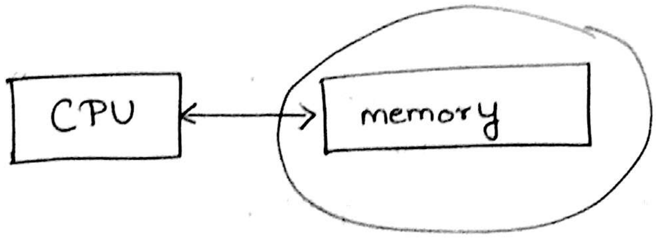
Theory of Computation

Computation:



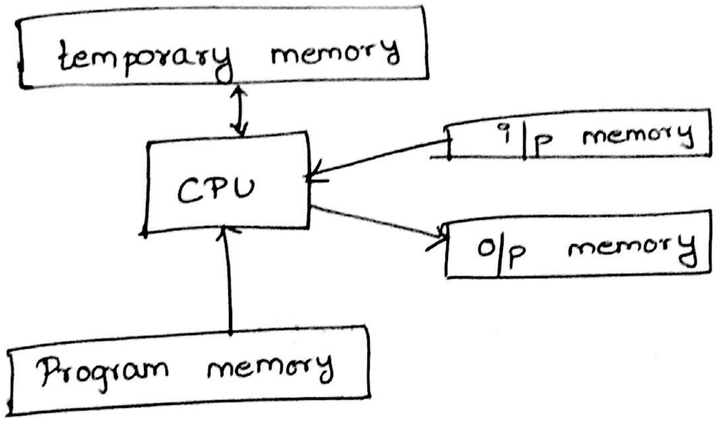
4

$z = 2 \times 2 = 4$

$f(x) = z \times 2$
$= 8$

Ex: $f(x) = x^3$

$x = 2$

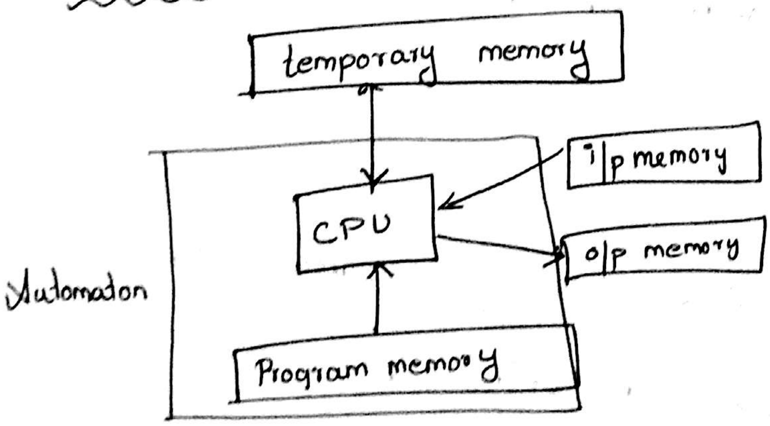


$f(x) = 8$

compute $x * x$

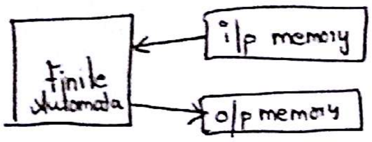
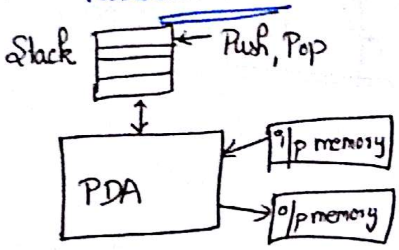
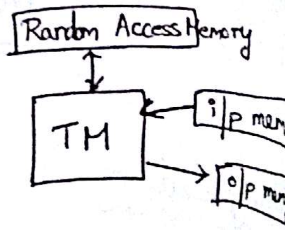
$x^2 * x$

Automaton:



Automata are distinguished by the temporary memory.

- Finite Automata : no temporary memory
- Pushdown Automata : stack
- Turing Machines : random access memory

Finite Automata	Pushdown Automata	Turing machine
		
Ex: Vending machines (small computing power)	Ex: Compilers for Prog. Languages (medium computing power)	Ex: Any Algorithm (Highest computing power)

Central Concepts of Automata Theory:

① Alphabets:

An alphabet is a finite non-empty set of symbols. (Usually denoted by Σ).

Ex: $\Sigma = \{0, 1\}$ => binary alphabet.

$\Sigma = \{a, b, \dots, z\}$ => lower-case letters.

Set of all ASCII characters, or the set of all printable ASCII characters.

② String:

A string over Σ is a sequence of symbols from Σ.

Ex: abbab, baabb are strings over $\Sigma = \{a, b\}$.

③ String Operations:

$w = a_1 a_2 \dots a_n$ abbab

$v = b_1 b_2 \dots b_m$ bbbaaab

Concatenation

$wv = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$ abbabbbbaaab

$w = a_1 a_2 \cdots a_n$ ababaaabbb

Reverse:

$w^R = a_n \cdots a_2 a_1$ bbbaaababa

String Length:

$w = a_1 a_2 \cdots a_n$

Length: $|w| = n$

$|abba| = 4$

$|aa| = 2$

$|a| = 1$

Length of Concatenation:

$|u \cdot v| = |u| + |v|$

$u = aab$, $|u| = 3$

$v = abaab$ $|v| = 5$

$|u \cdot v| = |aababaab| = 8$

$|u \cdot v| = |u| + |v| = 3+5 = 8$.

Empty String:

A string with no letters: λ

$|\lambda| = 0$.

$\lambda w = w \lambda = w$.

$\lambda abb = abb\lambda = abb$.

Substring:

a subsequence of consecutive characters.

String	Substring
abbab	ab
abbab	abba
abbab	b
abbab	bbab

Prefix and Suffix:

abbab

$w = uv$

prefix	suffix
λ	abbab
a	bbab
ab	bab
abb	ab
abba	b
abbab	λ.

Another Operation:

$$w^n = \underbrace{ww \dots w}_{n}$$

Ex: $(abba)^2 = abbaabba$

$$w^0 = \lambda$$

$$(abba)^0 = \lambda$$

The * Operation & $^+$ Operation

Σ^* : the set of all possible strings from alphabet Σ

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

$\Sigma^* = \{\underset{\in}{\underset{=}{\lambda}}, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$

$\Sigma^+ = \Sigma^* - \lambda$

$\Sigma^+ = \{a, b, aa, ab, ba, bb, \dots\}$.

Languages:

A language is any subset of Σ^*.

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

$\Sigma^* = \{\lambda, a, b, aa, \dots\dots\}$

Languages: $\{\lambda\}$

$\{a, aa, aab\}$

$\{\lambda, abba, baba, aa, ab, aaaaaa\}$

Note:

Sets $\phi = \{\ \} \neq \{\lambda\}$

Set size $|\{\ \}| = |\phi| = 0$

$|\{\lambda\}| = 1$

String length $|\lambda| = 0$

An infinite language

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

$\left.\begin{matrix}\lambda \\ ab \\ aabb\end{matrix}\right\} \in L$ $\qquad abb \notin L$

Operations on Languages:

$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$

$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$

$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$

Complement:

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

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \dots\}$$

Reverse:

$$L^R = \{w^R : w \in L\}$$

$$\{ab, aab, baba\}^R = \{ba, baa, abab\}$$

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

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

Concatenation:

$$L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$$

Ex: $\{a, ab, ba\}\{b, aa\}$

$= \{ab, aaa, abb, abaa, bab, baaa\}$

Another Operation:

$$L^n = \underbrace{LL \dots L}_{n}$$

$$\{a, b\}^3 = \{a, b\}\{a, b\}\{a, b\}$$

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

Special case: $L^0 = \{\lambda\}$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

Star Closure (Kleene *): (L^*)

$$L^* = L^0 \cup L^1 \cup L^2 \ldots$$

$$\{a, bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \ldots \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

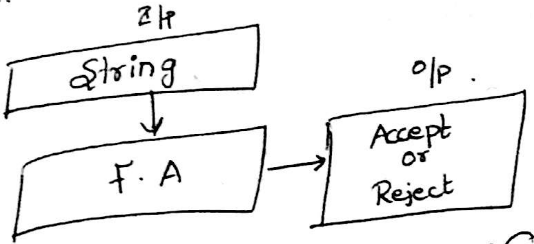
Positive Closure (L^+):

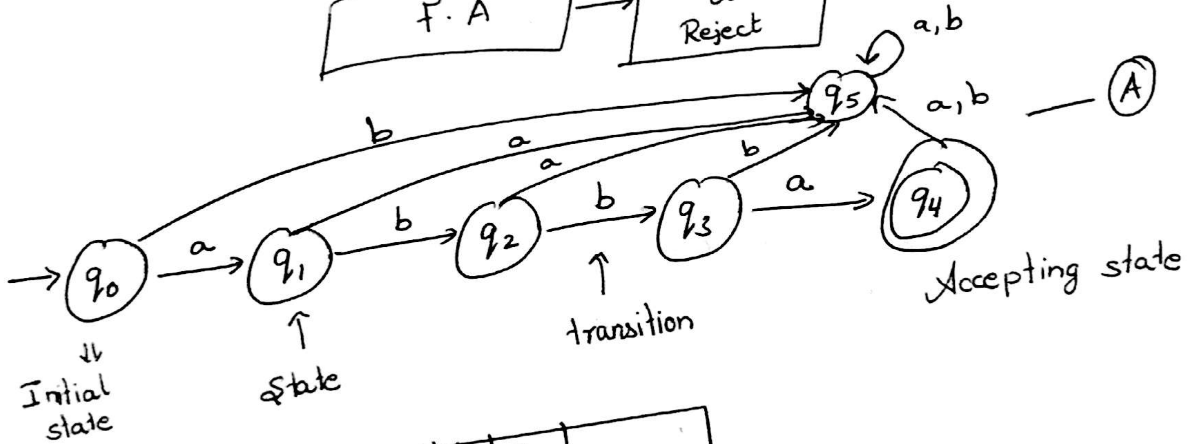
$$L^+ = L^1 \cup L^2 \cup \ldots$$

$$= L^* - \{\lambda\}$$

$$\{a, bb\}^+ = \left\{ \begin{array}{l} a, bb \\ aa, abb, bba, bbbb \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

Finite Automata:





I/P String

a	b	b	a	

q_0 q_1 q_2 q_3 $\underline{q_4}$
accept

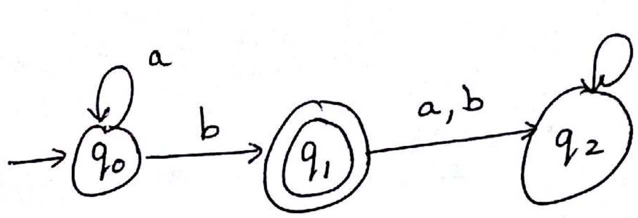
Acceptance

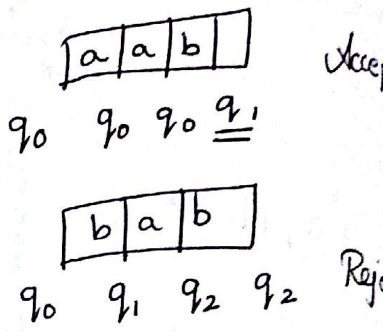
a	b	a	

q_0 q_1 q_2 q_5 = Reject

λ

reject





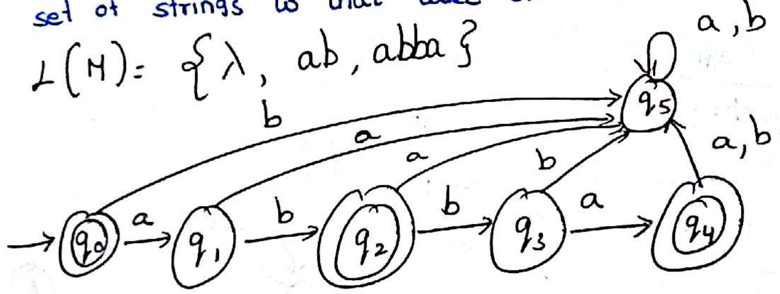
Languages Accepted by DFAs :

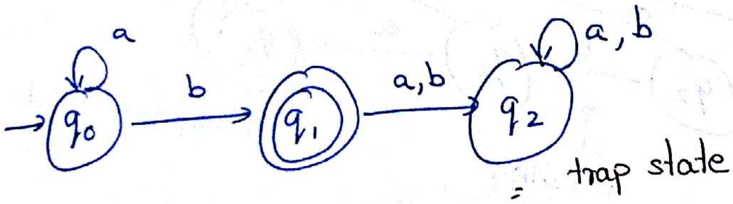
The language $L(M)$ contains all i/p strings accepted by M. (Regular Language)

$L(M)$ = { strings that bring M to an accepting state }

$= \{ w \mid \hat{\delta}(q_0, w)$ is in $F \}$ => the language of A is set of strings w that take start state q_0 to one of the accepting s

$L(M) = \{ \lambda, ab, abba \}$





$L(M) = ?$

$= \{ a^n b \; : n \geq 0 \}$

FA : < Deterministic / Non-deterministic

Det. Finite Automaton (DFA)

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

$Q \rightarrow$ set of states

$\Sigma \rightarrow$ i/p alphabet

$\delta \rightarrow$ transition f/n.

$q_0 \rightarrow$ initial state

$F \rightarrow$ set of accepting states

Refer (A):

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

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

$q_0 \rightarrow$ Initial State

$q_4 \rightarrow$ Final State

$\delta : Q \times \Sigma$ $\rightarrow Q$ $\rightarrow$ single state DFA

$\rightarrow 2^Q$ $\rightarrow$ set of states NFA

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_5$

Transition Table:

δ	a	b
$\rightarrow q_0$	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
$* q_4$	q_5	q_5
q_5	q_5	q_5

Extended Transition Function δ^*:

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

(1).

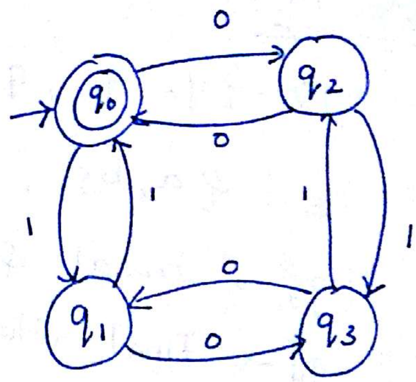
$\delta^*(q_0, ab) = q_2$

$\delta^*(q_0, abba) = q_4$

$\delta^*(q_0, abbbaa) = q_5$

②.

δ	0	1
$* \rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2



$w = 110101$

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

$\hat{\delta}(q_0, 1) = \delta(\hat{\delta}(q_0, \epsilon), 1) = \delta(q_0, 1) = q_1$

$\hat{\delta}(q_0, 11) = \delta(\hat{\delta}(q_0, 1), 1) = \delta(q_1, 1) = q_0$

$\hat{\delta}(q_0, 110) = \delta(\hat{\delta}(q_1, 1), 0) = \delta(q_0, 0) = q_2$

$\hat{\delta}(q_0, 1101) = \delta(\hat{\delta}(q_0, 0), 1) = \delta(q_2, 1) = q_3$

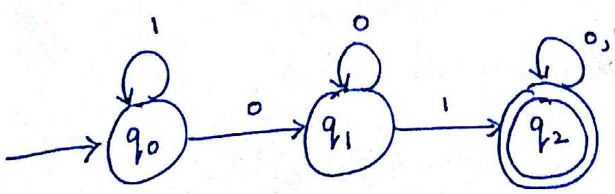
$\hat{\delta}(q_0, 11010) = \delta(\hat{\delta}(q_2, 1), 0) = \delta(q_3, 0) = q_1$

$\hat{\delta}(q_0, 110101) = \delta(\hat{\delta}(q_3, 0), 1) = \delta(q_1, 1) = q_0$

Construction of DFA:

①. Construct a DFA that accepts all & only the strings of 0's & 1's that have the sequence 01 somewhere in the string

$L = \{ x01y \mid x$ and y are any strings of 0's & 1's $\}$



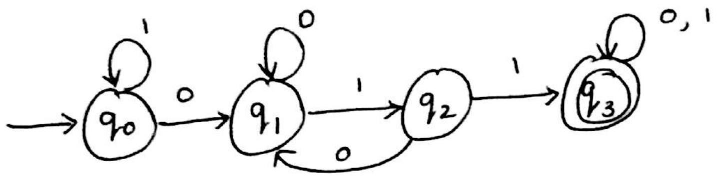
(2). Construct a DFA to accept strings over {0, 1} containing

(a). 011 as a substring.

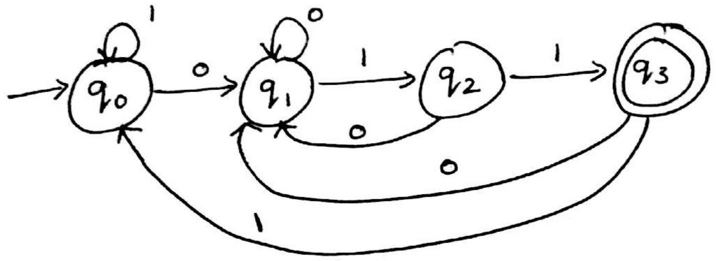
(b). 011 as a ending string.

(c). 011 as a starting string.

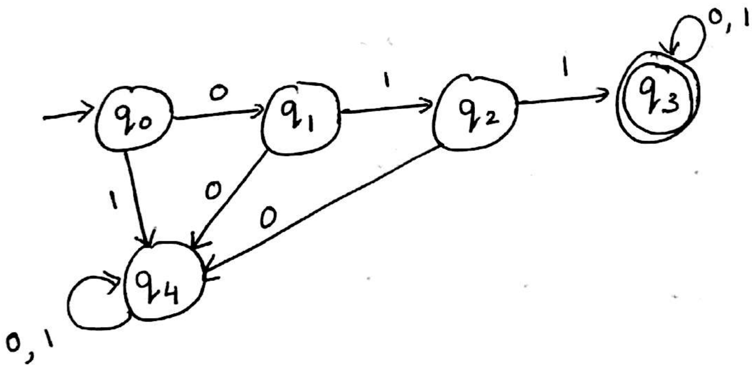
(a).



(b).



(c).



Hint:
1. Logic
2. Formal notations
3. Transition function (table or diagram)
4. Testing the Automata

} To design any FSM / FA

(3) Logic for number divisible by 3

* if we get remainder 0 after dividing a no. by 3
↓
then that no. is divisible by 3

Step 1. $0/3 = 3/3 = 6/3 = 0$ remainder $\rightarrow q_0$ $\Rightarrow$ consider q_0 End state since we wa remainder

$1/3 = 4/3 = 7/3 = 1$ remainder $\rightarrow q_1$

$2/3 = 5/3 = 8/3 = 2$ remainder $\rightarrow q_2$

Step 2:

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

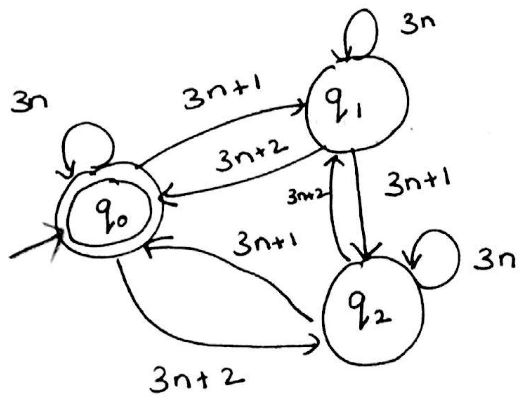
$\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$.

δ = (Delta) = Transition function.

q_0 = start state.

$F = \{q_0\}$ = end state.

		$3n$ (0,3,6,9)	$3n+1$ (1,4,7)	$3n+2$ (2,5,8)
rem 0	$\rightarrow q_0^*$	q_0	q_1	q_2
rem 1	q_1	q_1	q_2	q_0
rem 2	q_2	q_2	q_0	q_1



Testing the Automata:

i/p string 12345

$\hat{\delta}(q_0, 12345)$

$\hat{\delta}(q_0, 1) = q_1$

$\hat{\delta}(q_0, 12) = \delta(\hat{\delta}(q_0, 1), 2) = \delta(q_1, 2) = q_0$

$\hat{\delta}(q_0, 123) = \delta(\hat{\delta}(q_0, 12), 3) = \delta(q_0, 3) = q_0$

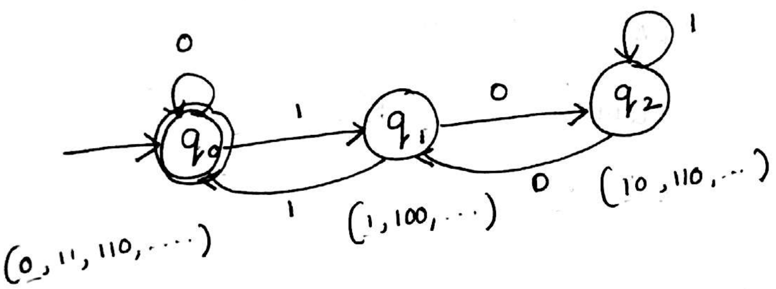
$\hat{\delta}(q_0, 1234) = \delta(\hat{\delta}(q_0, 123), 4) = \delta(q_0, 4) = q_1$

$\hat{\delta}(q_0, 12345) = \delta(\hat{\delta}(q_0, 1234), 5) = \delta(q_1, 5) = q_0$.

Given no is ~~divisible~~ by 3

(4). Construct a minimal DFA which accepts set of all strings over $\{0,1\}$ which when interpreted as a binary number is divisible by 3.

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

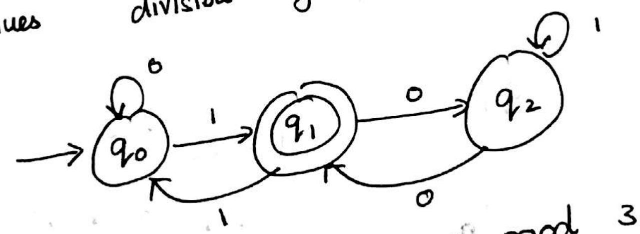


$q_0 \rightarrow$ rem 0

$q_1 \rightarrow$ rem 1

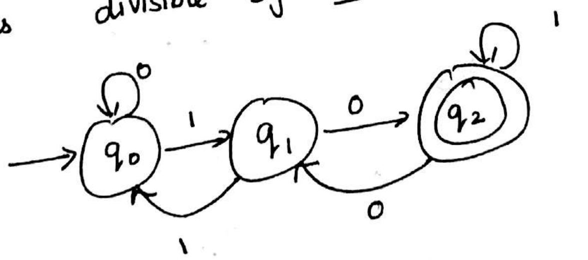
$q_2 \rightarrow$ rem 2

(4) continues divisible by $\cong$ 1 mod 3



	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

(4) continues divisible by $\cong$ 2 mod 3.



(4) continues divisible by n

The No. of states will be n.

5. Construct a minimal DFA which accepts set of all strings over {0, 1} which when interpreted as a binary number is divisible by 4.

	0	1
$\rightarrow {}^{*}q_0$	q_0^*	q_1
q_1	q_2	q_3
q_2	q_0^*	q_1
q_3	q_2	q_3

or

	0	1
$\rightarrow * q_0$	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_1

6. Construct a minimal DFA which accepts set of all strings over {0, 1, 2} which when interpreted as a binary number is divisible by 4.

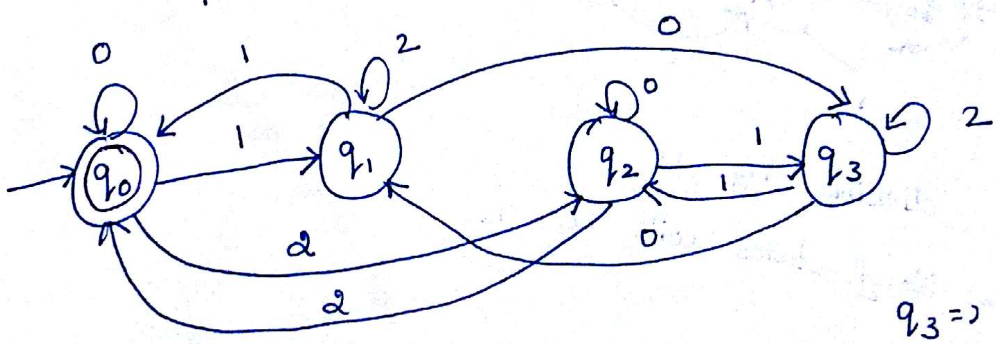
	0	1	2
$\rightarrow {}^{*}q_0$	q_0	q_1	q_2
q_1	q_3	q_0	q_1
q_2	q_2	q_3	q_0
q_3	q_1	q_2	q_3

q_1

$(10)_3 = 3$

$(11)_3 = (4)_{10}$

$(12)_3 = (5)_{10}$



$q_3 \Rightarrow (30)_3 = 9$

$(31)_3 = 10$

$(32)_3 = 11$

$q_2 \Rightarrow (20)_3 = 6$

$(21)_3 = 7$

$(22)_3 = 8$

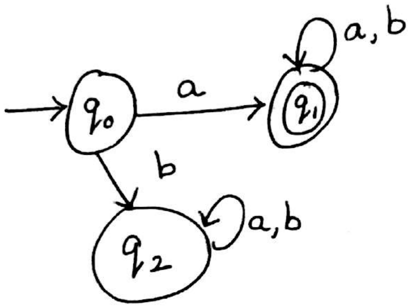
Hint :

Base k number $\{0, 1, 2, \ldots k-1\}$ => Alp

Divisible by m => No. of. states

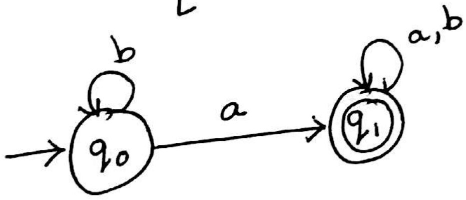
(7). Construct a minimal DFA which accepts set of all strings over $\{a, b\}$ where each string starts with an 'a'.

$L = \{a, aa, ab, aaa, \ldots\ldots\}$



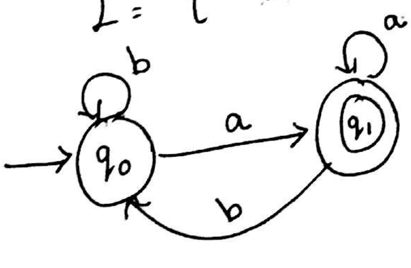
(8). Construct a min. DFA set of all strings containing 'a'.

$L = \{a, ab, ba, aa, abb, bab, \ldots\}$



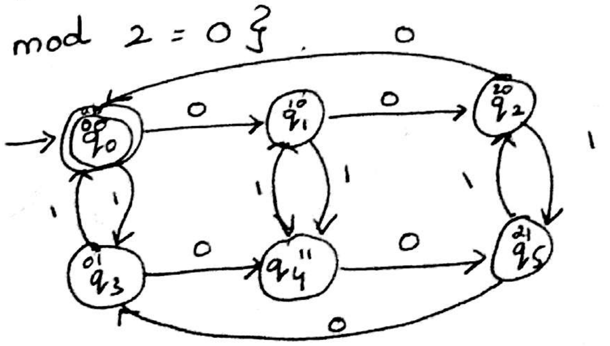
(9). Ending with 'a'.

$L = \{a, ba, bba, aa, aaa, abaa, \ldots\}$



(10). Construct a DFA to accept strings of 0's & 1's such that $L = \{w \mid w \in (0+1)^*$ such that $N_0(w) \bmod 3 = 0$, $N_1(w) \bmod 2 = 0\}$

(3x2 = 6 states)

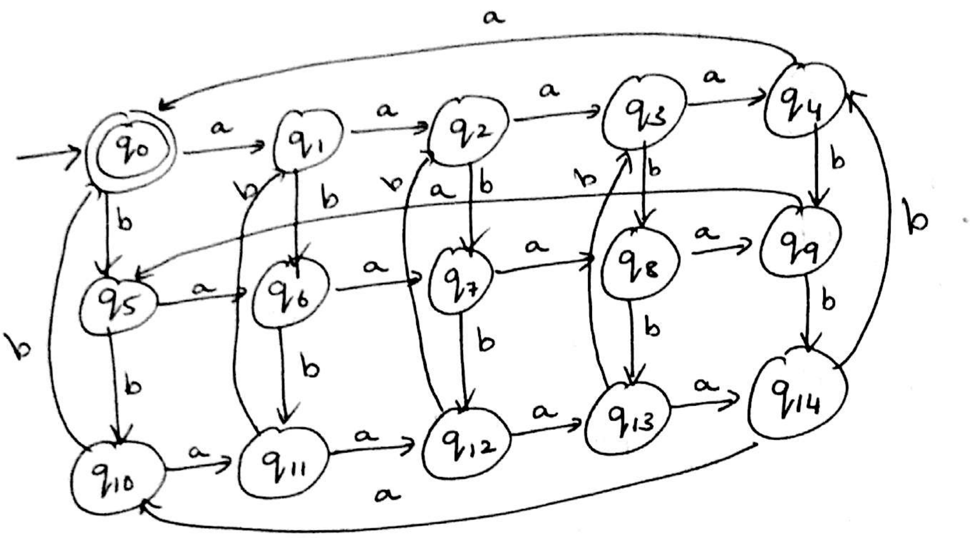


(11). Construct a DFA to accept strings of a's and b's such that the number of a's is divisible by 5 & number of b's is divisible by 3.

(5 x 3 = 15 states)

(a). $L = \{ w / w \in (a+b)^* \ Na(w) \bmod 5 = 0$ & $Nb(w) \bmod 3 = 0 \}$ ⇒ Final state q_0

(b). $L = \{ w / w \in (a+b)^* \ Na(w) \bmod 5 = 2$ & $Nb(w) \bmod 3 = 1 \}$. → q_7.



(12). Construct a DFA accepting the foll. language

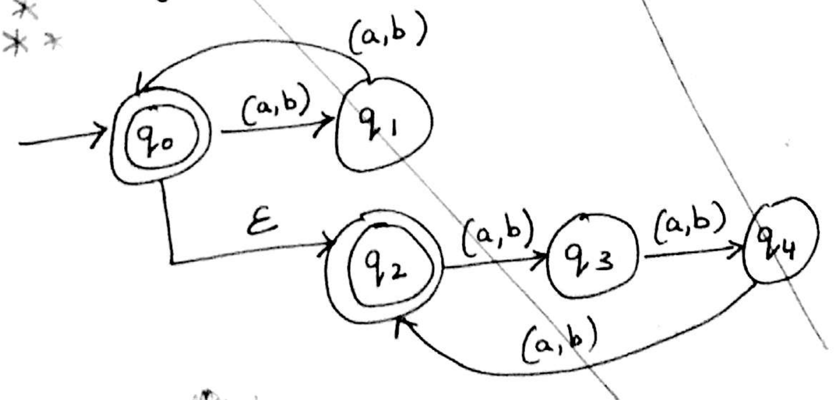
$L = \{ w$ such that $|w| \bmod 3 \neq |w| \bmod 2$

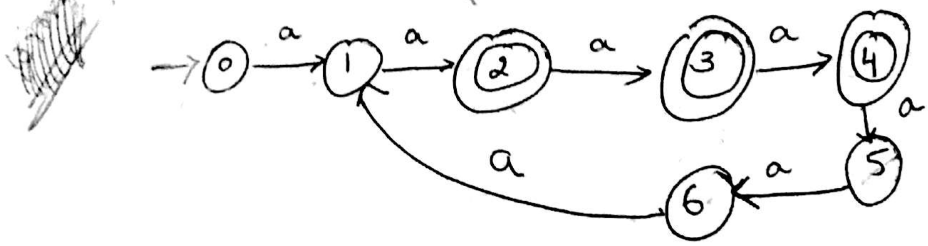
where $w \in \Sigma = \{a\}^*$.

(multiple of 3 or 2 but not both)

wrong

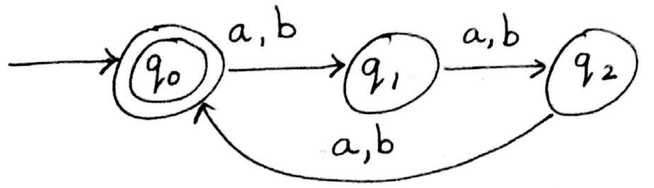
* *
*





(13). Design the DFA's for the language

$L = \{ \omega : |\omega| \bmod 3 = 0 \quad \omega \in (a+b)^* \}$

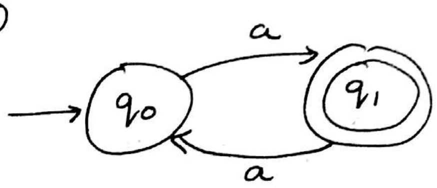


(14) Construct a DFA to accept ① ~~even~~ odd no. of a's and

② even no. of a's. $\Sigma = \{a\}$.

②

①



(15) Construct a DFA to accept

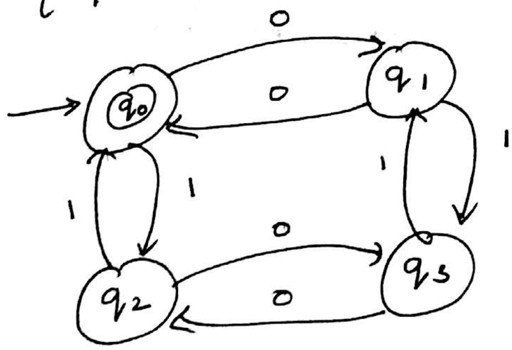
ⓐ. $L = \{ \omega / \omega$ has both an even no. of 0's & even no. of 1's$\}$

ⓑ. $L = \{ \omega / \omega$ has both an even no. of 0's & odd no. of 1's$\}$

ⓒ. $L = \{ \omega / \omega$ has both odd no. of 0's & even no. of 1's$\}$.

ⓓ. $L = \{ \omega / \omega$ has both ~~odd~~ odd no. of 0's & odd no. of 1's$\}$.

ⓐ.



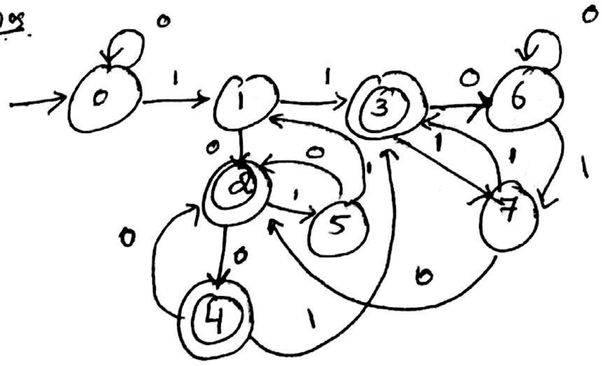
To $q_0 \rightarrow$ ⓐ

$q_1 \rightarrow$ ⓒ

$q_2 \rightarrow$ ⓑ

$q_3 \rightarrow$ ⓓ

for binary nos.



11

110

11001

Nondeterministic Finite Automata: (exhaustive search & back-tracking)

NFA can be defined as $(Q, \Sigma, \delta, q_0, F)$.

$Q \rightarrow$ finite set of states.

$\Sigma \rightarrow$ finite set of i/p symbols.

$q_0 \rightarrow$ member of Q, start state

$F \rightarrow$ subset of Q, is the set of final (or accepting) states

$\delta \rightarrow$ transition function

It takes a state in Q and an i/p symbol in Σ as arguments & returns a subset of Q.

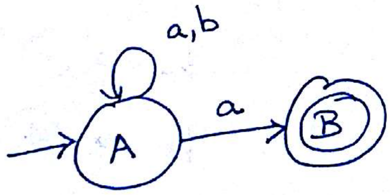
(set of states out of which atleast one state as final

$$Q \times \Sigma \rightarrow 2^{Q}.$$
(Power set of Q)

$Q = \{q_0, q_1\}$

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

① $L_1 = \{$strings ends with 'a'$\}$



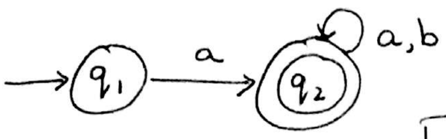
$A \xrightarrow{a} A, B$

$A \xrightarrow{b} A$

$B \xrightarrow{a} \phi$

$B \xrightarrow{b} \phi$

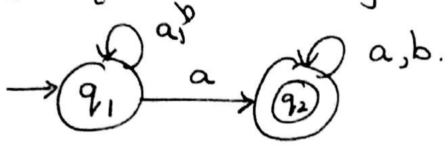
② $L_2 = \{$ starts with 'a' $\}$ $\Sigma = \{a,b\}$



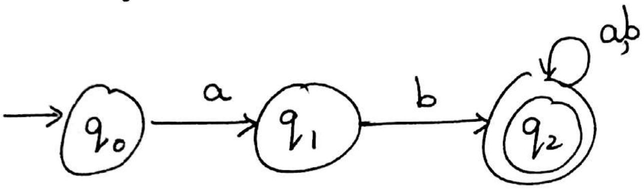
$\boxed{q_1 \xrightarrow{b} \phi \quad \text{Dead configuration}}$

In DFA we used trap state

③ $L_3 = \{$ contains 'a' $\}$



④ $L_4 = \{$ starts with 'ab' $\}$



abaab

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

$\hat{\delta}(q_0, a) = q_1$

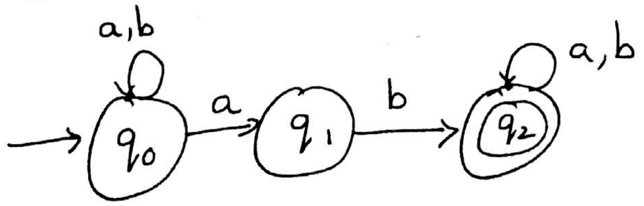
$\hat{\delta}(q_0, ab) = \delta(q_1, b) = q_2$

$\hat{\delta}(q_0, aba) = \delta(q_2, a) = q_2$

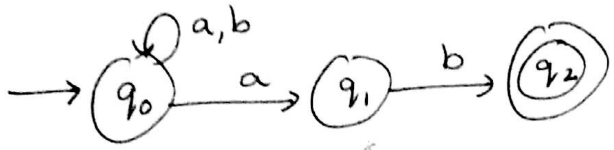
$\hat{\delta}(q_0, abaa) = \delta(q_2, a) = q_2$

$\hat{\delta}(q_0, abaab) = \delta(q_2, b) = \underline{\underline{q_2}}$ · accepting state

⑤ $L_5 = \{$ contains 'ab' $\}$.



⑥. $L_6 = \{$ strings ending with 'ab' $\}$.



aabab

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

$\hat{\delta}(q_0, a) = \delta(q_0, a) = \{q_0, q_1\}$

$\hat{\delta}(q_0, aa) = \delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \phi$

$\hat{\delta}(q_0, aab) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_0\} \cup \{q_2\}$
$= \{q_0, q_2\}$

$\hat{\delta}(q_0, aaba) = \delta(q_0, a) \cup \delta(q_2, a) = \{q_0, q_1\} \cup \phi$
$= \{q_0, q_1\}$

$\hat{\delta}(q_0, aabab) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_0\} \cup \{q_2\}$
$= \{q_0, \underline{\underline{q_2}}\}$ accepting state

Language of an NFA:

If $A = (Q, \Sigma, \delta, q_0, F)$ is an NFA, then

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

$L(A)$ is the set of strings w in Σ^* such that $\hat{\delta}(q_0, w)$ contains at least one accepting state

Scanned by CamScanner

Equivalence of Deterministic and Non-deterministic Finite Automata

Which is powerful?

(Both DFA & NFA are equally powerful).

Conversion of NFA $\longrightarrow$ DFA $\Rightarrow$ one of the ways

Subset construction method.

NFA $\qquad\qquad$ DFA

$$N = (Q_N, \Sigma, \delta_N, q_0, F_N) \xRightarrow{\text{Goal}} D = (Q_D, \Sigma, \delta_D, q_0, F_D)$$

such that $L(D) = L(N)$.

$Q_D \rightarrow$ set of subsets of Q_N

if n states then Q_D will have worst case 2^n states.

$F_D \rightarrow$ is all sets of N's states that include atleast one accepting state of N.

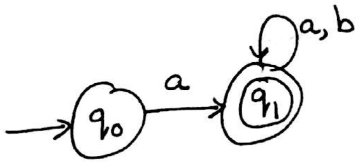
S $\Rightarrow$ set of subsets.

For each set $S \subseteq Q_N$ and for each i/p symbol a in Σ,

$$\delta_D(S, a) = \bigcup_{p \text{ in } S} \delta_N(p, a)$$

1. $L_1 = \{$starts with 'a'$\}$. $\Sigma = \{a, b\}$

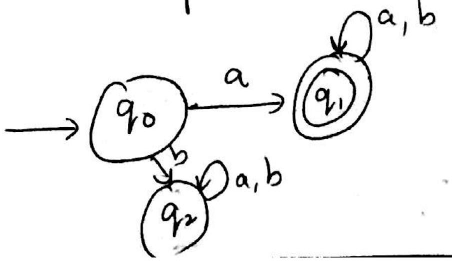
NFA



	a	b
$\rightarrow q_0$	q_1	ϕ
$* q_1$	q_1	q_1

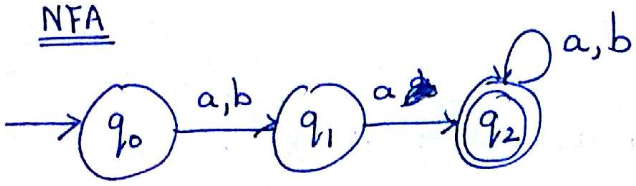
DFA

	a	b
$\rightarrow q_0$	q_1	q_2
$q_1 *$	q_1	q_1
trap state $\Leftarrow q_2$	q_2	q_2



(2). L_2 = { all strings in which second symbol from LHS is 'a' } $\Sigma = \{a, b\}$

NFA



	a	b
$\rightarrow q_0$	$\{q_1\}$	$\{q_1\}$
q_1	$\{q_2\}$	ϕ
q_2^*	$\{q_2\}$	$\{q_2\}$

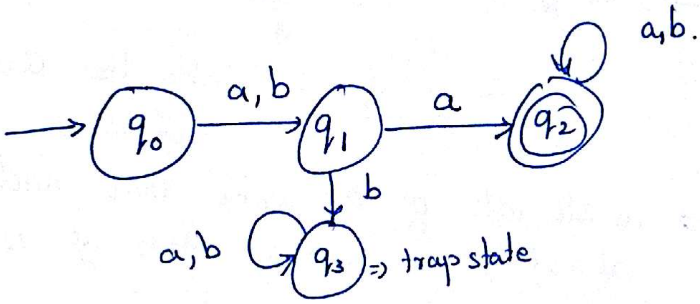
DFA

	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_3
q_2^*	q_2	q_2
q_3	q_3	q_3

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_1$

$\delta(q_1, a) = q_2$

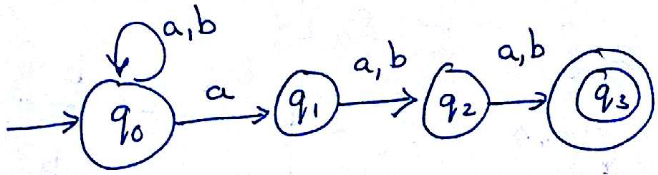


(3). L_3 = { strings in which 3^{rd} symbol from RHS is 'a' }

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

L = { [. ...] a [. | .], ... }

NFA



	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$
q_3^*	ϕ	ϕ

Lazy Evaluation

① q_0 on 'a' $\{q_0, q_1\}$

② $\delta(\{q_0, q_1\}, a) = \{q_0, q_1\} \cup q_2$; $\delta(\{q_0, q_1\}, b) =$
$= \{q_0, q_1, q_2\}$ $\{q_0, q_2\}$

q_0 on 'b' $\{q_0\}$

③. q_1 on 'a' $\{q_2\}$; q_1 on 'b' $\{q_2\}$

④. q_2 on 'a' $\{q_3\}$; q_2 on 'b' $\{q_3\}$

⑤. $\{q_0, q_1, q_2\}$ on 'a' & 'b'

$\delta(\{q_0, q_1, q_2\}, a) = \{q_0, q_1, q_2, q_3\}$

$\delta(\{q_0, q_1, q_2\}, b) = \{q_0, q_2, q_3\}$

⑥. $\{q_0, q_2, q_3\}$ on 'a' & 'b'

$\delta(\{q_0, q_2, q_3\}, a) = \{q_0, q_1, q_3\}$

$\delta(\{q_0, q_2, q_3\}, b) = \{q_0, q_3\}$

⑦. $\{q_0, q_2\}$ on 'a' & 'b'

$\delta(\{q_0, q_2\}, a) = \{q_0, q_1, q_3\}$

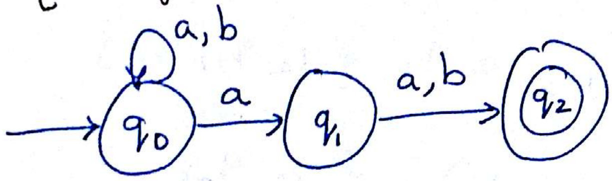
$\delta((\{q_0, q_2\}, b) = \{q_0, q_3\}$.

⑧. $\{q_0, q_3\}$ on 'a' & 'b'

$\delta(\{q_0, q_3\}, a) = \{q_0, q_1\}$; $\delta(\{q_0, q_3\}, b) = \{q_0\}$

$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$	$\{q_0, q_3\}$
$\{q_0, q_3\}^*$	$\{q_0, q_1\}$	$\{q_0,$ [illegible] $\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_3\}$
$\{q_0, q_1, q_3\}^*$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_1, q_2, q_3\}^*$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_2, q_2\}$
$\{q_0, q_2, q_3\}^*$	$\{q_0, q_1, q_3\}$	$\{q_0, q_3\}$

(4). L_4 = { strings inwhich second symbol from RHS is "a" }



NFA

	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
q_1	q_2	q_2
q_2^*	ϕ	ϕ

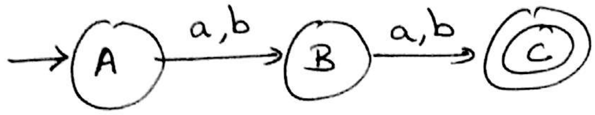
DFA

	a	b
q_0	$\{q_0, q_1\}$	q_0
~~q_1~~	~~q_2~~	~~q_2~~
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}^*$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1, q_2\}^*$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$

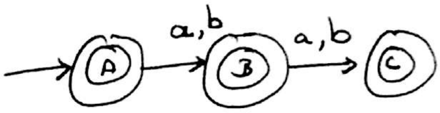
* Problems on NFA $L = \{$strings of length $\Sigma = \{a, b\}$

(a). exactly 2 (b) atmost 2 (c) atleast 2

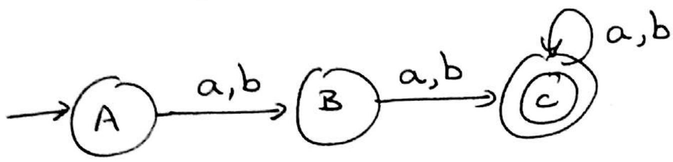
(a). $L = \{aa, ab, ba, bb\}$



(b). $L = \{\epsilon, a, b, aa, ab, ba, bb\}$



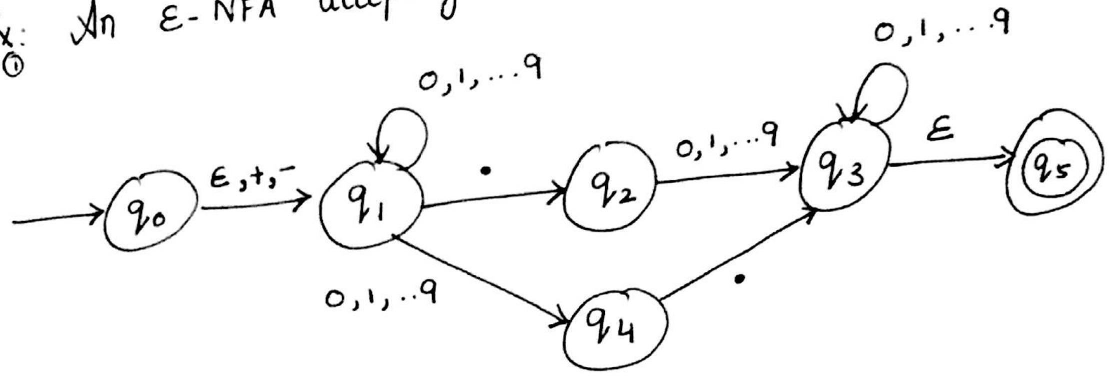
(c). $L = \{aa, ab, ba, bb, aa \ldots\}$



Complimenting NFA and DFA :

Epsilon NFA:

Ex: ① An ε-NFA accepting decimal numbers.



	ε	+, -	•	0, 1, ... 9
→ q_0	$\{q_1\}$	$\{q_1\}$	ϕ	ϕ
q_1	ϕ	ϕ	$\{q_2\}$	$\{q_1, q_4\}$
q_2	ϕ	ϕ	ϕ	$\{q_3\}$
q_3	$\{q_5\}$	ϕ	ϕ	$\{q_3\}$
q_4	ϕ	ϕ	$\{q_3\}$	ϕ
q_5^*	ϕ	ϕ	ϕ	ϕ

Language:

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

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

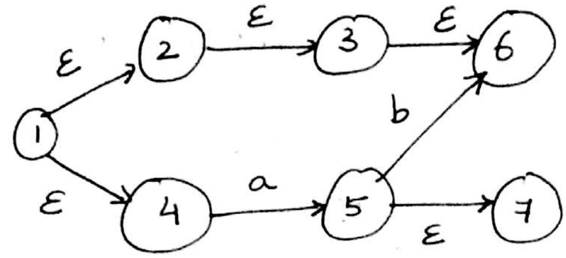
$\Downarrow$

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

$ECLOSE(q_0) = \{q_0, q_1\}$

$ECLOSE(q_3) = \{q_3, q_5\}$

Ex②



ECLOSE(1) = {1, 2, 3, 6, 4}

Extended Transition function.

5.67

$\hat{\delta}(q_0, \epsilon) = \text{ECLOSE}(q_0) = \{q_0, q_1\}$.

$\hat{\delta}(q_0, 5) =$

$\delta(\{q_0, q_1\}, 5) = \delta(q_0, 5) \cup \delta(q_1, 5)$

$= \phi \cup \{q_1, q_4\}$

$\hat{\delta}(q_0, 5.) =$

$\delta(\{q_1, q_4\}, 5.) = \delta(q_1, .) \cup \delta(q_4, .)$

$= \{q_2\} \cup \{q_3, q_5\}$

$= \{q_2, q_3, q_5\}$

$\hat{\delta}(q_0, 5.6) =$

$\delta(\{q_2, q_3, q_5\}, 5.6) = \delta(q_2, 6) \cup \delta(q_3, 6) \cup \delta(q_5, 6)$

$= \{q_3\} \cup \{q_3, q_5\} \cup \phi$

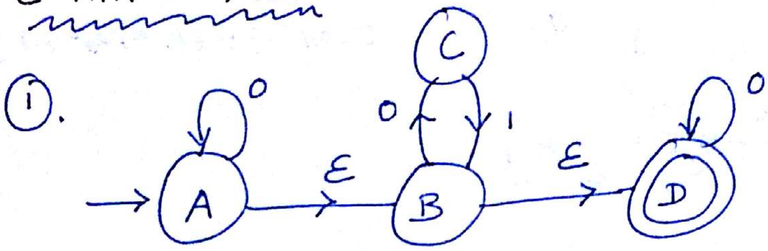
5.678

	ECLOSE	Σ 5.678 ⑤
Eclose (q_0)	$\{q_0, q_1\}$	$\{q_1, q_4\}$ ⊙
Eclose (q_1, q_4)	$\{q_1, q_4\}$	$\{q_2, q_3\}$ ⑥
Eclose (q_2, q_3)	$\{q_2, q_3, q_5\}$	$\{q_3\}$ ⑦
Eclose (q_3)	$\{q_3, q_5\}$	$\{q_3\}$ ⑧
Eclose (q_3)	$\{q_3, q_5\}$	$\{q_3\}$
Eclose (q_3)	$\{q_3, \underline{\underline{q_5}}\}$ accepting state.	

ε-NFA ⟶ NFA

DFA ≅ NFA ≅ ε-NFA

①.



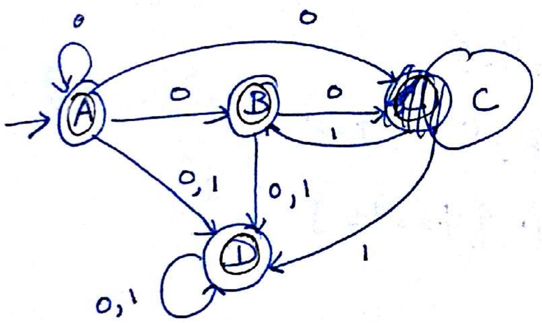
T.F for ε-NFA → NFA

NFA

	0	1
→ A	A, B, C, D	D
B	C, D	D
C	φ	B, D
D*	D	D

ε-closure(δ(ε-closure(A), 0))

	ε*	0	ε*
A	A, B, D	A, C, D	A, B, C, D
B	B, D	C, D	C, D
C	C	φ	φ
D	D	D	D

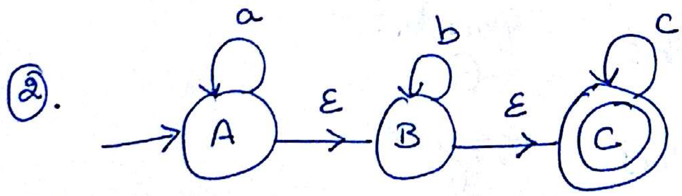


Final States A, B, D

ε-closure(δ(ε-closure(A), 1))

	ε*	1	ε*
A	A, B, D	D	D
B	B, D	D	D
C	C	B	B, D
D	D	D	D

②.



NFA

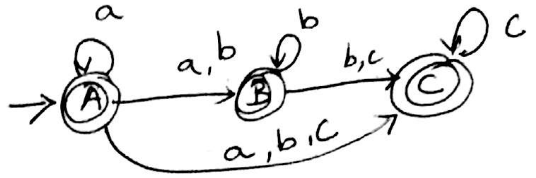
	a	b	c
→ A	{A, B, C}	{B, C}	{C}
B	φ	{B, C}	{C}
C*	φ	φ	{C}

	ε*	a	ε*
A	A, B, C	A	A, B, C
B	B, C	φ	φ
C	C	φ	φ

	ε*	c
		c
A	A, B, C	c
B	B, C	c
C	c	c

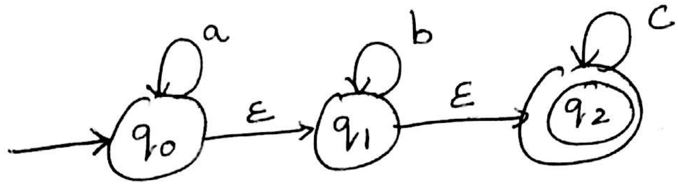
	ε*	b	ε*
A	A, B, C	B	B, C
B	B, C	B	B, C
C	C	φ	φ

Scanned by CamScanner



ϵ - NFA $\longrightarrow$ DFA

1.



Step 1:

Eclosure of initial state

And then follow sub set construction.

$\epsilon CLOSE(q_0) = \{q_0, q_1, q_2\}$ —— A

A on a

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

on Eclose

$Eclose(q_0) = \{q_0, q_1, q_2\}$ A

A on b

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

on Eclose

$Eclose(q_1) = \{q_1, q_2\}$ B

A on c

$\delta(\{q_0, q_1, q_2\}, c) = q_2$

on Eclose

$Eclose(q_2) = \{q_2\}$ C

B on a

$\delta(\{q_1, q_2\}, a) = \phi$

on Eclose

ϕ

B on b

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

on Eclose

$Eclose(q_2) = \{q_1, q_2\}$ B

B on c

$\delta(\{q_1, q_2\}, c) = q_2$

on Eclose

$Eclose(q_2) = \{q_2\}$ C

C on a ϕ

$\epsilon.close(\phi) = \phi$

C on b ϕ

$\epsilon.close(\phi) = \phi$

C on c C

$\epsilon.close\ \phi = C$

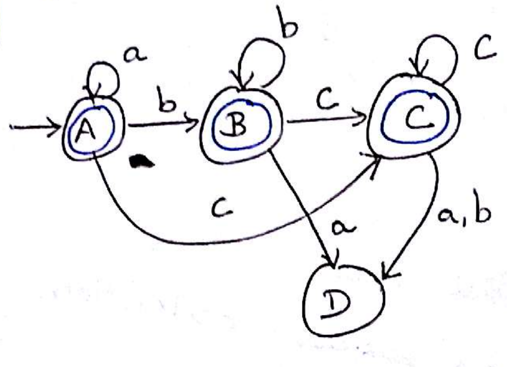
	a	b	c
A	A	B	C
B	ϕ	B	C
C	ϕ	ϕ	C

$A \rightarrow q_0, q_1, q_2$

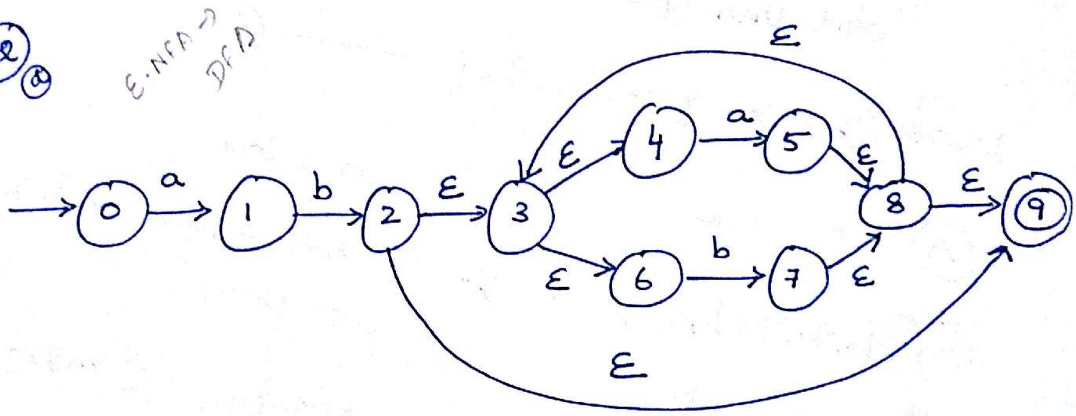
$B \rightarrow q_1, q_2$

$C \rightarrow q_2$.

$D \rightarrow \phi$



(2) (a) ε-NFA → DFA



$\varepsilon\text{-close}(0) = 0$ (A)

A on a $\quad$ ε-close

$\delta(A, a) = 1$ $\quad$ $\varepsilon\text{-close}(1) = 1$ (B)

A on b

$\delta(A, b) = \phi$ $\quad$ $\varepsilon\text{-close}(\phi) = \phi$

B on a $\quad \delta(B, a) = \phi$ $\quad$ $\varepsilon\text{-close}(\phi) = \phi$

B on b $\quad \delta(B, b) = 2$ $\quad$ $\varepsilon\text{-close}(2) = \{2, 3, 4, 6, 9\}$ (C)

C on a $\quad \delta(C, a) = 5$ $\quad$ $\varepsilon\text{-close}(5) = \{5, 8, 3, 4, 6, 9\}$ (D)

C on b $\quad \delta(C, b) = 7$ $\quad$ $\varepsilon\text{-close}(7) = \{7, 8, 9, 3, 4, 6\}$ (E)

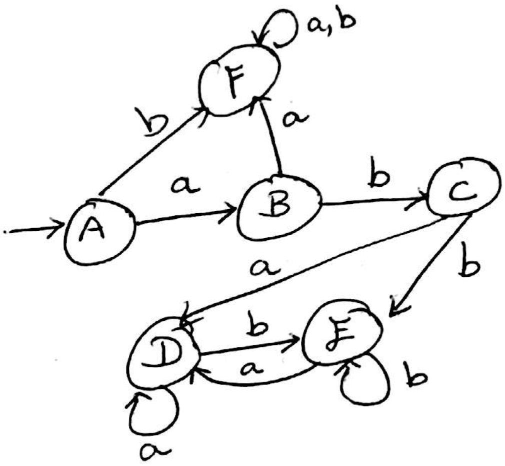
D on a $\quad \delta(D, a) = 5 \quad \varepsilon\text{-close}(5) = \{5,8,3,4,6,9\}$ D

D on b $\quad \delta(D, b) = 7 \quad \varepsilon\text{-close}(7) = \{7,8,9,3,4,6\}$ E

E on a $\quad \delta(E, a) = 5 \quad \varepsilon\text{-close}(5) = \{5,8,3,4,6,9\}$ D

E on b $\quad \delta(E, b) = 7 \quad \varepsilon\text{-close}(7) = \{7,8,9,3,4,6\}$ E

δ	a	b
→ A	B	ϕ
B	ϕ	C
C*	D	E
D*	D	E
E*	D	E



(b) ε-NFA to NFA

δ	a	b
→ 0	1	ϕ
1	ϕ	{2,3,4,6,9}
2*	{5,8,3,4,6,9}	{7,8,3,4,6,9}
3	{5,8,3,4,6,9}	{7,8,3,4,6,9}
4	{5,8,3,4,6,9}	ϕ
5*	{5,8,3,4,6,9}	{7,8,3,4,6,9}
6	ϕ	{7,8,3,4,6,9}
7*	{5,8,3,4,6,9}	{7,8,3,4,6,9}
8*	{5,8,3,4,6,9}	{7,8,3,4,6,9}
9*	ϕ	ϕ

$\delta(\text{e-closure}(0), a) = \text{eclosure}(1)$

$= 1$

$\delta(\text{e-closure}(0), b) = \text{eclosure}(\phi) = \phi$

$\delta(\text{e-closure}(1), a) = \text{eclosure}(\phi) = \phi$

$\delta(\text{e-closure}(1), b) = \text{eclosure}(2) = \{2,3,4,6,9\}$

$\delta(\text{e-closure}(2), a) = \text{eclosure}(5) = \{5,8,3,4,6,9\}$

$\delta(\text{e-closure}(2), b) = \text{eclosure}(7) = \{7,8,3,4,6,9\}$

$\delta(\text{e-closure}(3), a) = \text{e-closure}(5) = \{5,8,3,4,6,9\}$

$\delta(\text{e-closure}(3), b) = \text{e-closure}(7) = \{7,8,3,4,6,9\}$

$\delta(\text{e-closure}(4), a) = \text{e-closure}(5) = \{5,8,3,4,6,9\}$

$\delta(\text{e-closure}(4), b) = \text{e-closure}(\phi) = \phi$

$\delta(\text{e-closure}(5), a) = \text{e-closure}(5) =$

$\delta(\text{e-closure}(5), b) = \text{e-closure}(7) =$

$\delta(\text{e-closure}(6), a) = \text{eclosure}(\phi) = \phi$

$\delta(\text{e-closure}(6), b)) = \text{eclosure}(7) =$

$\delta(\text{e-closure}(7), a) = \text{eclosure}(5) =$

$\delta(\text{e-closure}(7), b) = \text{eclosure}(7) =$

$\delta(\text{e-closure}(8), a) = \text{eclosure}(5) =$

$\delta(\text{e-closure}(8), b) = \text{eclosure}(7) =$

$\delta(\text{e-closure}(9), a) = \text{eclosure}(\phi) =$

$\delta(\text{e-closure}(9), b) = \text{eclosure}(\phi) =$

Equivalence & Minimization of Automata:

Equivalence of 2 States:

①. Equivalent (Indistinguishable):

$$\delta(p, w) \in F \quad \& \quad \delta(q, w) \in F$$

or

$$\delta(p, w) \notin F \quad \& \quad \delta(q, w) \notin F$$

②. Distinguishable:

$$\delta(p, w) \in F \quad \& \quad \delta(q, w) \notin F.$$

Scanned by CamScanner

Partition Method for minimization of states of DFA

Step 1: Remove all unreachable states from the start state.

Step 2: Construct Πold : Partition the states into 2 groups one consisting of final states and other consisting of non-final states.

Step 3: Construct Πnew from Πold.

[Find $\delta(q, a)$ for each q in G on i/p symbol]

If transition results with the states in different groups proceed with partition

else no need for partition.

Step 4: If Πnew != Πold

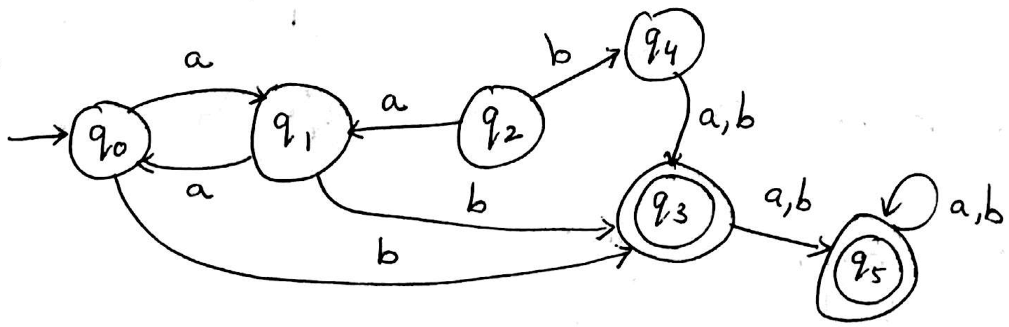
Copy Πnew to Πold.

Goto step 3.

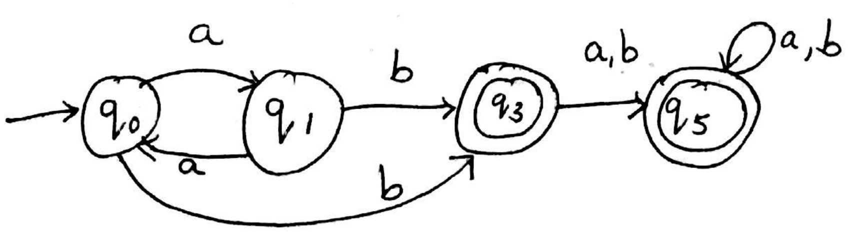
Step 5:

From the Πnew draw the transition table & DFA.

(1).



Remove unreachable states.



Step 0:

non-final $[q_0 \quad q_1]$ final $[q_3 \quad q_5]$

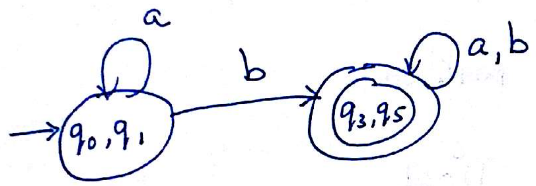
Step 1:

$[q_0 \quad q_1]$ $[q_3 \quad q_5]$

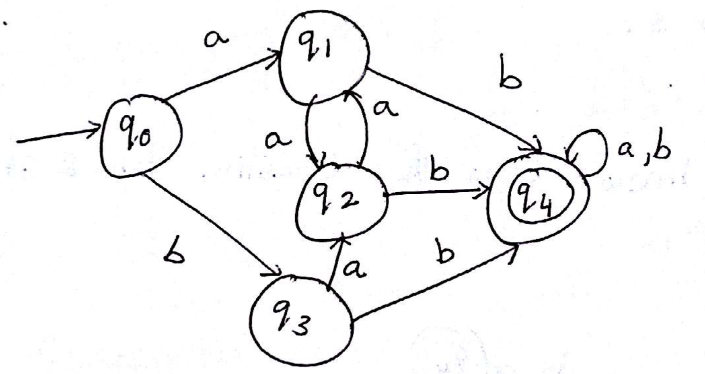
$\therefore$ $q_0 \equiv q_1$

$q_3 \equiv q_5$

δ	a	b
q_0, q_1	q_0, q_1	q_3
q_3, q_5	q_5	q_5



②



δ	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_4
q_2	q_1	q_4
q_3	q_2	q_4
q_4^*	q_4	q_4

No unreachable state.

Step 0:

non-final $[q_0 \quad q_1 \quad q_2 \quad q_3]$ final $[q_4]$

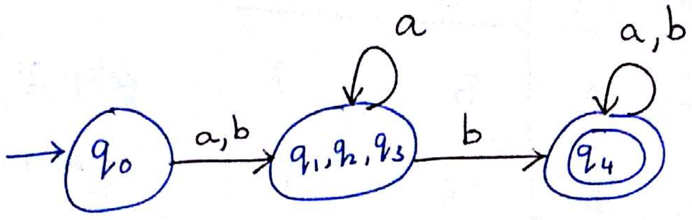
Step 1:

non-fi $[q_0]$ non-final final $[q_1 \quad q_2 \quad q_3]$ final $[q_4]$

Step 2:

$[q_0]$ $[q_1 \quad q_2 \quad q_3]$ $[q_4]$

δ	a	b
→ q_0	q_1	q_3
q_1,q_2,q_3	q_1,q_2	q_4
q_4^*	q_4	q_4



③.

δ	0	1
→ A	B	F
B	G	C^*
C^*	A	C^*
D	C	G
E	H	F
F	C^*	G
G	G	E
H	G	C^*

D is not reachable state & so it is removed.

Step 0:

non-final [A B E F G H] final [C]

Step 1:

[A E G] [B H] [F] [C]

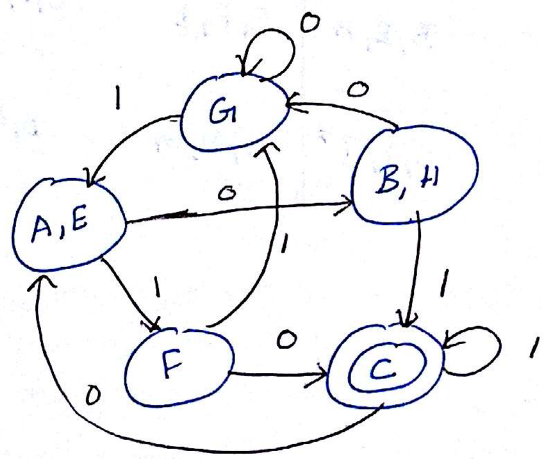
Step 2:

[A E] [G] [B H] [F] [C]

Step 3:

[A E] [G] [B H] [F] [C]

δ	0	1
A, E	B, H	F
G	G	E
B, H	G	C
F	C	G
C^*	A	C



(4).

δ	0	1
→ A	B	E
B	C*	F*
C*	D	H
D	E	H
E	F*	I*
F*	G	B
G	H	B
H	I*	C*
I*	A	E

There is no unreachable state

Step 0:

non-final final

[A B D E G H] [C F I]

Step 1:

[A D G] [B E H] [C F I]

Step 2:

[A D G] [B E H] [C F I]

$A \equiv D \equiv G$

$B \equiv E \equiv H$

$C \equiv F \equiv I$

δ	0	1
A, D, G	B, E, H	B, E, H
B, E, H	C, F, I	C, F, I
C, F, I*	A, D, G	B, E, H

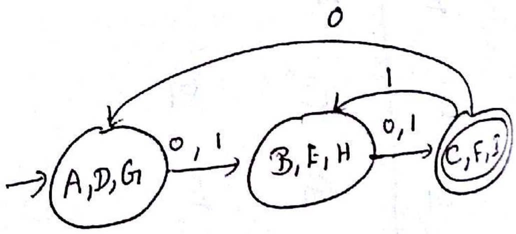


TABLE FILLING METHOD:

1.

δ	0	1
→ A	B	F
B	G	C
C*	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

	A	B	C	D	E	F	G
B	X						
C	X	X					
D	X	X	X				
E		X	X	X			
F	X	X	X	X	X		
G	X	X	X	X	X	X	
H	X	X	X	X	X	X	X

δ (p,q)	0 (r,s)	1 (r,s)
A,B	B,G	$\boxed{C,F}$
A,D	$\boxed{B,C}$	F,G
A,E	B,H	F,F
A,F	$\boxed{B,C}$	F,G
A,G	B,G	E,F
A,H	B,G	$\boxed{C,F}$
B,D	$\boxed{C,G}$	$\boxed{C,G}$
B,E	G,H	$\boxed{C,F}$
B,F	$\boxed{C,G}$	$\boxed{C,G}$
B,G	G,G	$\boxed{C,E}$
B,H	G,G	$\boxed{C,C}$
D,E	$\boxed{C,H}$	G,F
D,F	$\boxed{C,C}$	G,G
D,G	$\boxed{C,G}$	E,G
D,H	$\boxed{C,G}$	C,G
E,F	$\boxed{C,H}$	F,G
E,G	G,H	E,F
E,H	H,G	$\boxed{C,F}$
F,G	$\boxed{C,G}$	E,G
F,H	$\boxed{C,G}$	$\boxed{C,G}$
G,H	G,G	$\boxed{C,E}$

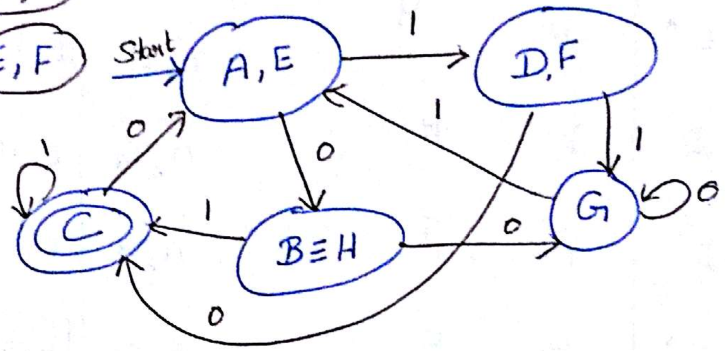
Scanned by CamScanner

δ	0	1
✓ A, E	B, H	F, F
✓ D, F	C, C	G, G
✓ B, H	G, G	C, C
E, G	G, H	E, F
A, G	B, G	E, F

$A \equiv E$

$D \equiv F$

$B \equiv H$



②.

δ	a	b
→ q_0	q_1	q_3
q_1	q_2	q_4
q_2	q_1	q_4
q_3	q_2	q_4
q_4^*	q_4	q_4

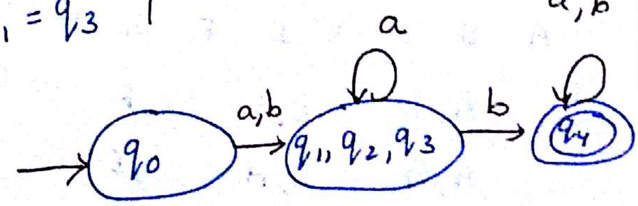
δ (p,q)	a (r,s)	b (r,s)
q_0, q_1	q_1, q_2	$\boxed{q_3, q_4}$
q_0, q_2	q_1, q_1	$\boxed{q_3, q_4}$
q_0, q_3	q_1, q_2	$\boxed{q_3, q_4}$
q_1, q_2	q_1, q_2	q_4, q_4
q_2, q_3	q_1, q_2	q_4, q_4
q_4^*	q_4	q_4

	q_0	q_1	q_2	q_3
q_1	X			
q_2	X			
q_3	X			
q_4	X	X	X	X

$q_1 = q_2$

$q_2 = q_3 \implies q_1 = q_2 = q_3$

$q_1 = q_3$



③.

δ	0	1
→ A	B	E
B	C	F
C*	D	H
D	E	H
E	F	I
F*	G	B
G	H	B
H	I	C
I*	A	E

δ	0	1
A, B	$\boxed{B, C}$	$\boxed{E, F}$
A, D	B, E	E, H
A, E	$\boxed{B, F}$	$\boxed{E, I}$
A, G	B, H	B, E
A, H	$\boxed{B, I}$	$\boxed{C, E}$
B, D	$\boxed{C, E}$	$\boxed{F, H}$
B, E	C, F	F, I
B, G	$\boxed{C, H}$	$\boxed{B, F}$
B, H	C, I	C, F
D, E	$\boxed{E, F}$	$\boxed{H, I}$
D, G	E, H	B, H
D, H	$\boxed{E, I}$	$\boxed{C, H}$
E, G	$\boxed{F, H}$	$\boxed{B, I}$
E, H	F, I	C, I
G, H	$\boxed{H, I}$	$\boxed{B, C}$
C, F	D, G	B, H
C, I	A, D	E, H
F, I	A, G	B, E

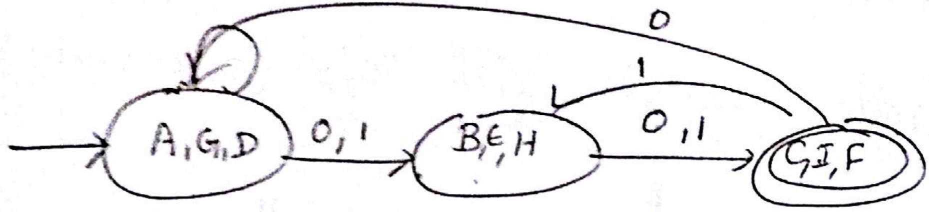
	A	B	C	D	E	F	G	H
B	X							
C	X	X						
D		X	X					
E	X		X	X				
F	X	X		X	X			
G		X	X		X	X		
H	X		X	X		X	X	
I	X	X		X	X		X	X

A, D   B, E   C, F

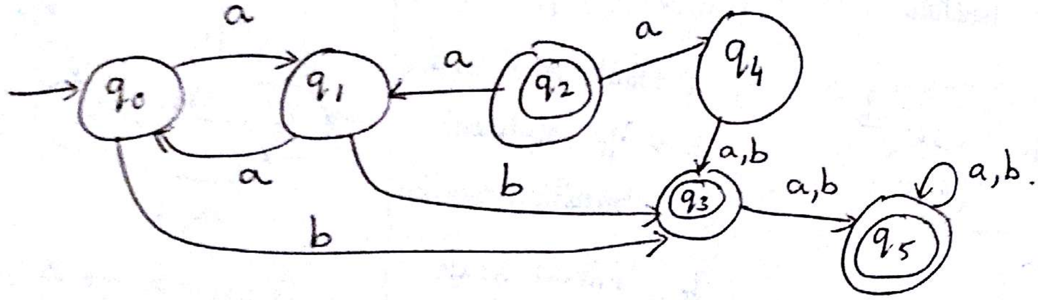
A, G   B, H   C, I

D, G   E, H   F, I

⇒ A ≡ G ≡ D   B ≡ E ≡ H   C ≡ I ≡ F

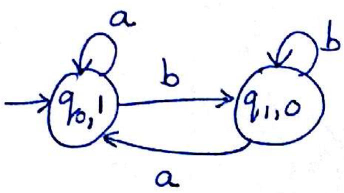


4



Finite Automata with o/p (Deterministic)

Moore machine



$\lambda : Q \to \Delta$

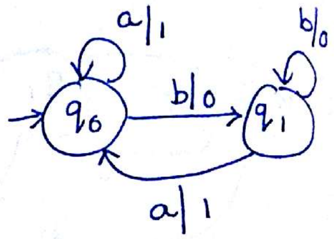
$q_0 \to 1$

$q_1 \to 0$

$(Q, \Sigma, \delta, q_0, \Delta, \lambda)$

- $Q \to$ finite set of states
- $\Sigma \to$ i/p alphabet
- $\delta \to$ transition function
- q_0 initial state
- Δ o/p alphabet
- λ : o/p function

Mealy machine



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

$(q_0, a) \to 1$

$(q_0, b) \to 0$

$(q_1, b) \to 0$

$(q_1, a) \to 1$

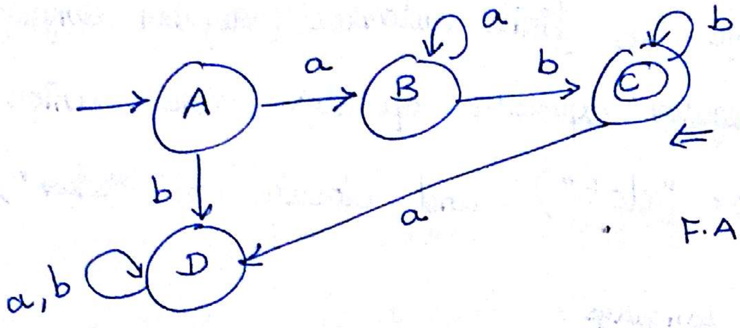
Families of Languages

Finite Automata

- with o/p
  - Moore
  - Mealy
- without o/p
  - DFA
  - NFA
  - ε-NFA

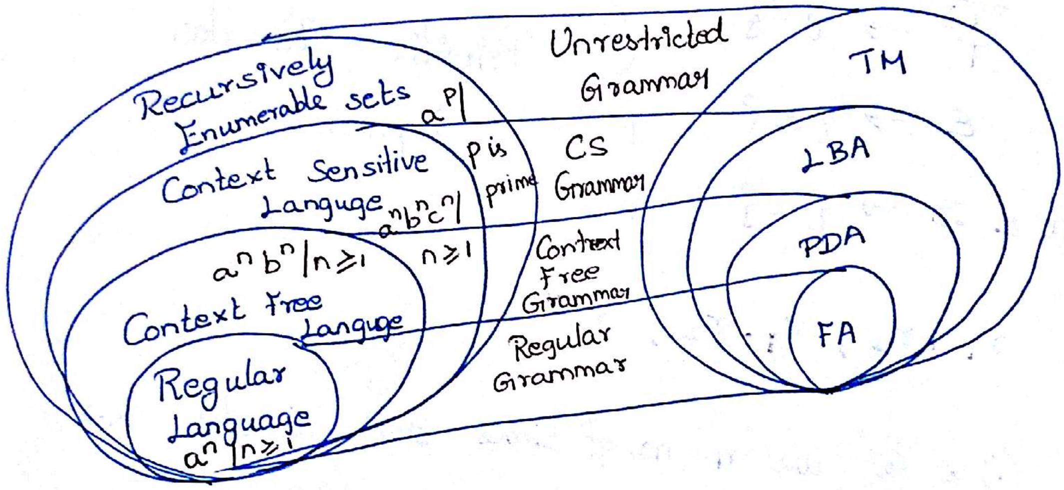
Ex: $L = \{a^n b^n \mid n \geq 1\}$

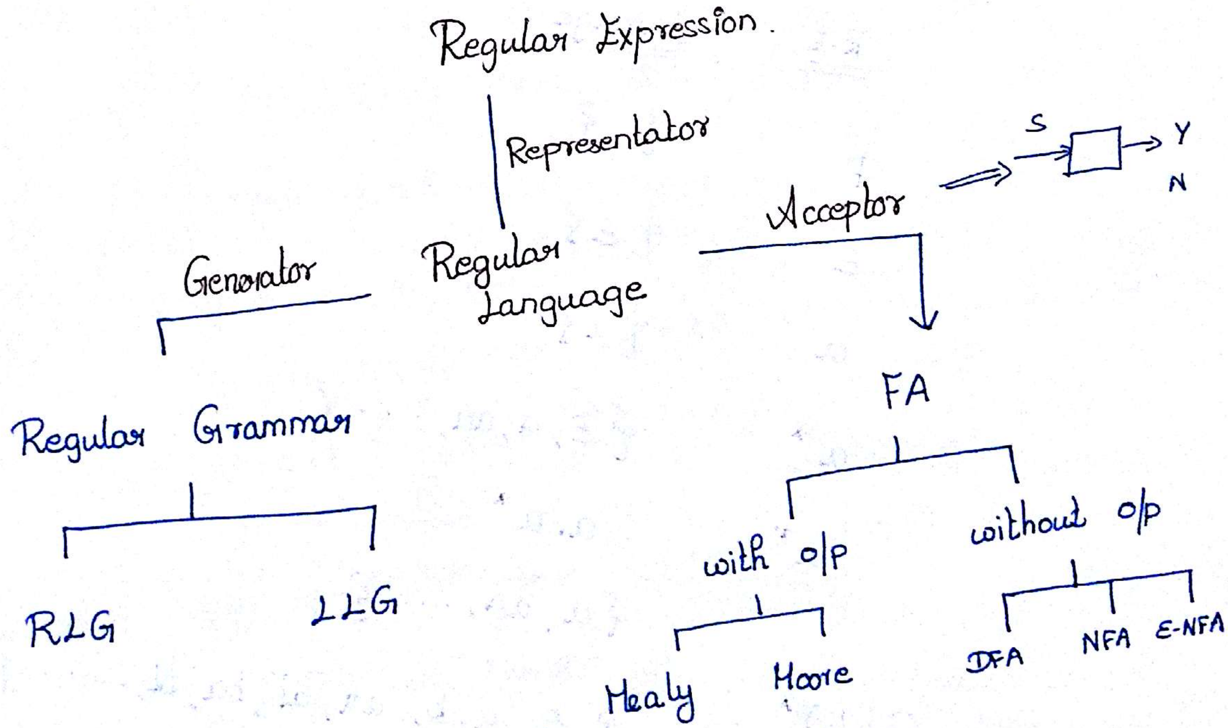
$L = \{ab, aabb, aaabbb, \ldots\}$



$\Leftarrow$ Not a valid Automata

F.A + $\underbrace{\text{memory}}_{\text{Stack}}$ = PDA





Regular Expression

It is the algebraic notation that describes exactly the same language as finite automata (regular languages).

The regular expression operators are union (+), contatenation (or "dot" •) and closure (or "star").

Note:

	R.E		Language
①.	ϕ	$\rightarrow$	$\{\ \}$
	ε	$\rightarrow$	$\{\varepsilon\}$
	$a \in \Sigma$	$\rightarrow$	$\{a\}$

} Primitive Regular Expressions

②. $r_1 + r_2$, $r_1 . r_2$, r_1^*

③. ① & ② use n no. of times that is also R.E

R.E	Langluge
ϕ	$\{\ \}$
ε	$\{\varepsilon\}$
a	$\{a\}$
a^*	$\{\varepsilon, a, aa, \dots\}$
a^+	$a.a^*$ $\{a, aa, \dots\}$
$(a+b)^*$	$\{\varepsilon, a, b, aa, ab, ba, bb, \dots\}$

Scanned by CamScanner

exactly
ages).
n, (+)
").

R.E

b, ... }

<u>Examples:</u> $\Sigma = \{a,b\}$

①. $L_1 = \{$strings having length exactly $2\}$

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

$= aa + ab + ba + bb$

$= a(a+b) + b(a+b)$

$= (a+b).(a+b)$ => length exactly 2

$(a+b)(a+b)(a+b)$ => length exactly 3

$(a+b)(a+b)(a+b)(a+b)$ => length exactly 4

②. $L_2 = \{$strings having length atleast $2\}$

$= \{aa, ab, ba, bb, aaa, \ldots\}$

length: 2, 3, 4, ...

$\Downarrow$

$(a+b)(a+b)(a+b)^*$ → 0 or more times

③. $L_3 = \{$strings having length atmost $2\}$

length = 0, 1, 2.

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

$= \varepsilon + a + b + aa + ab + ba + bb$

$= (\varepsilon + a + b)(\varepsilon + a + b)$

④. $L_4 = \{$even length strings$\}$

$L = \{\varepsilon, aa, ab, ba, bb, \ldots\}$

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

(5). $L_5 = \{$ strings having odd length $\}$

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

(6). $L_6 = \{$ strings length divisible by '3' $\}$

$L = 0, 3, 6, 9, 12, \ldots$

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

(7). $L_7 = \{$ sets of strings whose length is $\cong 2 \mod 3 \}$

$\Downarrow$

$$(a+b)^{3n+2} / n \geq 0$$

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

(8). $L_8 = \{$ no. of a's exactly 2 $\}$ $\Sigma = \{a, b\}$

$$b^* a b^* a b^*$$

(9). $L_9 = \{$ a's are atleast '2' $\}$

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

(10). $L_{10} = \{$ a's atmost '2' $\}$

$$b^* (\varepsilon + a) b^* (\varepsilon + a) b^*$$

(11). $L_{11} = \{$ set of a's are even $\}$ $L_{11} = \{$ aa, aaaa, ... b, bb, bbbaa $\}$

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

$$\cong (b^* a b^* a)^* \cdot b^*$$

12. L_{12} = { set of strings starts with a }

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

13. L_{13} = { set of strings ends with a }

$$(a+b)^* \cdot a$$

14. L_{14} = { set of all strings containing a }

$$(a+b)^* \cdot a \cdot (a+b)^*$$

15. L_{15} = { starting and ending with diff. symbols }

$$[a(a+b)^* b] + [b(a+b)^* a]$$

16. L_{16} = { starting and ending with same symbol }

$L = \{\epsilon, a, b, aa, bb \ldots\}$

$$[a(a+b)^* a] + [b(a+b)^* b] + [a+b+\epsilon]$$

17. L_{17} = { 2 a's should not come together }

(X)

$L = \{\epsilon, \underline{b}, bb, bbb, a, \underline{a}ba, ab, abab, ababab\}$
ba, bab, ba, baba, ...

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

$$= (b+ab)^* \cdot (\epsilon + a)$$

(or)

$$= (a+\epsilon) \cdot (b+ba)^*$$

(18). L_{18} : {no 2 a's & 2 b's come together}

= {ε, a, b, ab, ba, aba, bab, ...}

	Starts with	Ends with
{a, aba, ababa, ...}	a	a ⇒ $(ab)^*a$ or $a(ba)^*$
{ab, abab, ababab, ...}	a	b ⇒ $(ab)^*$ or $a(ba)^*b$
{ba, baba, bababa, ...}	b	a ⇒ $(ba)^*$ or $b(ab)^*a$
{b, bab, babab, bababab, ...}	b	b ⇒ $(ba)^*b$ or $b(ab)^*$

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

$$= (ab)^*(a+\varepsilon) + [b(ab)^*](a+\varepsilon)$$

$$= (\varepsilon+b)(ab)^*(a+\varepsilon)$$

(or)

$$(\varepsilon+a)(ba)^*(b+\varepsilon)$$

Scanned by CamScanner

Identities of Regular Expressions:

Empty set $\rightarrow$ RE (or) RL

⊛. $\phi + R = R + \phi = R$

⊛. $\phi \cdot R = R \cdot \phi = \phi$

⊛. $\varepsilon \cdot R = R \cdot \varepsilon = R$

⊛ $\varepsilon^* = \varepsilon$

⊛ $\phi^* = \varepsilon$

⊛ $\varepsilon + R \cdot R^* = R^* R + \varepsilon = R^*$

⊛. $(a+b)^* = (a^* + b^*)^*$

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

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

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

$= a^*(ba^*)^*$

$= b^*(ab^*)^*$

Kleen's Theorem

Part ①: Any regular language can be accepted by a FA.

②: The language accepted by any finite automaton is regular.

(*). $L + M = M + L$

(*). $(L+M)+N = L+(M+N)$

(*). $(LM).N = L(MN)$

(*). $L(M+N) = LM+LN$

(*). $(M+N)L = M.L+NL$.

(*) $L+L = L$.

(*). $(L^*)^* = L^*$

(*) $L^+ = LL^* = L^*L$

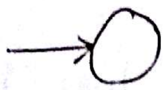
(*). $L^* = L^+ + \epsilon$

(*). $(L+M)^* = (L^*M^*)^*$

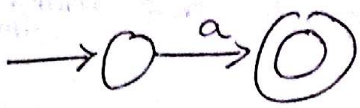
Scanned by CamScanner

Regular Expression to FA

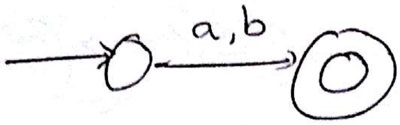
1. ϕ



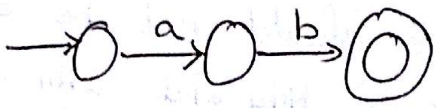
2. a



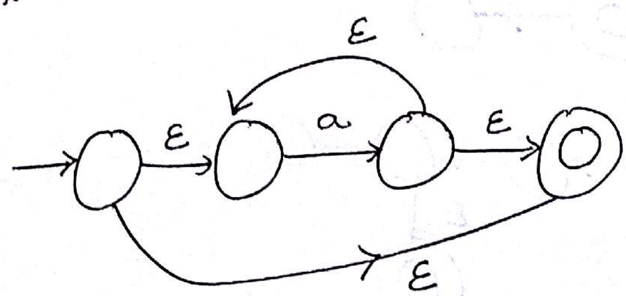
3. a+b



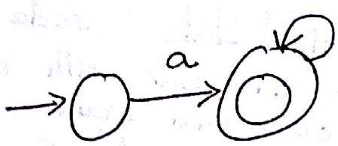
4. a.b



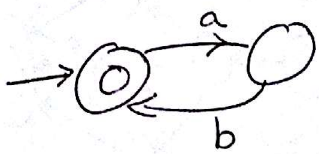
5. a^*



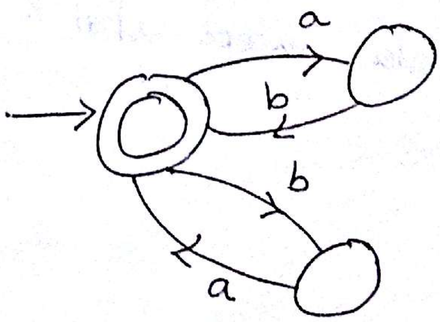
6. ab^*



7. $(ab)^*$

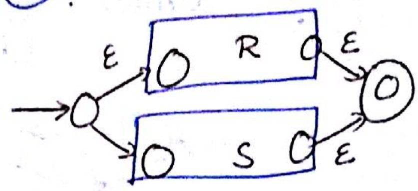


8. $(ab+ba)^*$

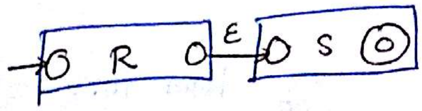


RE → ε-NFA

9. R+S



10. R.S



11. R^*

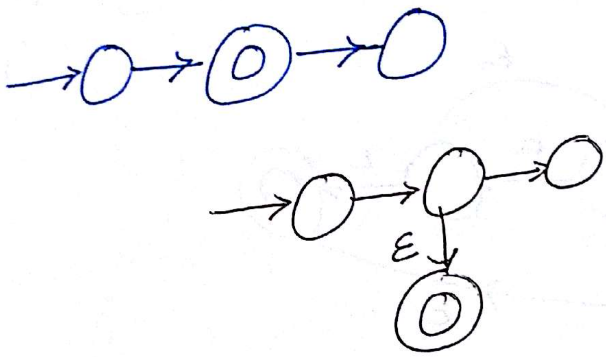
llar to 5.

Finite Automata to Regular Expression

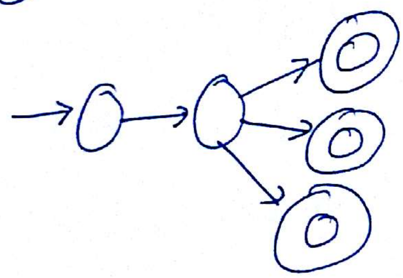
State Elimination Method:

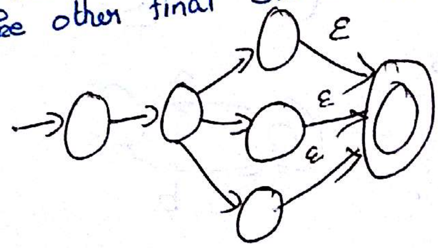
① Initial state should not have incoming edge (if so create new initial state with ε moves)

②ⓐ From final state there should not be any outgoing edge (if so create final state with ε. moves) & make old final state as non final



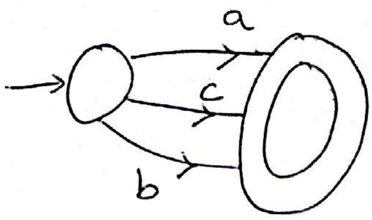
②ⓑ If there is more than one final state create only one new final state with ε-moves & make other final states non-final.

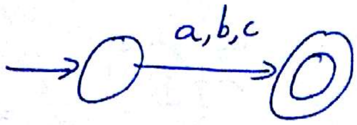




③. Eliminate all the other states except start & final states.

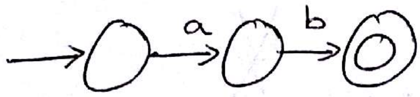
*1





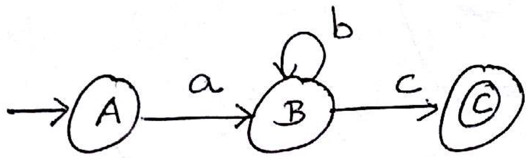
$a+b+c$

*2



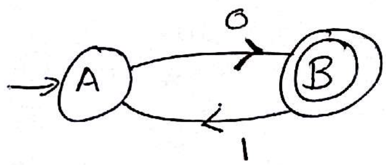
ab.

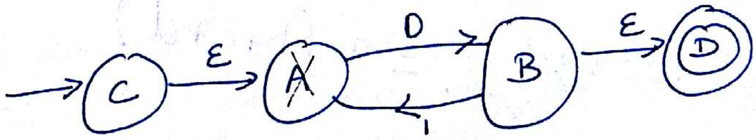
*3

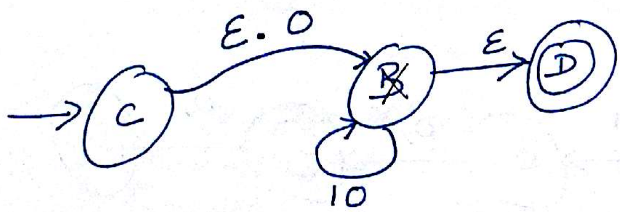


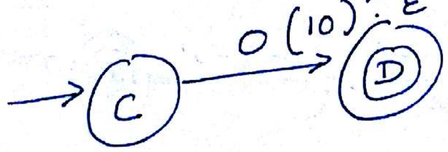
ab^*c

*4



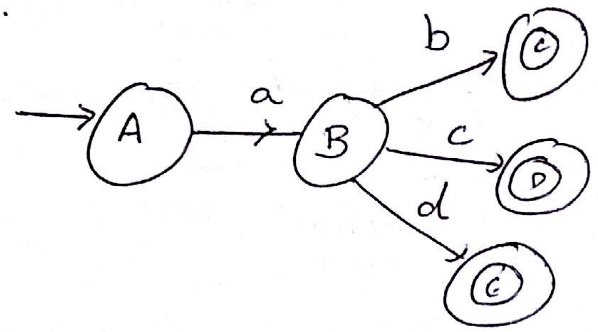


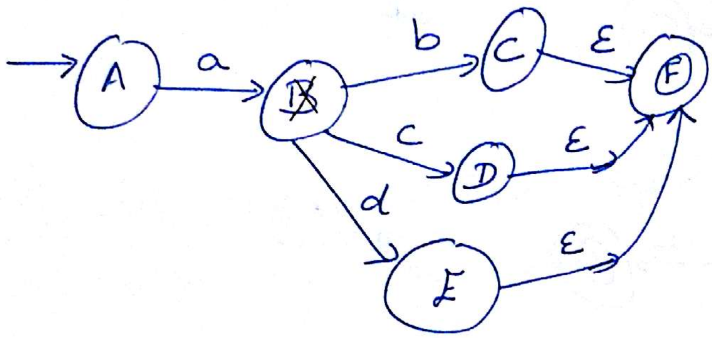


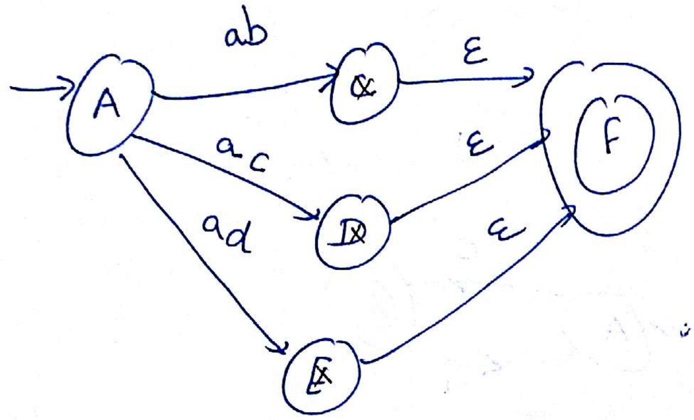


$\equiv 0(10)^*$

*5.

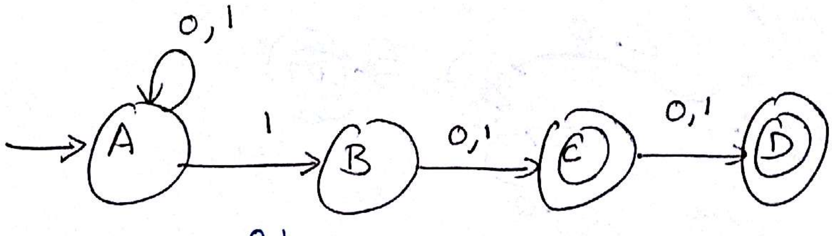


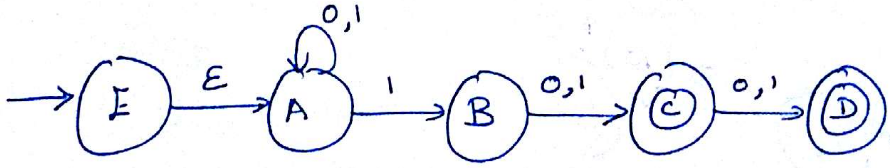


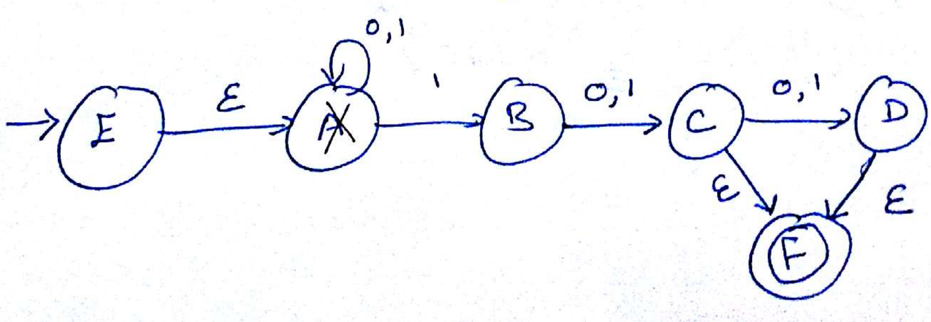


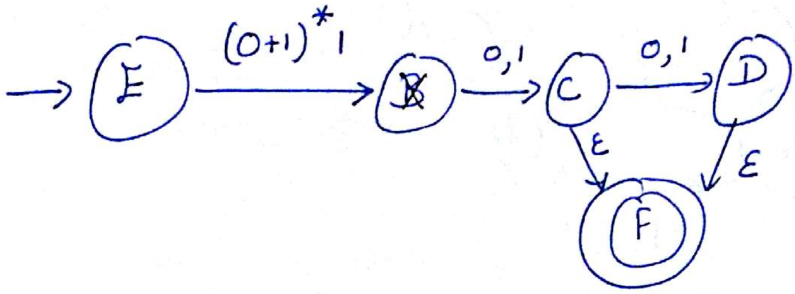
$$(ab + ac + ad) = a\,(b + c + d)$$

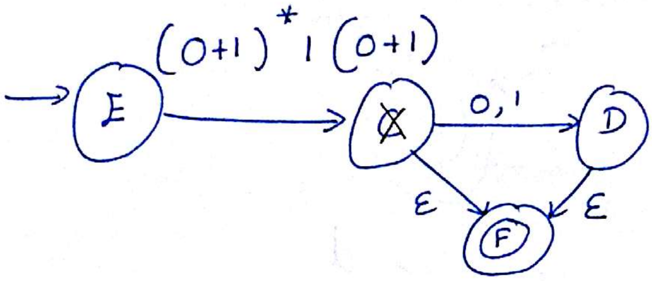
*6





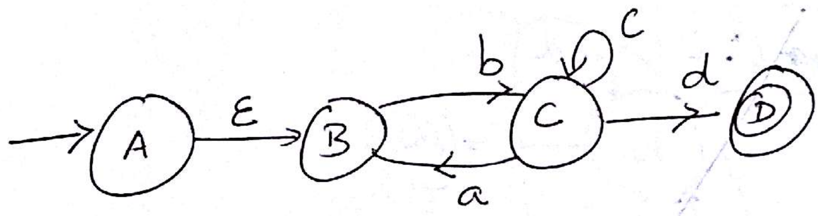


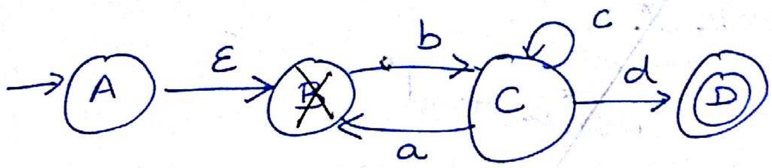


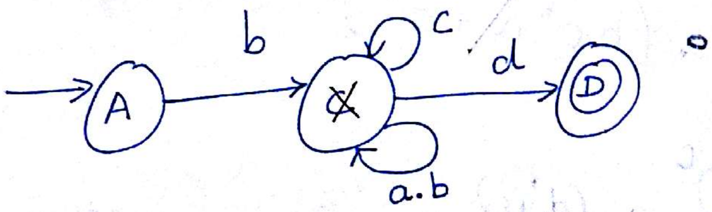


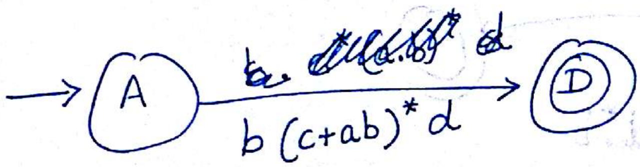
$(0+1)^* 1 (0+1) \quad + \quad (0+1)^* 1 (0+1)(0+1)$

Q7.



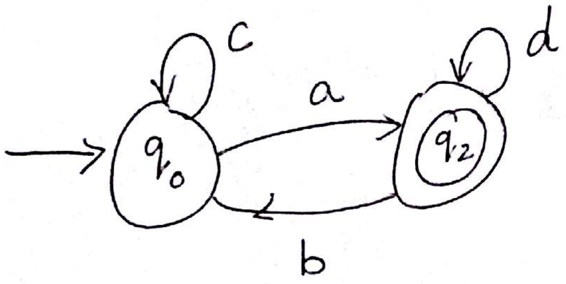


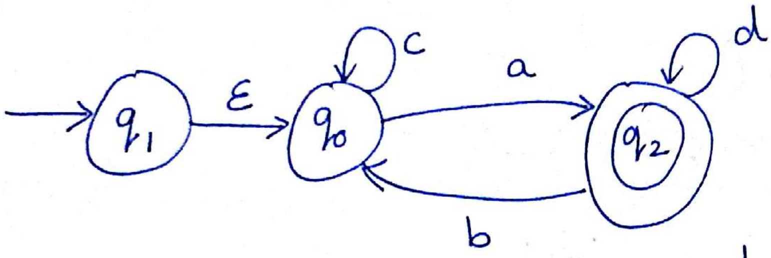


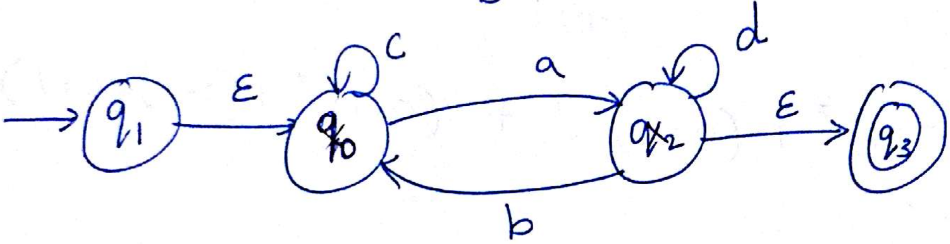


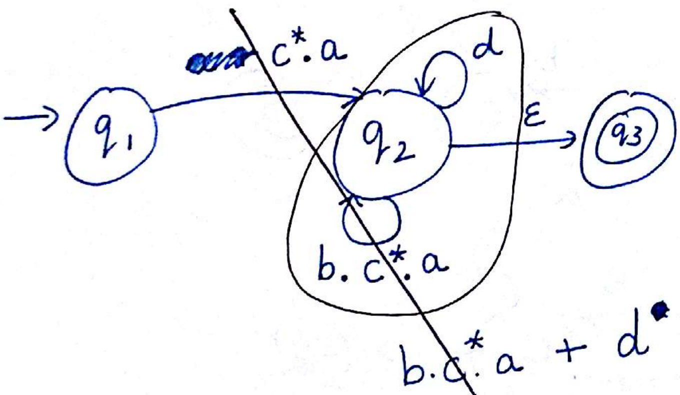
$b(c+ab)^* d.$

*8.

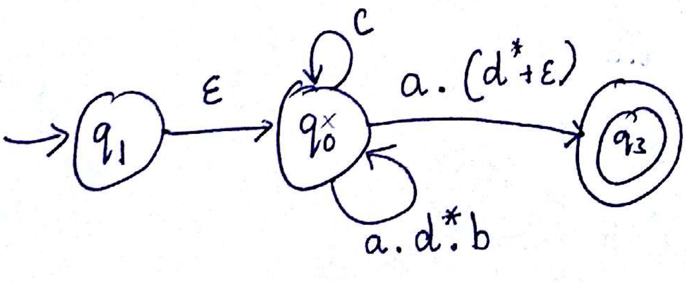


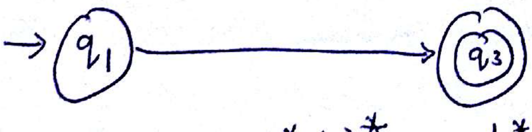






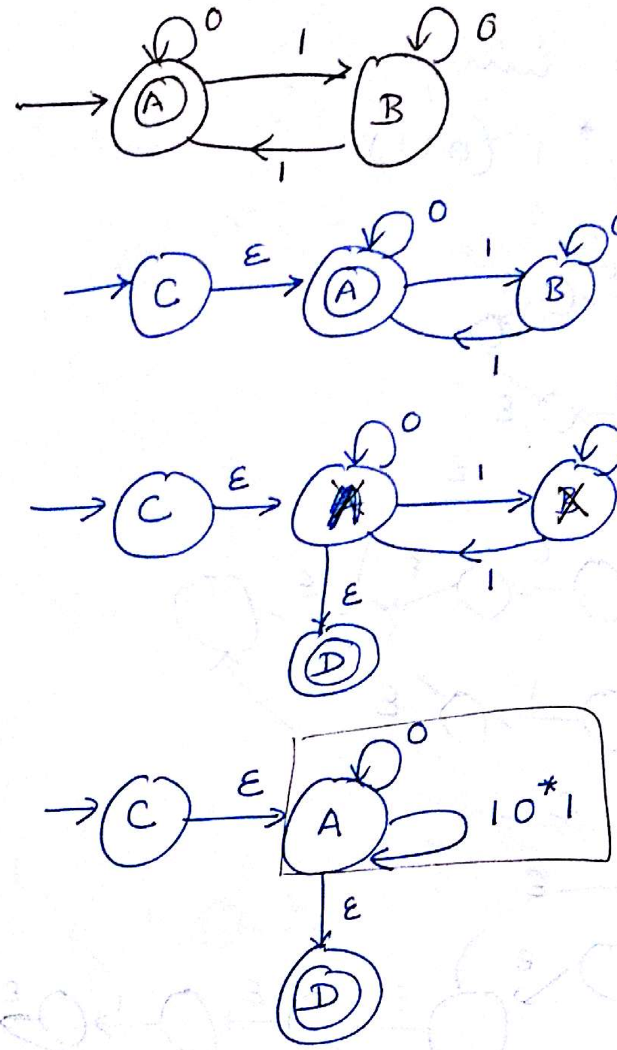
$[c^*.a] . (b.c^*.a + d^*)^*$





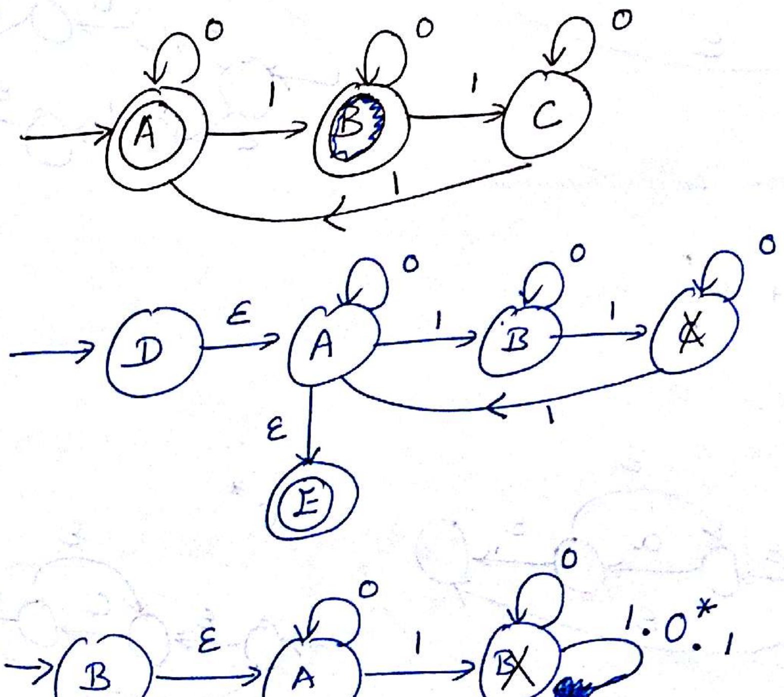
$(c + a.d^*.b)^*.a.d^*$

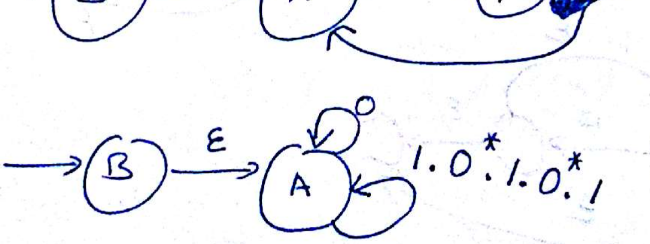
*9



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

*10





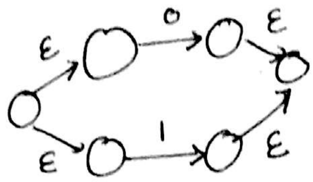
$(0 + 10^*10^*1)^*$

Scanned by CamScanner

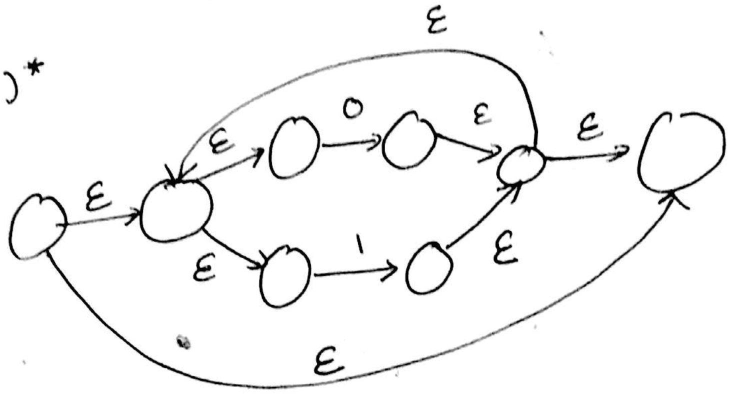
Convert the R.E to ε-NFA:

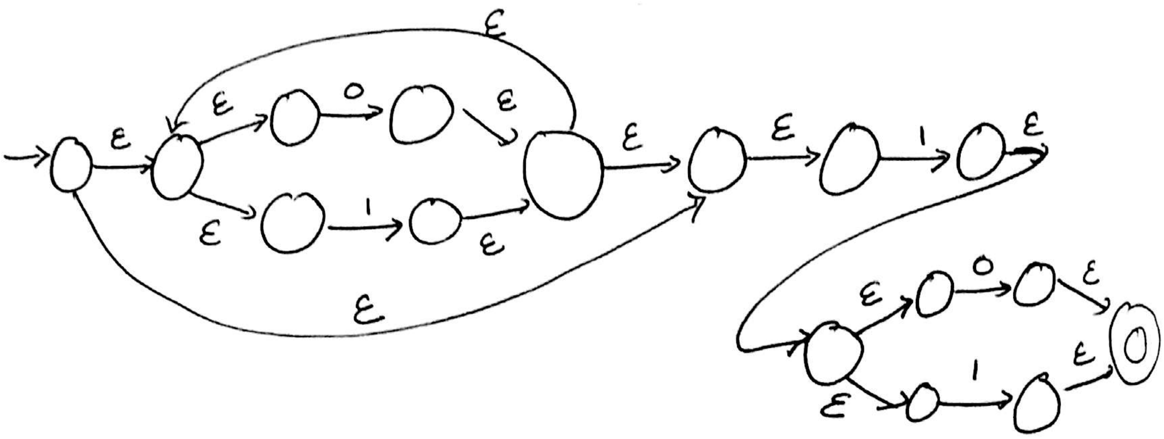
① $(0+1)^* 1 (0+1)$

0+1



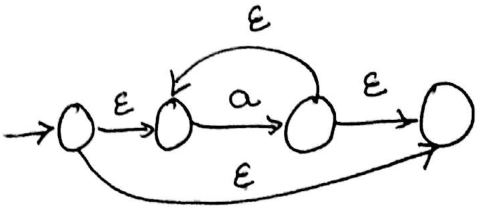
$(0+1)^*$



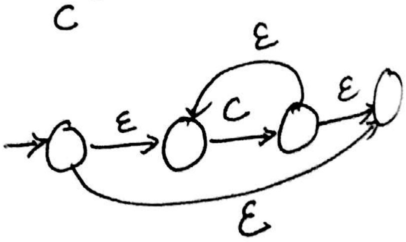


② $a^* + b^* + c^*$

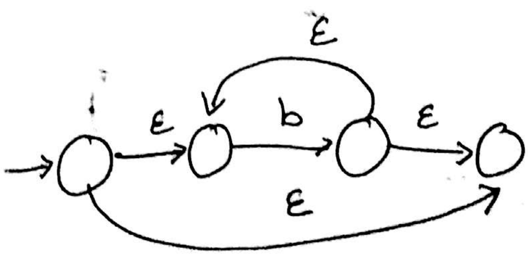
a^*

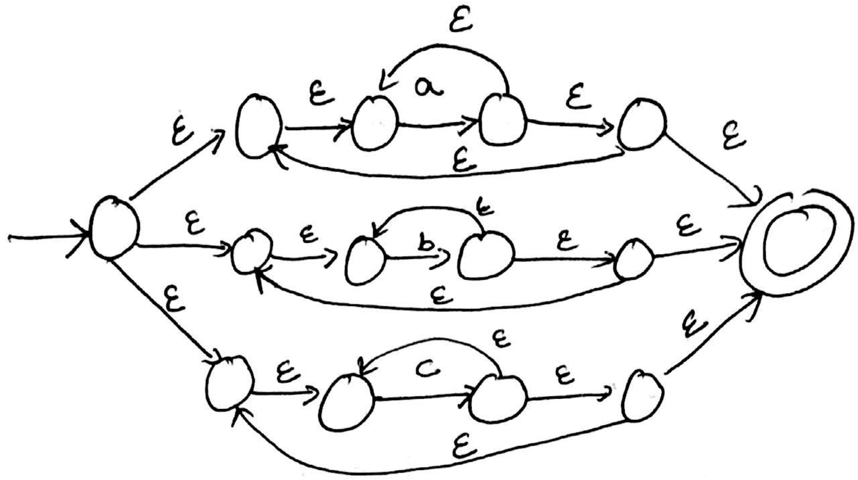


c^*



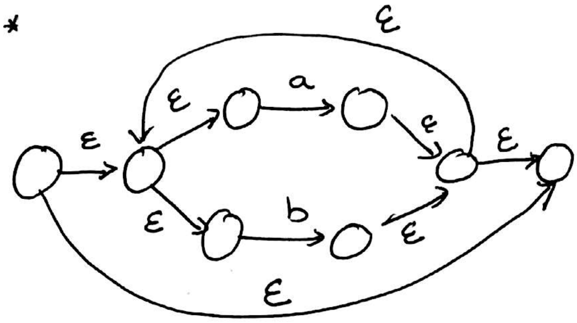
b^*

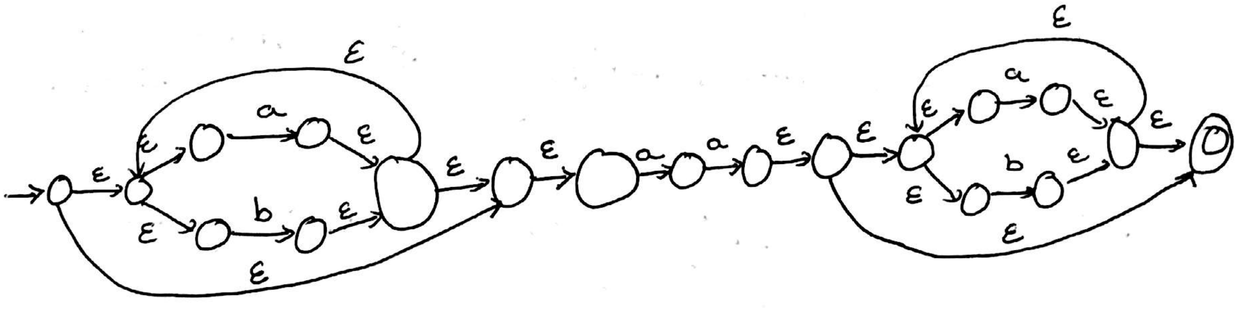




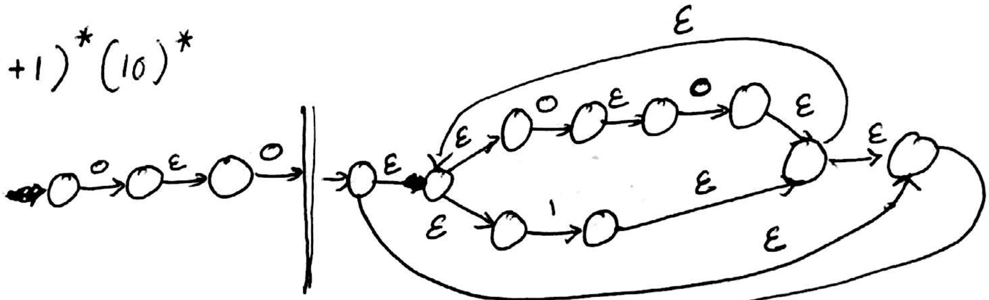
③. $(a+b)^* aa (a+b)^*$

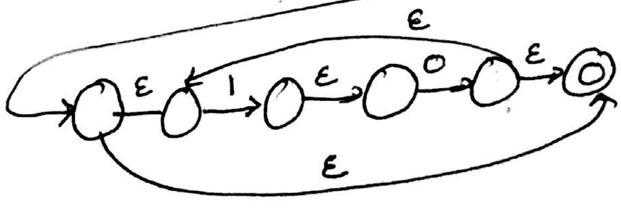
$(a+b)^*$





④. $(00+1)^*(10)^*$





Kleene's Theorem Part 2

The language accepted by any Finite Automata is regular. $\boxed{FA \to R \cdot E}$

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

$Q = \{1, 2, 3, \ldots, n\}$

$R_{ij}^{k} = \{w \in \Sigma^* \mid$ w is label from i to state j that goes through an intermidiate state whose no. is not greater than k $\}$.

$w = xy$

$\hat{\delta}(i, x) = k \quad \hat{\delta}(k, y) = j$.

$\boxed{k=0}$ $\quad i \to j$

① $i \neq j$

ⓐ. $R_{ij}^{0} = \phi$ (no i/p)

ⓑ. $R_{ij}^{0} = a$ (one i/p)

ⓒ. $R_{ij}^{0} = a_1 + a_2 + \cdots + a_m$ (multi i/p)

②. $i = j$

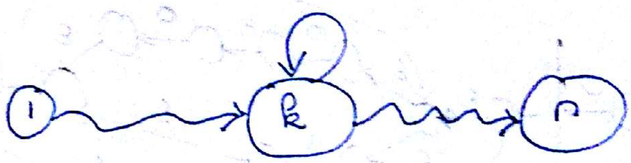
ⓐ. $R_{ij}^{0} = \phi + \varepsilon$ (no i/p)

$R_{ij}^{0} = a + \varepsilon$ (one i/p)

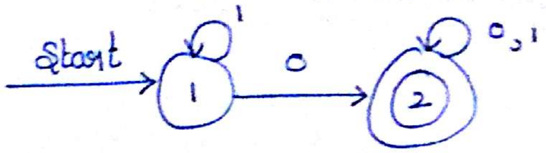
$R_{ij}^{0} = a_1 + a_2 + \cdots + a_m$ (multi i/p)

$\boxed{k \neq 0}$

$$R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$



1.



$k = 0, 1, 2.$

$\underline{k=0}$

$R_{11}^{0} = 1+\varepsilon$

$R_{12}^{0} = 0$

$R_{21}^{0} = \phi$

$R_{22}^{0} = 0+1+\varepsilon$

$\underline{k=1}$

$R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1}$

$R_{11}^{1} = R_{11}^{0} + R_{11}^{0} (R_{11}^{0})^* R_{11}^{0}$

$= (1+\varepsilon) + (1+\varepsilon)(1+\varepsilon)^* (1+\varepsilon)$

$= (1+\varepsilon)(1+\varepsilon)^* (1+\varepsilon)$

$= 1^*.$

$R_{12}^{1} = R_{12}^{0} + R_{11}^{0} (R_{11}^{0})^* R_{12}^{0}$

$= 0 + (1+\varepsilon)(1+\varepsilon)^*. 0$

$= 0 + 1^*.0 = 1^*.0$

$R_{21}^{1} = R_{21}^{0} + R_{21}^{0} (R_{11}^{0})^* R_{11}^{0}$

$= \phi + \phi (1+\varepsilon)^* (1+\varepsilon)$

$= \phi$

$R_{22}^{1} = R_{22}^{0} + R_{21}^{0} (R_{11}^{0})^* R_{12}^{0}$

$= (0+1+\varepsilon) + \phi . (1+\varepsilon)^* . 0$

$= 0+1+\varepsilon$

$\underline{k=2}$

$R_{11}^{2} = R_{11}^{1} + R_{12}^{1} (R_{22}^{1})^* R_{21}^{1} =$

$= 1^* + 1^*.0 (0+1+\varepsilon). \phi = 1^*$

$R_{12}^{2} = R_{12}^{1} + R_{12}^{1} (R_{22}^{1})^* R_{22}^{1} = 1^*.0 + 1^*.0 (0+1+\varepsilon)^* (0+1+\varepsilon)$

$\boxed{R_{12}^{2} = 1^*.0 \; (0+1)^*}$

$R_{21}^{2} = R_{21}^{1} + R_{22}^{1} (R_{22}^{1})^* R_{21}^{1} = \phi + (0+1+\varepsilon). (0+1+\varepsilon)^*. \phi$

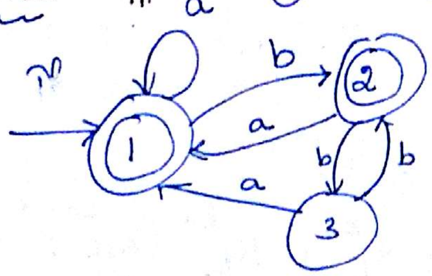
$= \phi$

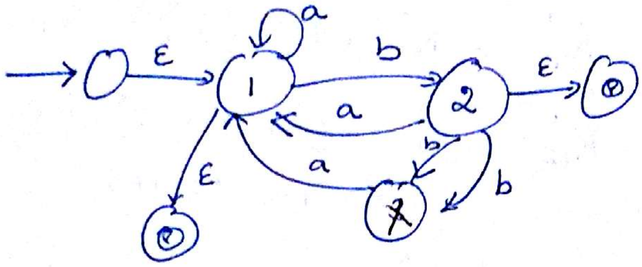
$R_{22}^{2} = R_{22}^{1} + R_{22}^{1} (R_{22}^{1})^* R_{22}^{1} = (0+1+\varepsilon) + (0+1+\varepsilon)(0+1+\varepsilon)^* (0+1+\varepsilon)^*$

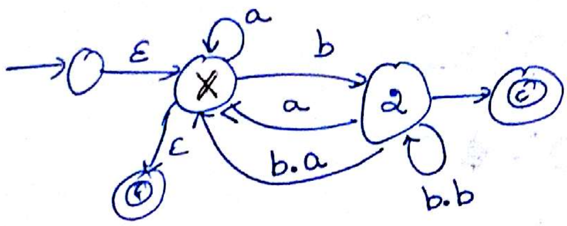
$= (0+1)^*$

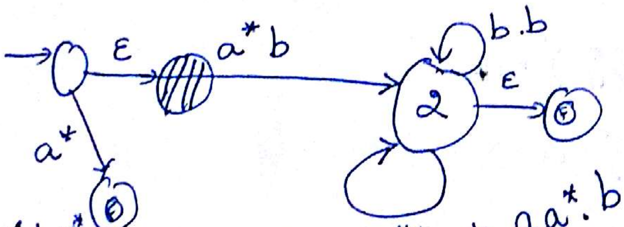
Kleene's Th (*11) State Elimination Method

ii







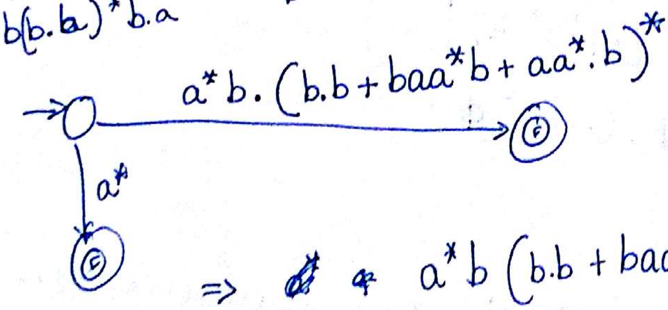


+

$a^*b(\epsilon+bb)^*a$

$a^*b(b.b)^*b.a$

$b.a\,a^*b + aa^*.b$



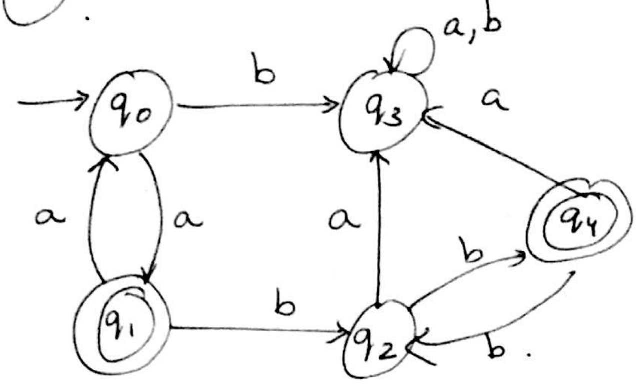
$\Rightarrow a^*b\,(b.b + baa^*b + a.a^*.b)^* +$

$\cancel{a^* + a^*ba + a^*b.b.a}$.

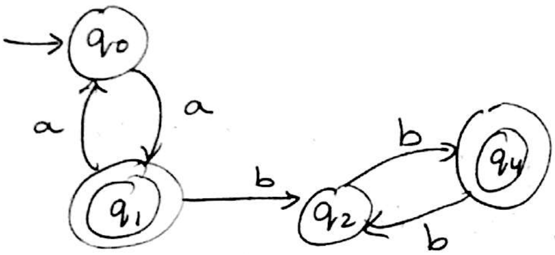
$a^* + a^*b\,(bb)^*a + a^*b\,(bb)^*b.a$

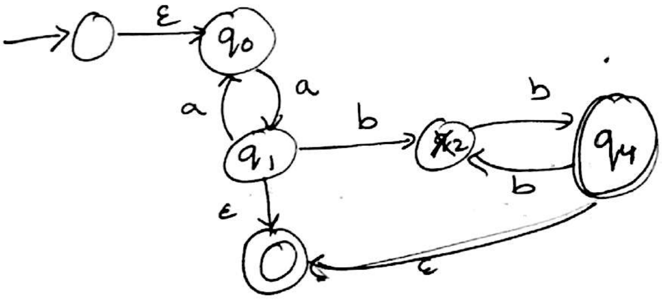
Scanned by CamScanner

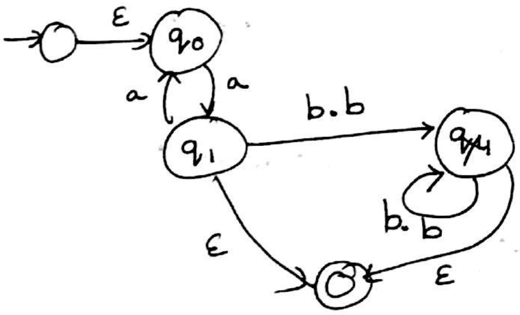
A12

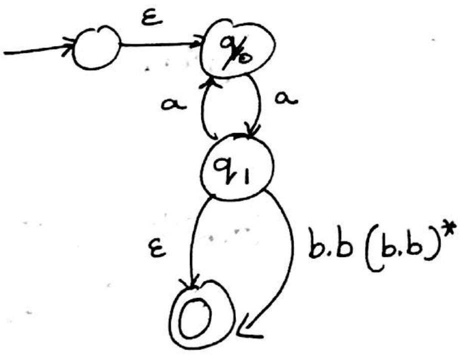


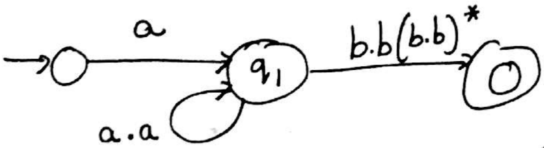
Remove non-reachable state q_3.











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

Pumping Lemma for Regular Languages:

Statement:

Let L be a regular language. There exists a constant n such that ~~any sentence~~ for every string w in L such that $|w| \geq n$, we can break w into three strings, $w = xyz$, such that:

$y \neq \epsilon \Rightarrow |y| \geq 1$

$|xy| \leq n$

For all $k \geq 0$, the string xy^kz is also in L.

(*) $y \Rightarrow$ if it is pumped any number of times or deleting it keeps the resulting string in the language L.

Proof:

Assume L is regular.

$\Rightarrow L = L(A)$ for some DFA A.

$|A| = n$

String w $|w| = n$ or more.

$w = a_1 a_2 \dots a_m$ $(m \geq n)$

Transition function of A $\quad i = 0, 1, \dots n$

$\hat{\delta}(q_0, a_1 a_2 \dots a_i)$

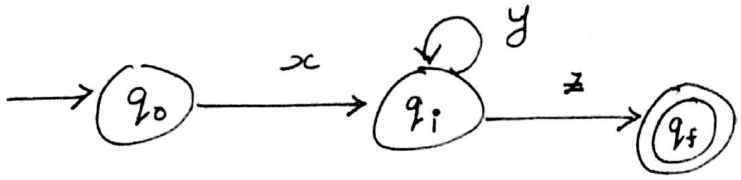
Break $w = xyz$ as follows:

$x = a_1 a_2 \dots a_i$

$y = a_{i+1} a_{i+2} \dots a_j$ $\quad 0 \leq i < j \leq n.$

$z = a_{j+1} a_{j+2} \dots a_m.$

[x may be empty; z may be empty. y cannot be empty, since i is strictly less than j]



[Every string longer than the number of states must cause a state to repeat]

(*)1 If $k=0$ then xz will be accepted.

(*2) If $k>0$ then xy^kz will be accepted.

Pumping lemma can be applied only to infinite languages.

Applications:

(*1) It is mainly used to show that certain languages are not regular.

(*) Points:

Language is Finite – Regular

Infinite < Regular ⇒ Some pattern ⇒ No FA is possible
Not Regular.

Pumping Lemma → Regular or not regular
→ Not Regular (Negativity Test)

1. Show that $L = \{a^n b^n \mid n \geq 0\}$ is not regular.

(i). Let L is regular and n be the number of states in FA.

String $w = a^n b^n$

(ii). $|w| = 2n \geq n$

$w = \cancel{}\ xyz$

such that $|xy| = n$ $|y| = 1$.

$$w = \underbrace{\overbrace{aaaa\cdots a}^{n}}_{x}\underbrace{a}_{y}\underbrace{\overbrace{bb\cdots b}^{n}}_{z}$$

According to pumping lemma.,.

$xy^k z \in L$ for $k = 0, 1, 2, \ldots$

(iii). $k = 0$.

y does not appear.

$\Rightarrow$ $n-1$ a's are followed by n b's $\Rightarrow$ is not in the language.

By contradiction to the assumption that the language is regular.

So, the language $L = \{a^n b^n \mid n \geq 0\}$ is not regular.

2. Show that $L = \{ww^R \mid w \in (0+1)^*\}$ is not regular.

(i). Let L is regular and n be the number of states in FA.

$$\text{String } w = \overbrace{\underbrace{1\cdots1}_{n}\underbrace{00}_{n}}^{w}\overbrace{\underbrace{00}_{n}\cdots\underbrace{11}_{n}}^{w^R}$$

tates

(ii). $|w| = 4.n \geq n$.

$w = xyz$

such that $|xy| = n \quad |y| = 1$

$$w = \underbrace{\overbrace{1 \cdots 1}^{x} \overbrace{1}^{y}}_{xy} \underbrace{0 \cdots 0 \; 0 \cdots 01 \cdots 0}_{z}$$

$x = 1 \cdots 1 \Rightarrow |x| = n-1$

$|y| = 1$

According to pumping lemma.,.

$xy^kz \in L$ for $k = 0, 1, 2, \ldots$

(iii). $k = 0$

y does not appear $\Rightarrow$ no. of 1's appear in LHS will be $<$ than RHS.

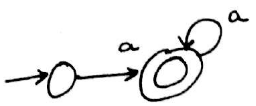
not in

So $xy^kz \notin L$ when $i = 0$, which is a contradiction to the assumption.

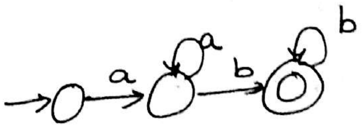
$\therefore L = \{ww^R \mid w \in \{0+1\}^*\}$ is not regular.

nguage.

the

(3). $L_1 = \{a^n \mid n \geq 1\}$  Regular

regular.

$L_2 = \{a^n b^m \mid n, m \geq 1\}$  Regular.

$L_3 = \{a^n b^n \mid n \leq 10^{10^{10^{10}}}\} \Rightarrow$ n is bounded so language is finite & so L_3 is Regular

ular.

$L_4 = \{ww^R \mid |w| = 2 \; \Sigma = \{a, b\}\} \Rightarrow$ Language is finite & so L_4 is Regular

states

$L_5 = \{a^n b^m c^k \mid n, m, k \geq 1\} \Rightarrow$ Regular. (R.E $aa^*bb^*cc^*$)

$L_6 = \{a^i b^{2j} \mid i, j \geq 1\} \Rightarrow$ Regular. ($aa^* \, bb(bb)^*$).

$L_7 = \{a^i b^{4j} \mid i, j \geq 1\} \Rightarrow$ Regular. ($aa^* \, bbbb \, (bbbb)^*$).

(4). $\Sigma = \{a\}$

Arth. progression

$\boxed{\text{Common diff} = 2.}$

$a^n \mid n$ is even $\Rightarrow \{a^0, a^2, a^4, a^6, a^8, \dots\}$

$a^n \mid n$ is odd $\Rightarrow \{a, a^3, a^5, a^7, \dots\}$ AP.

$a^n \mid n$ is prime $\Rightarrow$ ×

$a^{n^2} \mid n \geq 1. \Rightarrow \{a, a^4, a^9, a^{16} \dots\}$ × A.P.

$a^{2^n} \mid n \geq 1 \Rightarrow \{a, a^2, a^4, a^8, a^{16}, \dots\}$ × A.P.

(5). Show that $L = \{0^n \mid n$ is prime$\}$ is not regular.

(i). Let L is regular & n be the number of states in FA.

String $\omega = 0^n$.

(ii). $|\omega| = n$.

$$\omega = 0^n = \underbrace{0^j}_{x} \; \underbrace{0^k}_{y} \; \underbrace{0^{n-j-k}}_{z}$$

$|x| = j$

$|y| = k \geq 1$

$|x \cdot y| = |x| + |y| = j + k \leq n$.

(iii). $i = 0$.

$$0^j (0^k)^i \, 0^{n-j-k} \in L.$$

$j + ki + n - j - k = n + k(i-1)$ is prime for all $i \geq 0$.

if $i = n+1$

then $n + k(i-1) = n + kn = n(k+1)$ is also prime for each $k \geq 1$.

which is contradiction to the assumption that the language is regular.

So $L = \{0^n / n \text{ is prime}\}$ is not regular.