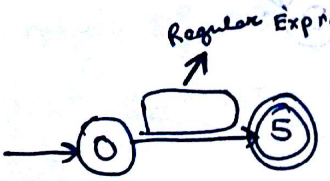
DFA → RE



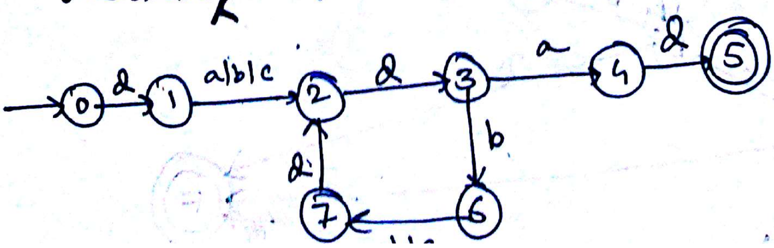
Example:

1.



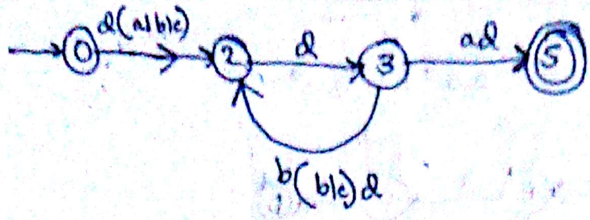


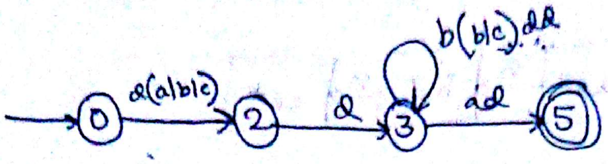
• ~~Combine~~ Remove all parallel edges



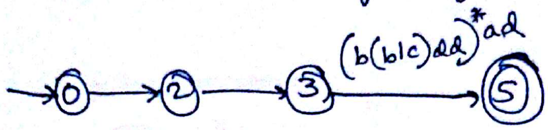
Scanned by CamScanner

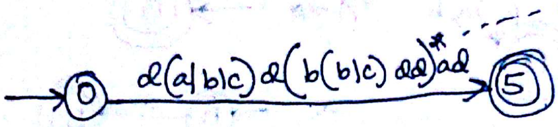
• Resolve all edges in series





• Resolving the self loop—



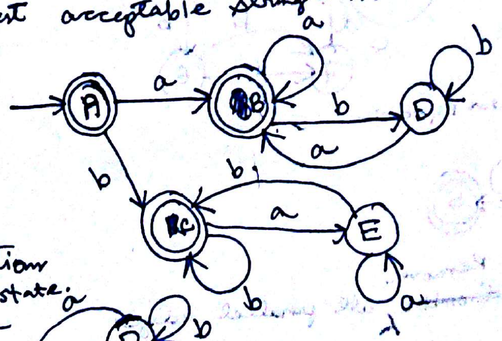


Regular Expression

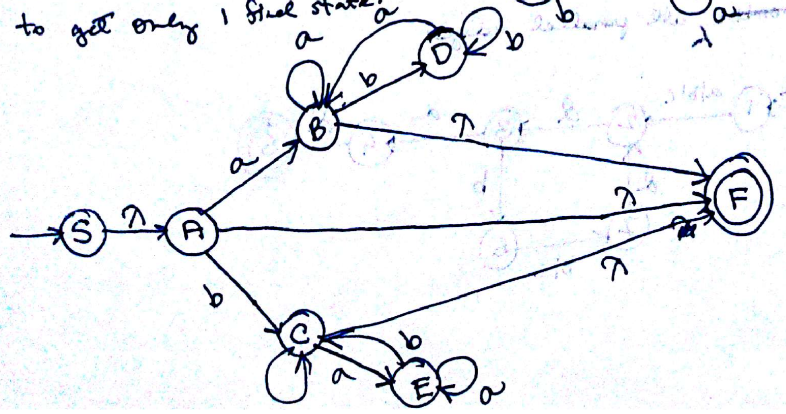
• Draw the DFA for all strings that start and end with same symbol. $\Sigma = \{a,b\}$.

Then draw the regular expression.

• Smallest acceptable string = λ.

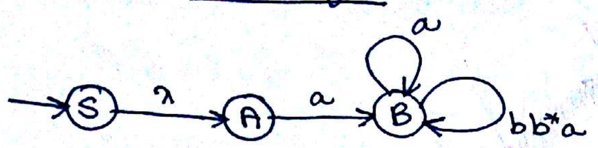


• Introduce λ transition to get only 1 final state.

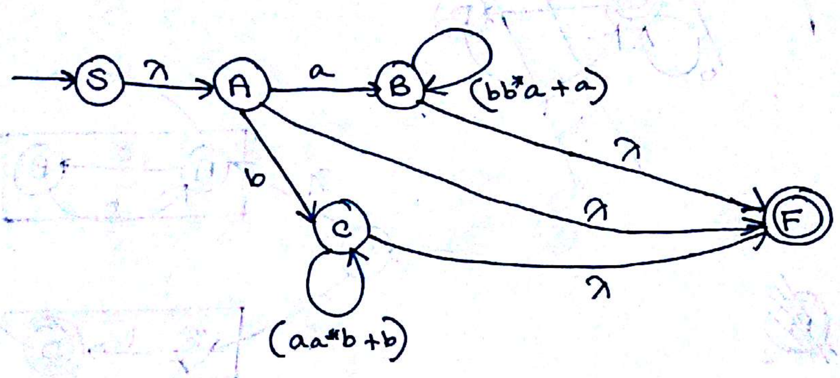


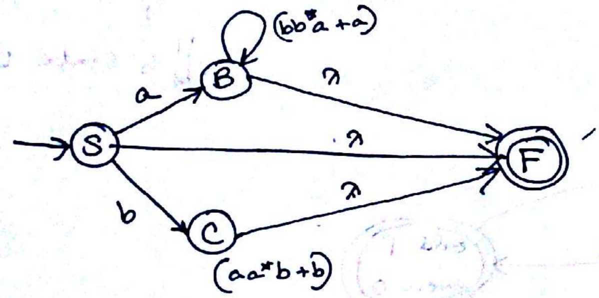
Scanned by CamScanner

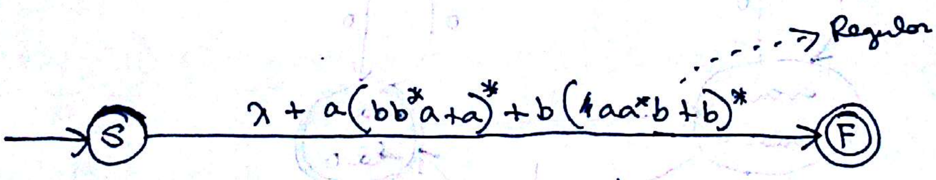
Resolving D —



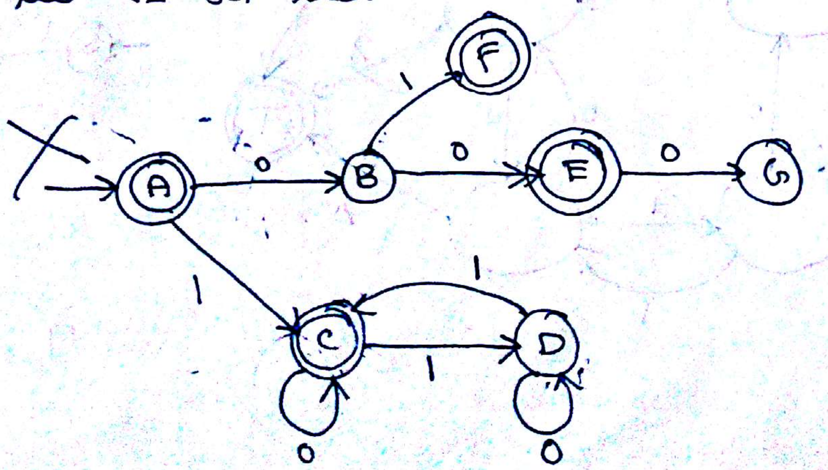
⇓

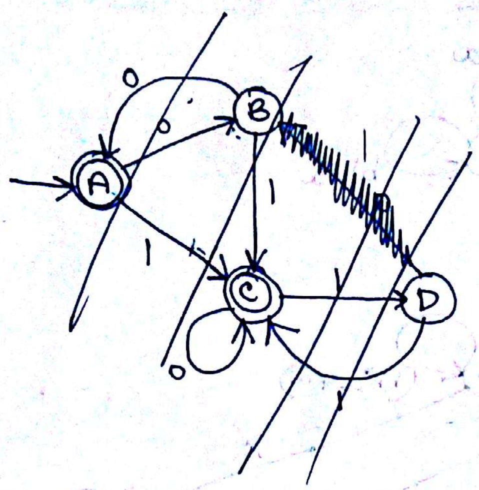


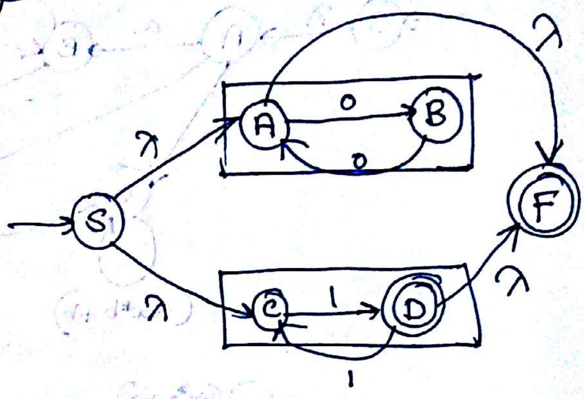




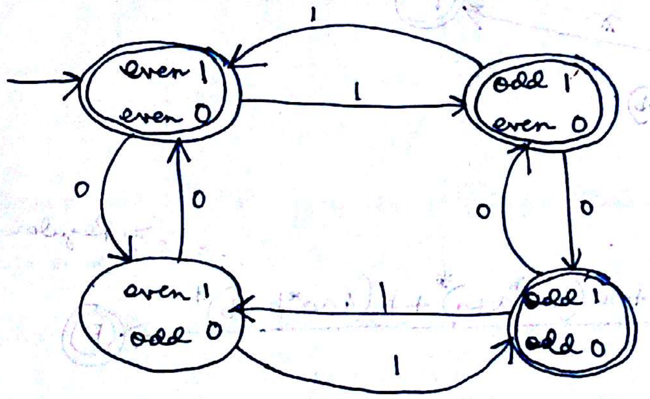
- Draw the DFA for the strings having either even number of 0's or odd number of 1's.
  Find the RE for this.

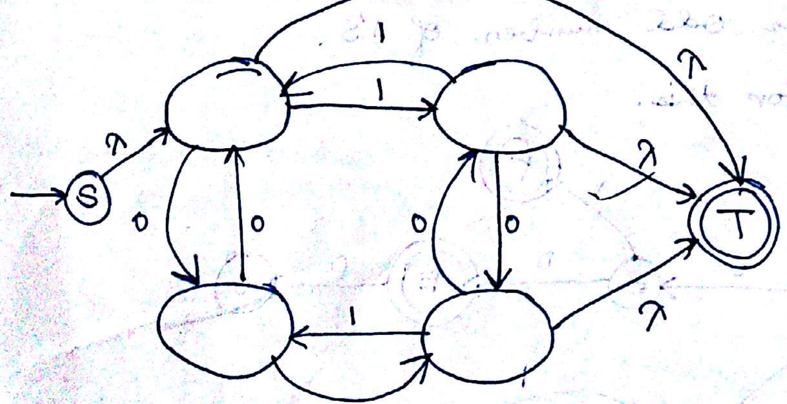


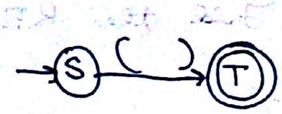




⇓ 4 state DFA







Scanned by CamScanner

Rules of Regular Expressions:

1) $\phi + R = R$
2) $\phi R = R\phi = \phi$
3) $\lambda R = R\lambda = R$
4) $\lambda^* = \lambda$
5) $R + R = R$
6) $R^* R^* = R^*$

7) $RR^* = R^*R$
8) $(R^*)^* = R^*$
9) $\lambda + RR^* = R^*$
10) $(P+Q)^* = (P^*Q^*)^*$

$A + AB^*B$

1. Prove that

$$\underbrace{(1+00^*1)}_{A} + \underbrace{(1+00^*1)}_{A}\underbrace{(0+10^*1)^*}_{B}\underbrace{(0+10^*1)}_{B} = 0^*1\underbrace{(0+10^*1)^*}_{B}$$

$\therefore$ L.H.S

$= A + AB^*B$

$= A(\lambda + B^*B)$

$= A(\lambda + BB^*)$ [Rule 7]

$= AB^*$ [Rule 9]

$= (1+00^*1)(0+10^*1)^*$

$= (\lambda + 00^*)1(0+10^*1)^*$

$= 0^*1(0+10^*1)^*$ [Rule 9]

$=$ RHS [Proved]

Scanned by CamScanner

Arden's Theorem

$R = Q + RP$ R, P, Q are RE

~~R =~~ $R = \cancel{PQ^*} QP^*$ is unique solution of the equation

Proof:

L.H.S

$R = \cancel{PQ^*} QP^*$

R.H.S

$Q + RP \cancel{= Q + PQ^*P}$

$= Q + QP^*P$

$= Q(\lambda + P^*P)$

$= Q(\lambda + PP^*)$

$= QP^*$

$\therefore$ L.H.S = R.H.S

$\therefore$ $R = QP^*$ is a solution.

$R = Q + RP$

$= Q + (Q + RP)P$

$= Q + QP + RP^2$

$= Q + QP + (Q + RP)P^2$

$= Q + QP + QP^2 + RP^3$

$\vdots$

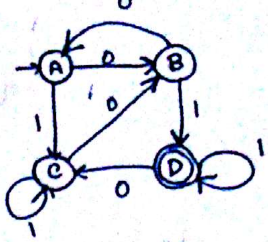
$= Q + QP + QP^2 + \ldots\ldots + QP^i + RP^{i+1}$

$\Rightarrow R = Q(\lambda + P + P^2 + \ldots\ldots + P^i) + RP^{i+1}$

Scanned by CamScanner

Example:

1)



$A = \lambda + B0$

$B = A0 + C0$

$C = C1 + A1 + D0$

Final state $D = B1 + D1$

Regular Expression of DFA

For more than 1 final state OR of all of them.

If B and D are final states, the RE = B + D.

$D = B1 + D1$ $\left[\begin{array}{l} R = D \\ Q = B1 \\ P = 1 \end{array}\right.$

$D = B11^* \ldots$ (1)

$A = \lambda + B0 = B0 \ldots$ (2)

$B = A0 + C0$

$= B00 + C0$

$B = C0 + B00$ $\left[\begin{array}{l} R = B \\ Q = C0 \\ P = 00 \end{array}\right.$

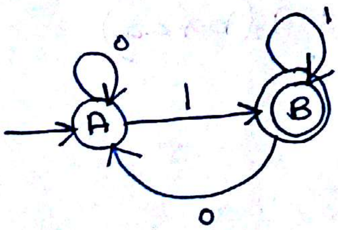
$\Rightarrow B = C0(00)^*$

$C = A1 + D0 + C1$

$C = B01 + B11^*0 + C1$ $\left[\begin{array}{l} R = C \\ Q = B01 + B11^*0 \\ P = 1 \end{array}\right.$

$C = (B01 + B11^*0)1^*$

2)



$A = \lambda + A0 + B0$

$B = A1 + B1$

$B = \boxed{A}11^*$

$B = A(0 + 11^*0)11^*$

$A = A0 + B0$

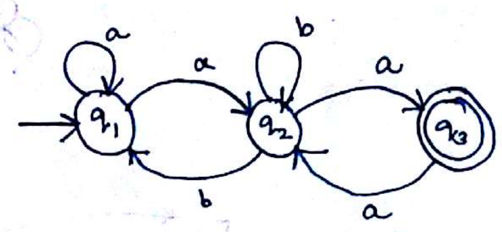
$= A0 + A11^*0$

$A = A(0 + 11^*0)$

Scanned by CamScanner

Use Arden's Theorem to find RE —

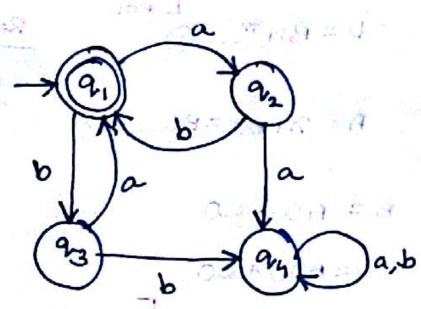
1.



$R = Q + RP$

$R = QP^*$

2.



1.

$q_1 = \lambda + q_1 a + q_2 b \ldots\ldots (1)$

$q_2 = q_1 a + q_2 b + q_3 a \ldots\ldots (2)$

$q_3 = q_2 a \ldots\ldots (3)$

Putting (3) in (2),

$q_2 = q_1 a + q_2 b + q_2 aa = q_1 a + q_2 (b + aa)$

$\therefore \quad R = Q + RP$

$R = q_2$

$Q = q_1 a$

$P = b + aa$

$\therefore q_2 = q_1 a (b + aa)^* \ldots (4)$

Putting (4) in (1),

$q_1 = \lambda + q_1 a + q_1 a (b + aa)^* b$

$= \lambda + q_1 (a + a(b + aa)^* b)$

Scanned by CamScanner

$\therefore R = q_1$

$Q = \lambda$

$P = a + a(b+aa)^* b$

$\therefore q_1 = \lambda(a+a(b+aa)^* b)^*$

$= (a + a(b+aa)^* b)^*$

From ①,

$\therefore q_2 = (a+a(b+aa)^* b)^* a(b+aa)^*$

From ③,

$\boxed{q_3 = (a+a(b+aa)^* b)^* a(b+aa)^* a}$ (Ans)

2.

$q_1 = \lambda + q_2 b + q_3 a$ --- ①

$q_2 = q_1 a$ --- ②

$q_3 = q_1 b$ --- ③

$q_4 = q_4 a + q_4 b + q_2 a + q_3 b$ --- ④

From ①,

$q_1 = \lambda + q_1 ab + q_1 ba$

$= \lambda + q_1(ab+ba)$

$R = q_1$

$Q = \lambda$

$P = ab+ba$

$\therefore q_1 = \lambda(ab+ba)^*$

$\Rightarrow \boxed{q_1 = (ab+ba)^*}$ (Ans)

Scanned by CamScanner

~~∴ From ②,~~

$$\cancel{q_2 = (ab+ba)^* a}$$

~~From ②,~~

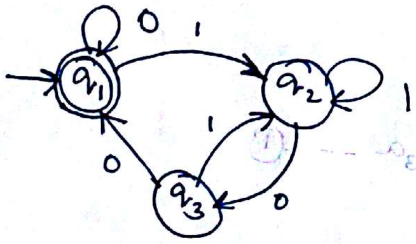
$$\cancel{q_3 = (ab+ba)^* b}$$

~~∴ From ⑤,~~

$$\cancel{q_4 =}$$

$$\therefore \cancel{q_1 = \lambda}$$

3.



$$q_1 = \lambda + q_1 0 + q_3 0 \quad \text{----} \; ①$$

$$q_2 = q_2 1 + q_1 1 + q_3 1 \quad \text{----} \; ②$$

$$q_3 = q_2 0 \quad \text{----} \; ③$$

$$\therefore \; q_1 = \lambda + q_1 0 + q_2 00 \quad \text{----} \; ④$$

From ②,

$$q_2 = q_2 1 + q_1 1 + q_2 01$$

$$\cancel{= q_2(1+01)}$$

$$\boxed{= q_1 1 + q_2(1+01)}$$

$$= q_1 1 (1+01)^*$$

From (4),

$$q_1 = \lambda + q_1 0 + q_2 00$$

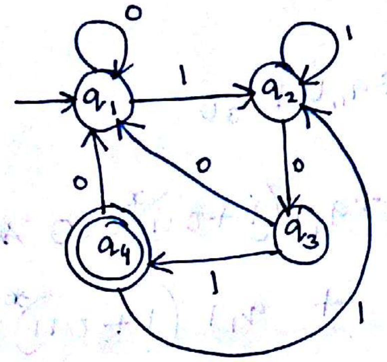
$$= \lambda + q_1 0 + q_1 1 (1+01)^* 00$$

$$= \lambda + q_1 \left(0 + 1 (1+01)^* 00\right)$$

$$= \lambda \left(0+1(1+01)^* 00\right)^*$$

$$\Rightarrow \boxed{q_1 = \left(0+1(1+01)^* 00\right)^*} \quad \text{(Ans)}$$

4.



$q_1 = q_1 0 + q_4 0 + q_3 0$ --- (1)

$q_2 = q_2 1 + q_1 1 + q_4 1$ --- (2)

$q_3 = q_2 0$ --- (3)

$q_4 = q_3 1$ --- (4)

∴ From (4),

$$q_4 = q_3 1$$

$$= q_2 01$$

Scanned by CamScanner

$q_2 = q_1 1 + q_2 1 + q_2 011$

$= q_1 1 + q_2 (1 + 011)$

$\Rightarrow q_2 = q_1 1 (1+011)^*$

$\therefore q_3 = q_1 1 (1+011)^* 0$ --- (4)

$\therefore q_4 = q_1 1 (1+011)^* 01$ ------ (5)

From (1),

$q_1 = \lambda + q_1 0 + q_4 0 + q_3 0$

$= \lambda + q_1 0 + q_1 1 (1+011)^* 010 + q_1 1 (1+011)^* 00$

$= \lambda + q_1 (0 + 1(1+011)^* 010 + 1(1+011)^* 00)$

$= \lambda (0 + 1(1+011)^* 010 + 1(1+011)^* 00)^*$

$= (0 + 1(1+011)^* 010 + 1(1+011)^* 00)^*$

$$\therefore \boxed{q_4 = (0 + 1(1+011)^* 010 + 1(1+011)^* 00)^* 1 (1+011)^* 01}$$

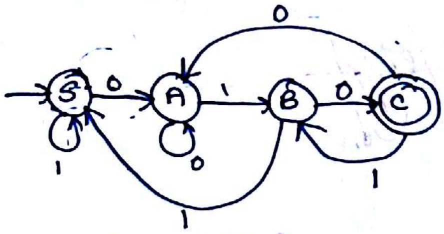
(Ans)

Scanned by CamScanner

1. $G = (V, \Sigma, P, S)$

Draw an FA, find the ~~grammar~~ grammer (G)

$L = \{x \in \{0,1\}^* \mid x \text{ ends in } 010\}$



right linear grammer

$P \begin{cases} S \to 1S \mid 0A \\ A \to 0A \mid 1B \\ B \to 0C \mid 1S \\ C \to 0A \mid 1B \mid \lambda \end{cases}$

$V = \{S, A, B, C\}$

$S = \{S\}$

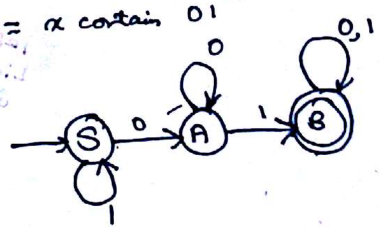
$\Sigma = \{0,1\}$

2. $G = (V, \Sigma, P, S)$

Draw an FA for $L = \{x \in \{0,1\}^* \mid x \text{ does not contain } 01 \text{ as substring}\}$.

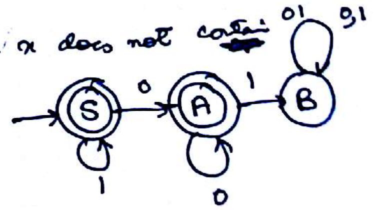
Then find the corresponding grammer.

$\overline{L}$ = x contain 01



complement to find

L = x does not contain 01



right linear grammer

$V = \{S, A, B\}$

$S = \{S\}$

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

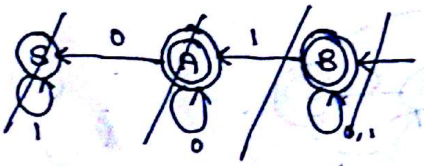
$P \begin{cases} S \to 1S \mid 0A \mid \lambda \\ A \to 0A \mid 1B \mid \lambda \\ B \to 0B \mid 1B \end{cases}$

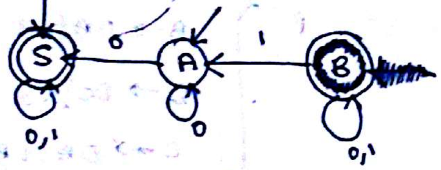
Scanned by CamScanner

To find the left linear grammar

- First draw L^R

Make the start state, the final state.
Make the ~~start~~ final state, the start state.

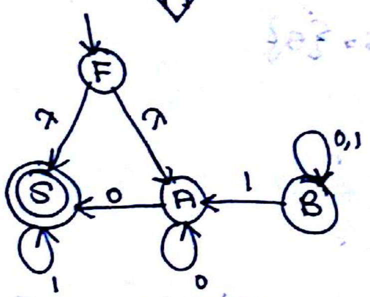




~~B → 0B | 1B | λ~~

~~A →~~

$\Downarrow$



$F \to S \mid A$

$S \to 1S \mid \lambda$

$A \to 0A \mid 0S$

$B \to \cancel{0}1A \mid 0B \mid 1B$

$\downarrow$ left linear

Left Linear Grammar

$F \to S \mid A$

$S \to S1 \mid \lambda$

$A \to A0 \mid S0$

$B \to A1 \mid B0 \mid B1$

(101) → not acceptable

(01) → not acceptable

S

S1

S11

11

Scanned by CamScanner

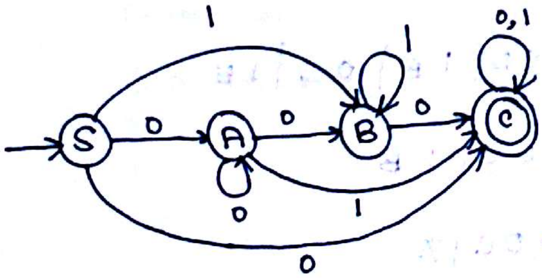
Given a grammar, draw the FA.

1. $S \rightarrow 0A \mid 0C \mid 1B$

   $A \rightarrow 0A \mid 1C \mid 0B$

   $B \rightarrow 0C \mid 1B$

   $C \rightarrow 0C \mid 1C \mid \lambda$



2. Construct regular grammar for all binary strings

$L = \{ x \in \{0,1\}^* \}$

Regular Expression: ~~0+~~

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

$P \{ S \rightarrow 0S \mid 1S \mid \lambda$

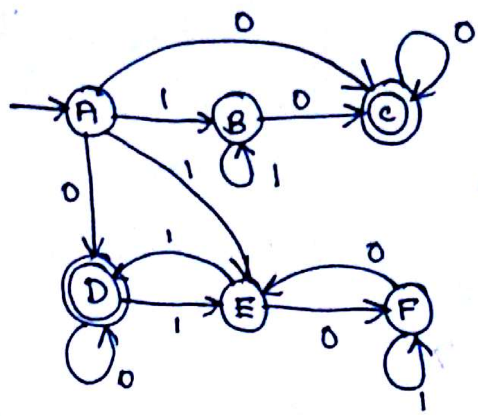
$V = \{ S \}$

$S = \{ S \}$

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

Scanned by CamScanner

3. Find regular grammer for



$$P \begin{cases} A \rightarrow 0D \mid 1B \mid 0C \mid 1E \\ B \rightarrow 0C \mid 1B \\ C \rightarrow 0C \mid \lambda \\ D \rightarrow 1E \mid 0D \mid \lambda \\ E \rightarrow 0F \mid 1E \\ F \rightarrow 0E \mid 1F \end{cases}$$

$V = \{A, B, C, D, E, F\}$

$S = \{A\}$

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

- A ~~ff~~ grammer is valid if we can ~~produce a~~ terminate with a particular ~~alt~~ string containing only alphabets.

Scanned by CamScanner

$S \to aSb \mid bA$

$A \to bB$

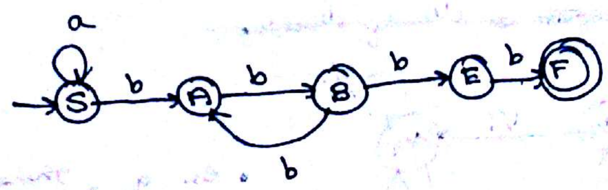
$B \to bb \mid bA$

- Valid Gramm~~er~~ ✓
- Non Regular Grammar, as $S \to aSb$, S lies in the middle, and we don't know where to go
- Finite Automata Cannot be drawn

- $S \to aS \mid bA$

  $A \to bB$

  $B \to bb \mid bA$

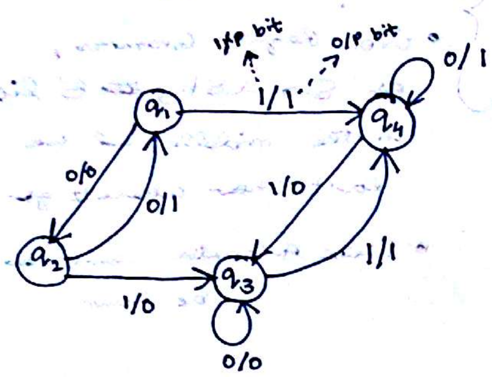


- Language with even number of b's, and starts with 4 b's

$f(\text{present state}, i/p) \to \underbrace{\text{next state}}_{o/p}$

- Output → 1 → Successfull State
- Output → 0 → Unsuccessful State

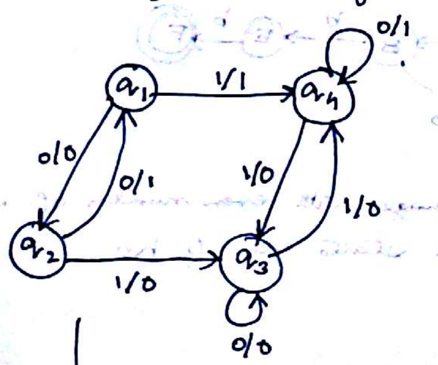
Mealy ~~Moore~~ Machine



Present State	Next State I/P = 0	O/P	Next State I/P = 1	O/P
q_1	q_2	0	q_4	1
q_2	q_1	1	q_3	0
q_3	q_3	0	q_4	1
q_4	q_4	1	q_3	0

Moore Machine

- ~~Same diagram as Mealy Machine.~~

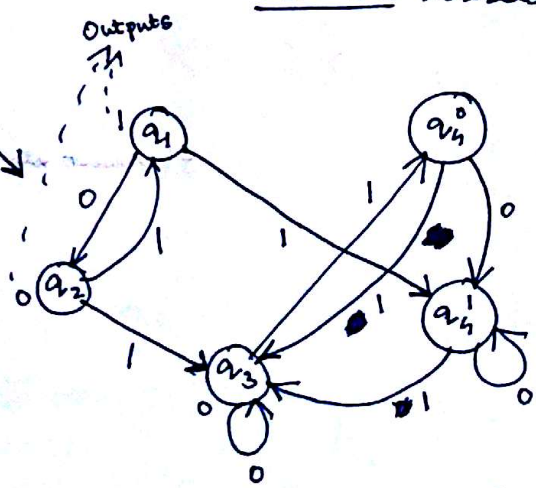


- In q_4, there are 3 incoming edges, 2 have output 1, and 1 has output 0.

∴ ambiguous.

Solution: Partition

Conversion from Mealy Machine to Moore Machine



Moore Machine

- q_1, q_4 are the acceptable states, as their output are 1.