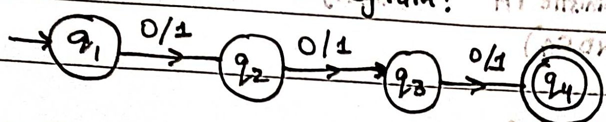
• Finite Automata ⤵

Examples ⤵

(1) Binary language having string length 3

So examples can be 000, 001, 110, 100, etc.

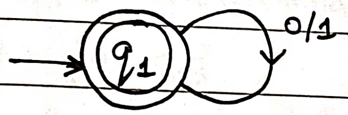
∴ Transition state diagram:



(2) Binary language having any combination of 0 and 1.

So examples can be ∈ (null), 0, 1, 01, 1110, etc.

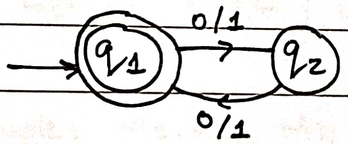
∴ Transition state diagram: (since minimum length = 0, no. of states = 1)



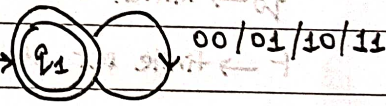
(3) Binary language of even length

So examples can be ∈, 00, 11, 1000, 0110, etc.

∴ Transition state diagram:



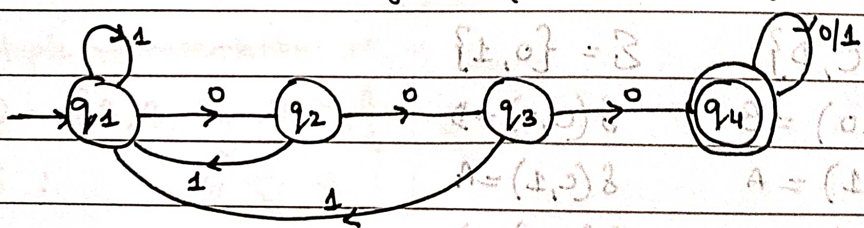
OR



(4) Binary language with 3 consecutive zeroes.

So examples can be 000, 1000, 00011, 101000, etc.

∴ Transition state diagram: (since minimum length = 3, no. of states = 4)



- **Classification of Finite Automata ↓**

Finite Automata

- Language Acceptor (no output)
  - Deterministic FA (DFA)
  - Non-Deterministic FA (NFA / NDFA)
    - ↳ with ϵ move
    - ↳ without ϵ move
- Output Generator
  - Mealy Machine
  - Moore Machine

- **Formal Definition of Automata ↓**

5-tuple $(Q, \Sigma, \delta, q_0, F)$

$Q \rightarrow$ finite set of states

$\Sigma \rightarrow$ finite set of input symbols

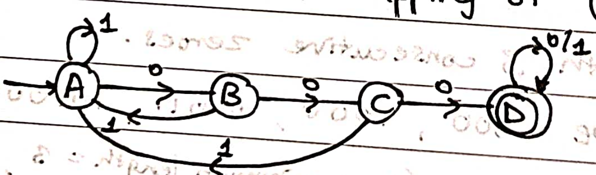
$\delta \rightarrow$ transition function

$q_0 \rightarrow$ initial state

$F \rightarrow$ finite set of final states

[Transition function : It is a mapping of $(Q \times \Sigma) \longrightarrow Q$]

Example :



$Q = \{A, B, C, D\}$ $\Sigma = \{0, 1\}$

$\delta \Rightarrow$

$\delta(A, 0) = B$ $\delta(C, 0) = D$

$\delta(A, 1) = A$ $\delta(C, 1) = A$

$\delta(B, 0) = C$ $\delta(D, 0) = D$

$\delta(B, 1) = A$ $\delta(D, 1) = D$

$q_0 = A$

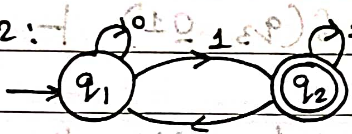
$F = \{D\}$

- **Transition Table** : (another representation method)

Taking same example as previous page,

Q \ Σ	0	1
→ A	B	A
B	C	A
C	D	A
(D)	D	D

Example: Machine M2:



5-tuple representation : $M_2 = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_1, q_2\}$, $\Sigma = \{0, 1\}$, $q_0 = q_1$, $F = \{q_2\}$

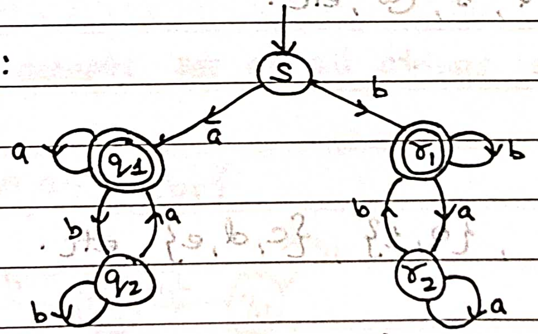
for δ:

Q \ Σ	0	1
→ q_1	q_1	q_2
(q_2)	q_1	q_2

Also,

$L(M_2) = \{w \mid w \text{ ends in a } 1\}$

Example: Machine M4:



5-tuple representation: $M_4 = (Q, \Sigma, \delta, q_0, F)$

$Q = \{S, q_1, q_2, r_1, r_2\}$, $\Sigma = \{a, b\}$, $q_0 = S$, $F = \{q_1, r_1\}$

δ :

Q \ Σ	a	b
S	q_1	r_1
q_1	q_1	q_2
q_2	q_1	q_2
r_1	r_2	r_1
r_2	r_2	r_1

$L(M4) = \{w \mid w$ begins and ends with same letter$\}$

Q. Consider transition table and check if '110101' lies in L(M). Given initial state q_0, and $F = \{q_0\}$, also write 5-tuple representation.

	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Ans. To check by transition table,

$\delta(q_0, \underline{1}10101) \vdash \delta(q_1, \underline{1}0101) \vdash \delta(q_0, \underline{0}101)$
$\vdash \delta(q_2, \underline{1}01) \vdash \delta(q_3, \underline{0}1) \vdash \delta(q_1, \underline{1}) \vdash q_0$

Thus, final state is accepted. String lies in L(M).

Also,

$Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, (others are already given)

→ Symbols:

E.g. \$, a, b, @, etc.

→ Alphabets:

E.g. {a,b}, {0,1}, {c,d,e}, etc.

→ Strings:

E.g. aa, abba, abcda, etc.

→ Language:

Example.

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

L_1 = set of all strings with length over 2

$\therefore L_1 = \{aaa, aab, abba, baaaa, \ldots\}$

- **Powers of Σ:**

  Σ represents alplabet

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

  Σ^1 = all strings of length 1 = a, b

  Σ^2 = all strings of length 2 = aa, ab, ba, bb

  Σ^3 = all strings of length 3 = ...

  and,

  $\Sigma^0 = \epsilon$ (epsilon represents null)

  Also, $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$

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

Σ^* $\boxed{L_1 \quad L_2 \quad L_3}$

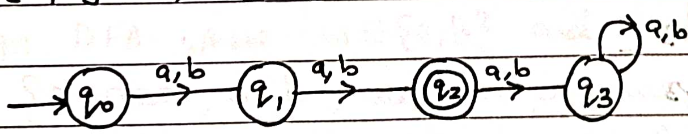
$L_1 \subseteq \Sigma^*$

$L_2 \subseteq \Sigma^*$

$L_3 \subseteq \Sigma^*$

Q. Construct a DFA that accepts set of all strings over $\Sigma = \{a, b\}$ of length 2.

Ans. $\Sigma = \{a, b\}$, $L_1 = \{aa, ab, ba, bb\}$



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

$q_0 = q_0$

$F = \{q_2\}$

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

$\delta(q_0 \times a) \Rightarrow q_1$ $\qquad \delta(q_0 \times b) \Rightarrow q_1$

$\delta(q_1 \times a) \Rightarrow q_2$ $\qquad \delta(q_1 \times b) \Rightarrow q_2$

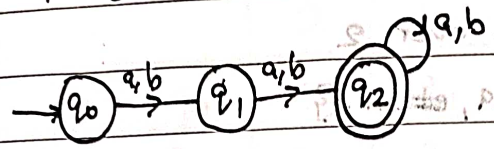
$\delta(q_2 \times a) \Rightarrow q_3$ $\qquad \delta(q_2 \times b) \Rightarrow q_3$

$\delta(q_3 \times a) \Rightarrow q_3$ $\qquad \delta(q_3 \times b) \Rightarrow q_3$

(Do for 1 from L), otherwise, do $\delta(q_0 \times ab) \Rightarrow \delta(q_1 \times b) \Rightarrow q_2$ (simulation)

Q. $w = \{a, b\}$, $|w| \geq 2$. Construct DFA.

Ans. $L_1 = \{aa, ab, ba, bb, aaa, aab, \ldots\}$



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

$q_0 = q_0$

$F = \{q_2\}$

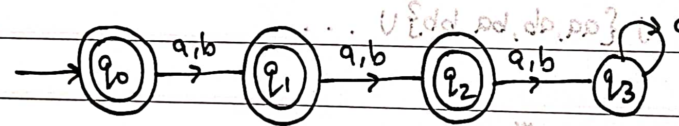
For ab → $\delta(q_0 \times ab) \Rightarrow \delta(q_1 \times b) \Rightarrow q_2$

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

	a	b
→ q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_2	q_2

Q. $W = \{a, b\}$, $|w| \leq 2$ Construct DFA.

Ans. $L_1 = \{\epsilon, a, b, aa, ab, ba, bb\}$



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

$q_0 = q_0$

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

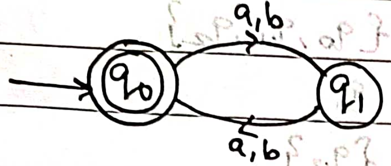
$\Sigma = \{a, b\}$

For ~~ab~~ ba → $\delta(q_0 \times ba) \Rightarrow \delta(q_1 \times a) \Rightarrow \cancel{\delta(q_2)}\ q_2$

	a	b
→ q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_3	q_3

Q. Design a DFA where $w \in \{a,b\}$ and length of string is even.

Ans. $L_1 = \{\epsilon, aa, bb, ab, ba, aaaa, \ldots\}$



$Q = \{q_0, q_1\}$

$q_0 = q_0$

$F = \{q_0\}$

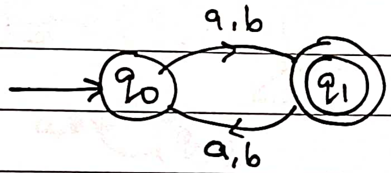
For aa $\rightarrow$ $\delta(q_0 \times aa) \Rightarrow \delta(q_1 \times a) \Rightarrow q_0$

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

	a	b
$\rightarrow$ (q0)	q_1	q_1
q_1	q_0	q_0

Q. Design DFA where $w \in \{a,b\}$ and $|w| \bmod 2 = 1$.

Ans. $L_1 = \{a, b, aaa, aba, aaaab, \ldots\}$



$Q = \{q_0, q_1\}$

$q_0 = q_0$

$F = \{q_1\}$

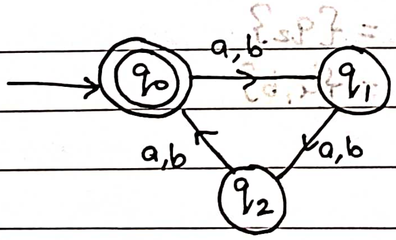
For aba $\rightarrow$ $\delta(q_0 \times aba) \Rightarrow \delta(q_1 \times ba) \Rightarrow \delta(q_0 \times a) \Rightarrow q_1$

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

	a	b
$\rightarrow q_0$	q_1	q_1
(q1)	q_0	q_0

H.W. Q. Design DFA where $w \in \{a,b\}$ and $|w| \bmod 3 = 0$.

Ans. $L_1 = \{\epsilon, aaa, aab, aabbaa, abbaaa, \ldots\}$



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

$q_0 = q_0$

$F = \{q_0\}$

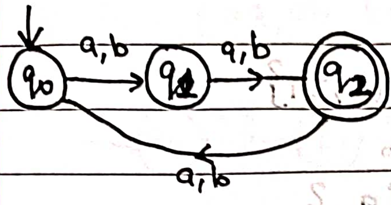
	a	b
$\rightarrow$ (q0)	q_1	q_1
q_1	q_2	q_2
q_2	q_0	q_0

For aab $\rightarrow$ $\delta(q_0 \times aab) \Rightarrow \delta(q_1 \times ab) \Rightarrow \delta(q_2 \times b) \Rightarrow q_0$

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

Q. Design DFA where $W = \{a, b\}$ and $|w| \bmod 3 = 2$

Ans. $L_1 = \{aa, ab, aabbb, abaaa, \ldots\}$



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

$q_0 = q_0$

$F = \{q_2\}$

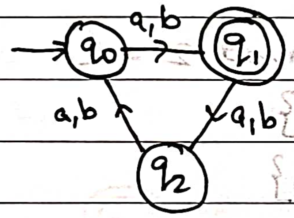
$\Sigma = \{a, b\}$

	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2 (final)	q_0	q_0

$\delta(q_0 \times abb) \Rightarrow \delta(q_1 \times bb) \Rightarrow \delta(q_2 \times b) \Rightarrow q_0$

Q. $W = \{a, b\}$ and $|w| \bmod 3 = 1$

Ans. $L_1 = \{a, b, aabb, abaa, baaa, \ldots\}$



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

$q_0 = q_0$

$F = \{q_1\}$

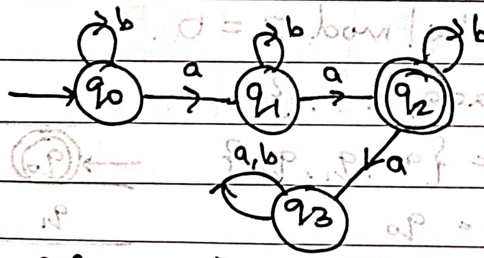
$\Sigma = \{a, b\}$

	a	b
$\rightarrow q_0$	q_1	q_1
q_1 (final)	q_2	q_2
q_2	q_0	q_0

$\delta(q_0 \times a) \Rightarrow q_1$

Q. $n_a(w) = 2$ (means number of 'a's is 2)

Ans. $L_1 = \{aa, aba, aab, aabbb, ababb, \ldots\}$



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

$q_0 = q_0$

$F = \{q_2\}$

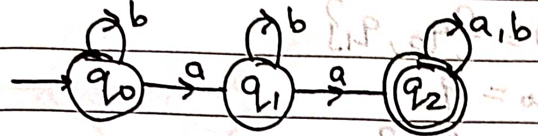
$\Sigma = \{a, b\}$

$\delta(q_0 \times aa) \Rightarrow \delta(q_1 \times a) \Rightarrow q_2$

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
q_2 (final)	q_3	q_2
q_3	q_3	q_3

Q. $n_a(w) \geq 2$

Ans. $L_1 = \{aa, aaba, aaab, \ldots\}$



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

$q_0 = q_0$

$F = \{q_2\}$

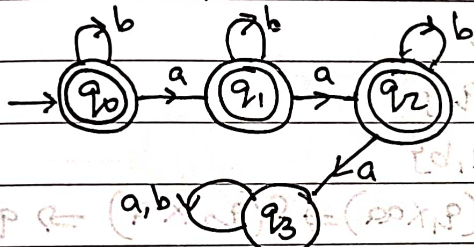
$\Sigma = \{a, b\}$

$\delta(q_0 \times aa) \Rightarrow \delta(q_1 \times a) \Rightarrow q_2$

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
q_2	q_2	q_2

Q. $n_a(w) \leq 2$

Ans. $L_1 = \{aa, abb, aabb, \ldots\}$



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

$q_0 = q_0$

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

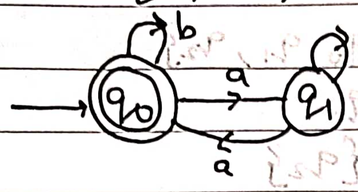
$\Sigma = \{a, b\}$

$\delta(q_0 \times abb) \Rightarrow \delta(q_1 \times bb) \Rightarrow \delta(q_1 \times b) \Rightarrow q_1$

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

Q. $w = \{a,b\}^*$, $n_a(w) \bmod 2 = 0$

Ans. $L_1 = \{\epsilon, aa, aab, abaaa, \ldots\}$



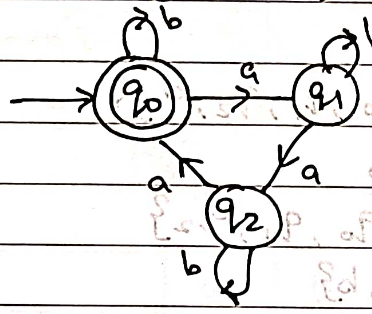
$Q = \{q_0, q_1\}$
$q_0 = q_0$
$F = \{q_0\}$
$\Sigma = \{a,b\}$

$\delta(q_0 \times \epsilon) \Rightarrow q_0$

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_0	q_1

Q. $w = \{a,b\}^*$, $n_a(w) \bmod 3 = 0$

Ans. $L_1 = \{\epsilon, aaa, aaab, baaaaaa, \ldots\}$



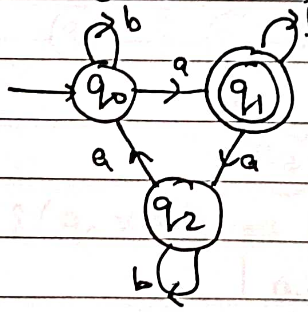
$Q = \{q_0, q_1, q_2\}$
$q_0 = q_0$
$F = \{q_0\}$
$\Sigma = \{a,b\}$

$\delta(q_0 \times aaa) \Rightarrow \delta(q_1 \times aa) \Rightarrow \delta(q_2 \times a) \Rightarrow q_0$

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
q_2	q_0	q_2

Q. $w = \{a,b\}^*$ $n_a(w) \bmod 3 = 1$

Ans. $L_1 = \{a, abb, aaaba, \ldots\}$



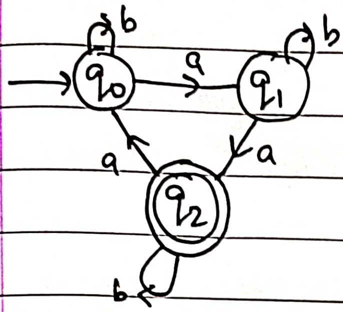
$Q = \{q_0, q_1, q_2\}$
$q_0 = q_0$
$F = \{q_1\}$
$\Sigma = \{a,b\}$

$\delta(q_0 \times a) \Rightarrow q_1$

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
q_2	q_0	q_2

Q. $w = \{a, b\}^*$, $n_a(w) \bmod 3 = 2$

Ans. $L_1 = \{aa, aab, baaaaa, \ldots\}$



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

$q_0 = q_0$

$F = \{q_2\}$

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

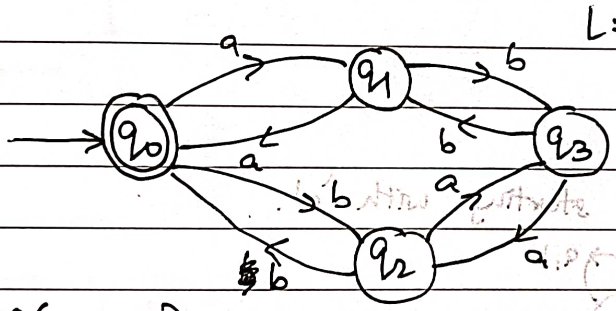
$\delta(q_0 \xrightarrow{x} aa) \Rightarrow \delta(q_1, xa) \Rightarrow q_2$

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
q_2	q_0	q_2

Q. $w \in \{a, b\}^*$, $n_a(w) \bmod 2 = 0$, $n_b(w) \bmod 2 = 0$

Ans.

$L = \{\epsilon, aabb, abab, \ldots\}$



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

$q_0 = q_0$

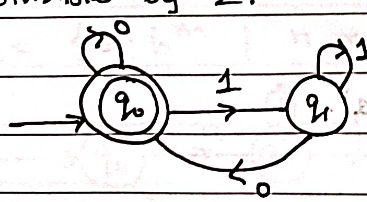
$F = \{q_0\}$

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

$\delta(q_0 \times \epsilon) \Rightarrow q_0$

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

Q. Divisible by 2.

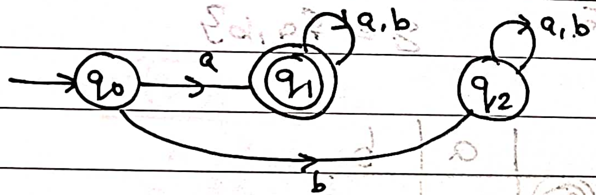


Q. Divisible by 4.

Q. Divisible by 3.

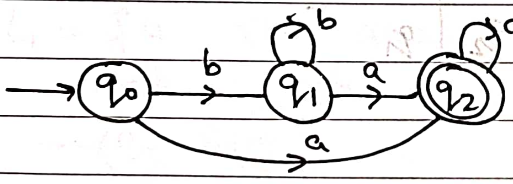
Q. Design DFA for all strings starting with 'a'.

Ans.



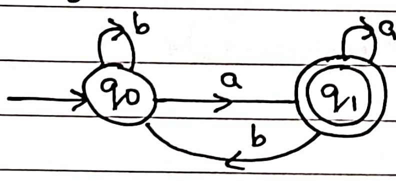
Q. Design DFA for all strings containing 'a'.

Ans.



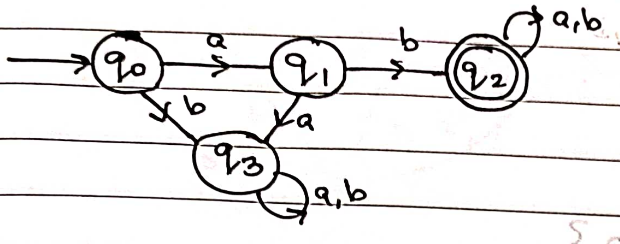
Q. Design DFA for all strings ending with 'a'.

Ans.



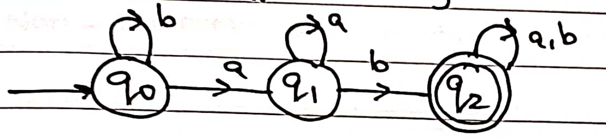
Q. Design DFA that accepts a string starting with 'ab'.

Ans.



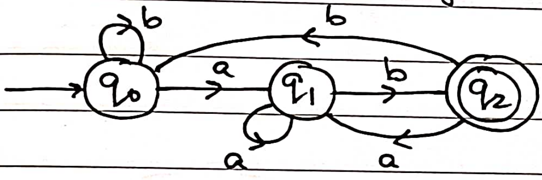
Q. Design DFA for strings containing ~~aba~~ 'ab'.

Ans.



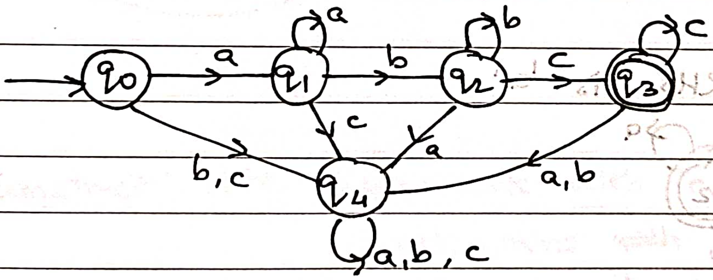
Q. Design DFA for strings ending with 'ab'.

Ans.



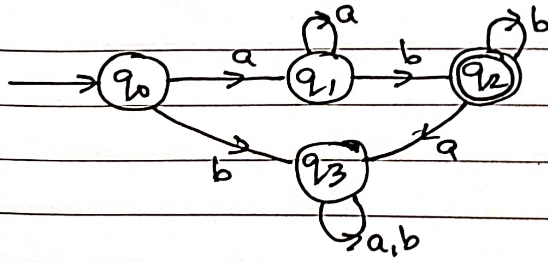
Q. $L = \{a^n b^m c^l \mid n, m, l >= 1\}$

Ans.



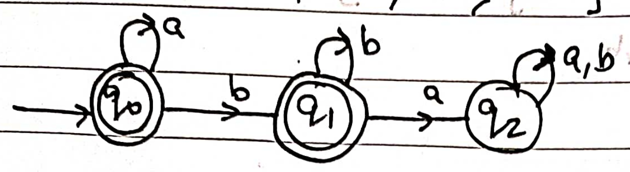
Q. $L = \{a^n b^m \mid n, m >= 1\}$

Ans.



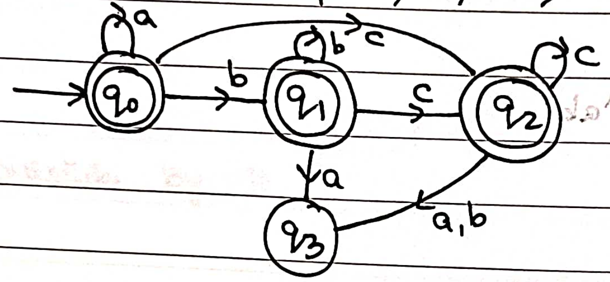
Q. $L = \{a^n b^m \mid n, m \geq 0\}$

Ans.



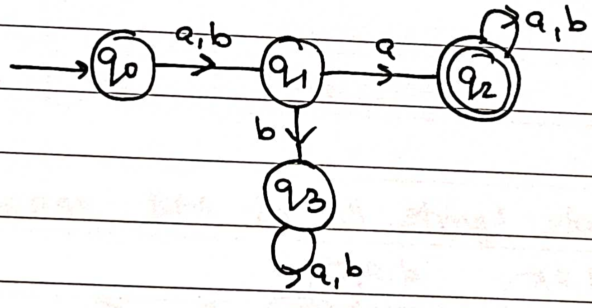
Q. $L = \{a^n b^m c^l \mid n, m, l \geq 0\}$

Ans.



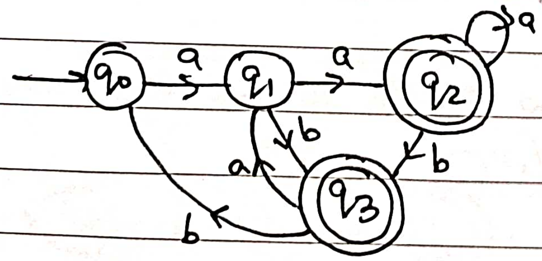
Q. Second symbol from LHS is 'a'.

Ans.



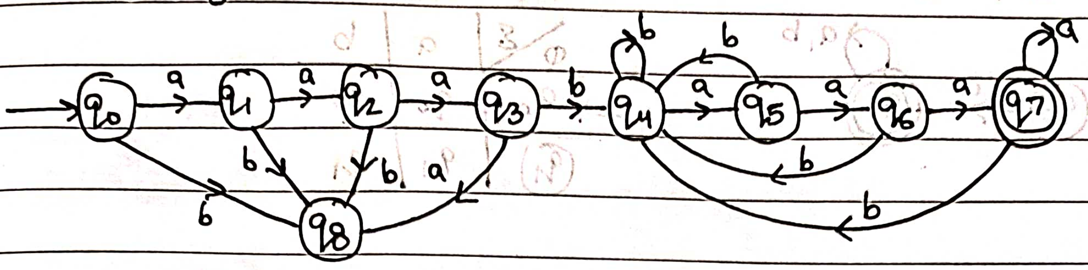
Q. Second symbol from RHS is 'a'.

Ans.



Q. $w = \{a,b\}^*$ $a^3 b w a^3$

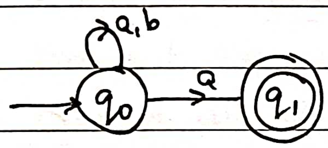
Ans.

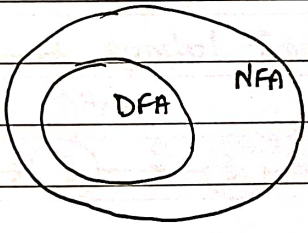


- Non-deterministic Finite Automata (NFA)

Q. Design NFA that ends with 'a' over ~~the~~ input alphabet a,b.

Ans.





Q. Construct NFA that (i) starts with 'a'.

(ii) contains ~~with~~ 'a'

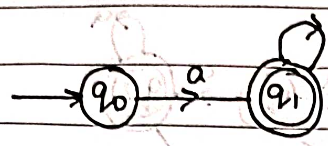
(iii) starts with 'ab'.

(iv) contains 'ab'.

(v) ends with 'ab'.

(i) starts with 'a'.

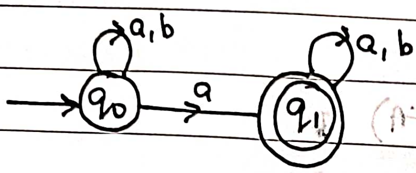
Ans.



ϕ \ Σ	a	b
$\rightarrow q_0$	q_0	ϕ
(q_1)	q_1	q_1

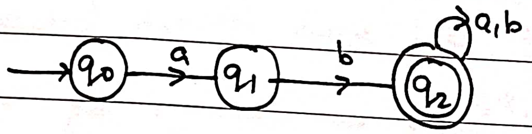
(ii) contains 'a'.

Ans.



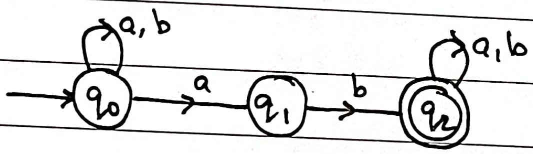
(iii) starts with 'ab'.

Ans.



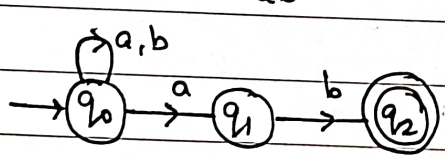
(iv) contains 'ab'.

Ans.



(v) ends with 'ab'.

Ans.



★★ IMP!

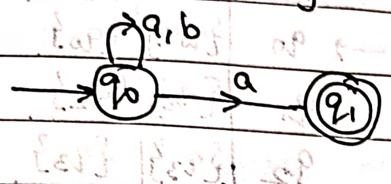
To convert NFA to DFA:

- Draw NFA.
- Draw its transition table.
- Replace 'φ' with dead state.
- Any input on dead state goes back to itself.
- Draw DFA from modified transition table.

Date: 6|2|23

Q. Convert ~~Construct~~ NFA to DFA for:
(i) Strings ending with 'a'.

Ans.



(NFA)

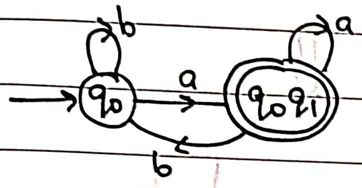
$\Rightarrow$

Q \ Σ	a	b
→ q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	{ }	{ }

$\Downarrow$

Q \ Σ	a	b
→ $[q_0]$	$[q_0 q_1]$	$[q_0]$
$[q_0 q_1]$	$[q_0 q_1]$	$[q_0]$

$\Leftarrow$

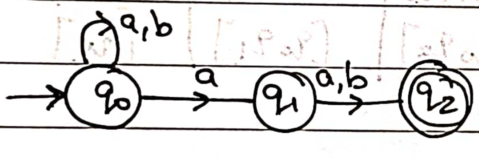


(DFA)

☆☆ IMP!
- → Starts with starting state.
- → Find out the next reachable state.
- → Discard all non-reachable states. Look for new states.

Q. Second symbol from RHS is 'a'.

Ans.



(NFA)

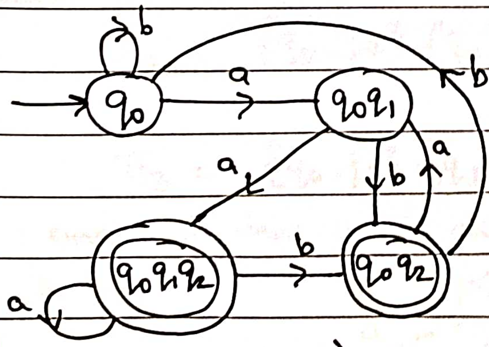
$\Rightarrow$

Q \ Σ	a	b
→ q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	{ }	{ }

$\Downarrow$

Q \ Σ	a	b
→ $[q_0]$	$[q_0 q_1]$	$[q_0]$
$[q_0 q_1]$	$[q_0 q_1 q_2]$	$[q_0 q_2]$
* $[q_0 q_1 q_2]$	$[q_0 q_1 q_2]$	$[q_0 q_2]$
* $[q_0 q_2]$	$[q_0 q_1]$	$[q_0]$

$\Leftarrow$

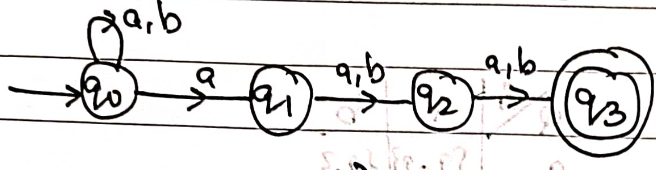


(DFA)

☆☆
→ When writing table for new states, use union of old table results.

Q. Design NFA for third symbol from RHS is 'a'. Convert to DFA.

Ans.



(NFA)

⇒

Q \ Σ	a	b
→ q_0	$\{q_0 q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$
q_3	{ }	{ }

⇓

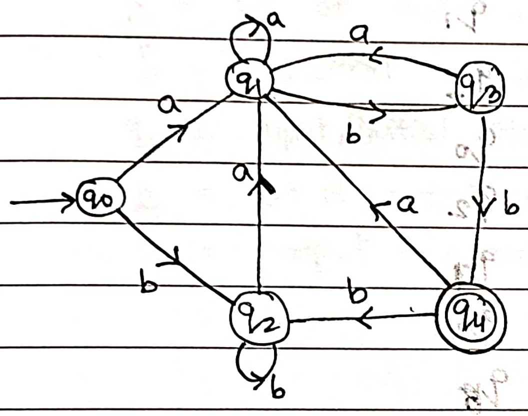
Q \ Σ	a	b
→ $[q_0]$	$[q_0 q_1]$	$[q_0]$
$[q_0 q_1]$	$[q_0 q_1 q_2]$	$[q_0 q_2]$
$[q_0 q_1 q_2]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_2 q_3]$
$[q_0 q_2]$	$[q_0 q_1 q_3]$	$[q_0 q_3]$
* $[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_2 q_3]$
* $[q_0 q_2 q_3]$	$[q_0 q_1 q_3]$	$[q_0 q_3]$
* $[q_0 q_1 q_3]$	$[q_0 q_1 q_2]$	$[q_0 q_2]$
* $[q_0 q_3]$	$[q_0 q_1]$	$[q_0]$

- Minimization of DFA ↓

Rules:

→ Identify start and end/final state.

→ Try to delete all the states to which we cannot reach from initial state i.e. we need to check reachability



⇒

Q \ Σ	a	b
→ q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4
(q_4)	q_1	q_2

zero equivalent

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

set of non-final states — set of final states

1 equivalent — Checking from this, if transitions to state that is not part of group, is put in a separate group

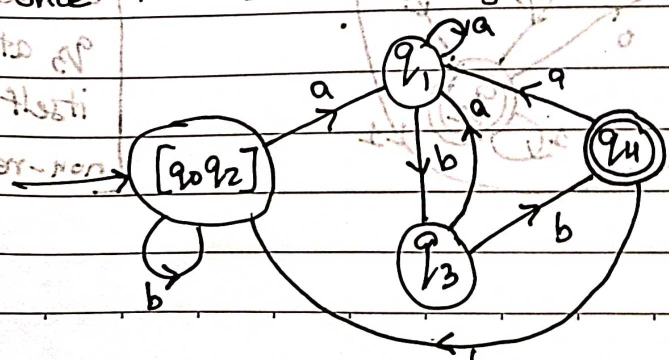
$\Pi_1 = \{q_0, q_1, q_2\}\ \{q_3\}\ \{q_4\}$

Similarly,

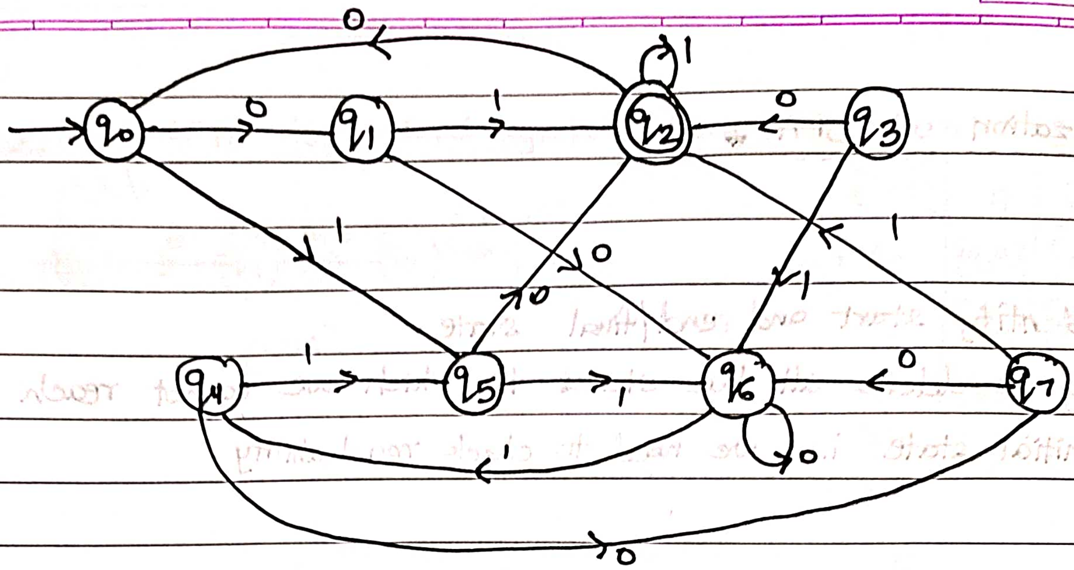
$\Pi_2 = \{q_0, q_2\}\ \{q_1\}\ \{q_3\}\ \{q_4\}$

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

Once it does not change, we have gotten final minimization.



Q.



Ans.

Q \ Σ	1	0
→ q_0	q_5	q_1
q_1	q_2	q_6
q_2	q_2	q_0
q_3	q_6	q_2
q_4	q_5	q_7
q_5	q_6	q_2
q_6	q_4	q_6
q_7	q_2	q_6

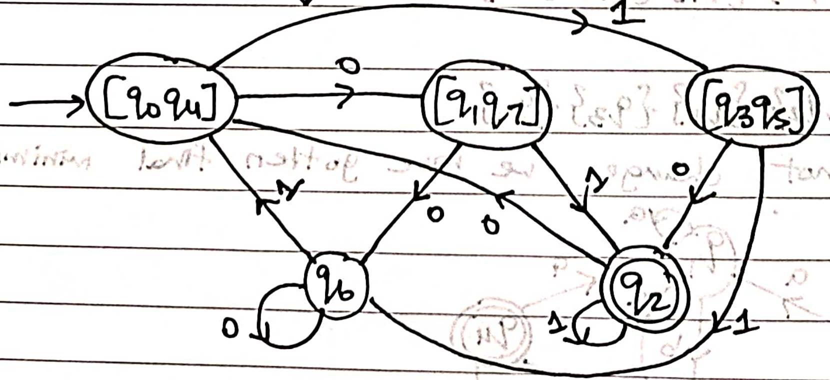
$\Pi_0 = \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}\{q_2\}$.

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

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

$\Pi_3 = \{q_0, q_4\}\{q_6\}\{q_1, q_7\}\{q_3, q_5\}\{q_2\}$ ← same found again

Minimized DFA ↓



!!!

Should remove q_3 at start itself, due to non-reachability

→ FA with output

Moore Machine | Mealy Machine

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

$(Q \times \Sigma = Q)$

(Δ = output symbols, λ = output function)

$(\lambda : Q \to \Delta \mid \lambda : Q \times \Sigma \to \Delta)$

(Moore) — [output symbol is state-dependent]

(Mealy) — [~~output symbol is state~~ and input symbol dependent]

Q = set of states

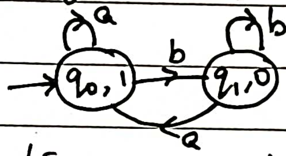
Σ = set of input symbols

δ = transition function

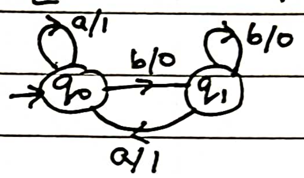
q_0 = input/initial state

Δ = set of output symbols

λ = output function



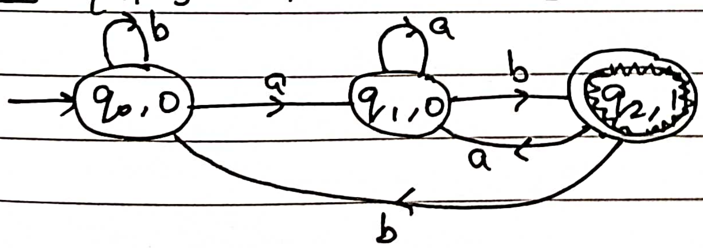
(For every state, we have one output)



Q. Construct a Moore machine for counting the occurrences of substring 'ab'.

Ans. Construct a Moore machine that takes ~~state~~ set of all strings over input alphabet (a,b), and prints '1' as output for every occurrence of 'ab' as substring.

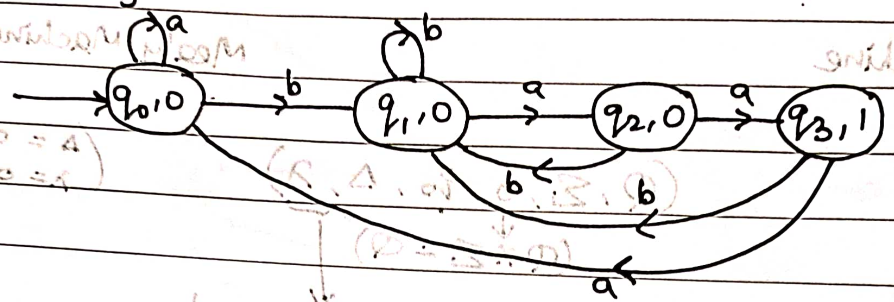
$\Sigma = \{a, b\}$, $\Delta = \{0, 1\}$



(no final state)

Q. Design a Moore Machine for counting the occurrences of substring 'baa'.

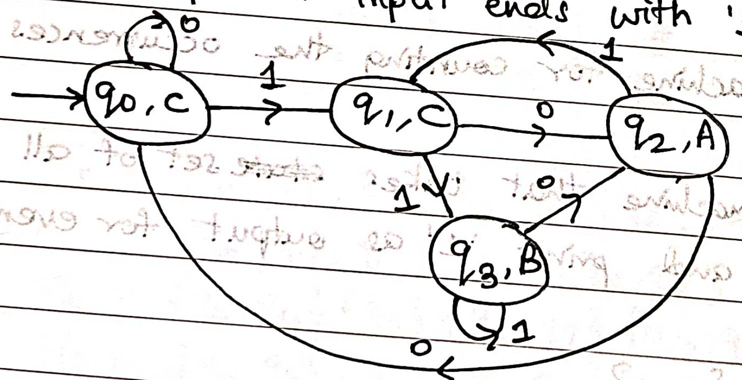
Ans.



Gives output '1' on every occurrence of 'baa', otherwise output '0'

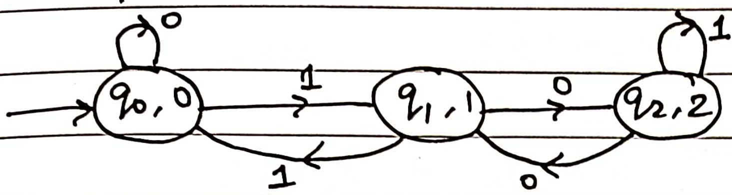
Q. Construct a Moore machine that takes input alphabet {0,1} and produces 'A' as output if input ends with '10' or produces 'B' as output if input ends with '11', otherwise produces 'C'.

Ans.



Q. Construct a Moore Machine that takes binary number as input, and produces residue modulo 3 as output.

Ans.



Q. Construct a Moore Machine that takes base 4 number as input and produces residue modulo 5 as output

Ans.

Q \ Σ	0	1	2	3
q_0	q_0	q_1	q_2	q_3
q_1	q_4	q_0	q_1	q_2
q_2	q_3	q_4	q_0	q_1
q_3	q_2	q_3	q_4	q_0
q_4	q_1	q_2	q_3	q_4

→ RL (Regular Language)

- Representor → RE (Regular Expression)
- generator → RG (Regular grammar)
  - LLG (Left Linear Grammar)
  - RLG (Right Linear Grammar)
- acceptor → FA
  - with output
    - Mealy machine
    - Moore machine
  - without output
    - DFA
    - NFA
    - ϵ-NFA

$+$ union

$\cdot$ concatenation

$*$ clean closure

$\phi = \{\}$, $\epsilon = \{\epsilon\}$, $a = \{a\}$

$a^* = \{\epsilon, a, aa, aaa, \dots\}$

$a^+ = a \cdot a^* = a^* \cdot a = \{a, aa, aaa, \dots\}$

$(a+b)^* = \{\epsilon, a, b, ab, aa, bb, ba, \dots\}$

Q. Write regular expression over input symbol $\{a,b\}$ where set of all strings size whose length is 2.

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

$L_1 = \{aa, ab, ba, bb\}$ language is finite.

$aa + ab + ba + bb$

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

$= (a+b) \cdot (a+b)$ //

Q. Write regular expression over input symbol {a, b} where set of all strings size whose length is 3.

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

$L_1 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$ Language is finite

$aaa + aab + aba + abb + baa + bab + bba + bbb$

$= a\cdot(aa + ab + ba + bb) + b\cdot(aa + ab + ba + bb)$

$= (a+b)\cdot(aa + ab + ba + bb)$

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

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

$= (a+b)\cdot(a+b)\cdot(a+b)$ //

Q. Write reguler expression over input symbol {a, b} where set of all strings whose length is 4.

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

$L_1 = \{aaaa, aaab, aaba, aabb, abaa, abab, abba, abbb, baaa, baab, baba, babb, bbaa, bbab, bbba, bbbb\}$

$aaaa + aaab + aaba + aabb + abaa + abab + abba + abbb + baaa + baab + baba + babb + bbaa + bbab + bbba + bbbb$

$= a$

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

Q. Write regular expression over $\{a,b\}$ with string length atleast 2.

Ans. $(a+b)\cdot(a+b) \longrightarrow$ string length 2

For atleast 2,

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

OR

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

Q. Write regular expression over $\{a,b\}$ with string length at most 2.

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

$\epsilon + a + b + aa + ab + ba + bb$

$= \epsilon + (a+b) + (a+b)\cdot(a+b)$

$(a+b) = (a+b+\epsilon) \leftarrow$ We know

$= (a+b+\epsilon) + (a+b+\epsilon)\cdot(a+b+\cancel{\epsilon})$

Also,

$\epsilon\cdot(a+b+\epsilon) = (a+b+\epsilon)$

$\therefore \epsilon\cdot(a+b+\epsilon) + (a+b)\cdot(a+b+\epsilon)$

$= (a+b+\epsilon)\cdot(a+b+\epsilon)$

Q. Length of string should be even. Write regular expression over $\{a,b\}$

Ans. $L = \{\epsilon, aa, ab, ba, bb, aaaa, aaab, \ldots\}$

For length 2,

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

For getting all multiples of 2, (and ϵ),

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

Q. Length of string is odd. Write regular expression over {a, b}.

Ans. L = {a, b, aaa, abb, baa, bbbbb, . . .}

We know for even length,

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

$\therefore$ For odd length,

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

Q. length of string divisible by 3.

Ans. ~~We know~~ L = {aaa, ϵ, aabbaa, . . .}

We know for divisible by 2 (even),

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

$\therefore$ For divisible by 3,

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

Q. Length of string mod 3 = 2.

L = {aa, ab, aabba, . . .}

For divisible by 3,

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

$\therefore$ For mod 3 = 2,

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

☆ (Practice)

Q.1 Number of 'a's should be exactly 2.

Q.2 Number of 'a's atleast 2.

Q.3 Number of 'a's at most 2.

Q.4 Number of 'a's even.

Q.5 Starts with 'a'.

Q.6 Ends with 'a'.

Q.7 Containing 'a'.

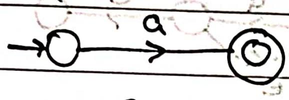
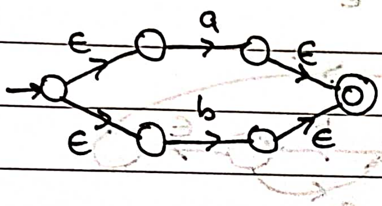
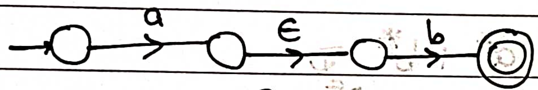
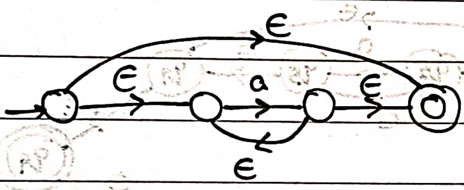
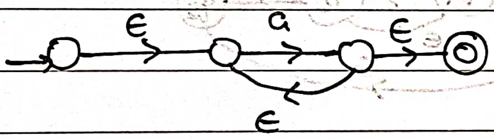
Q.8 Starting and ending with different symbol.

Q.9 Starting and ending with same symbol.

Q.10 No 2 'a's together.

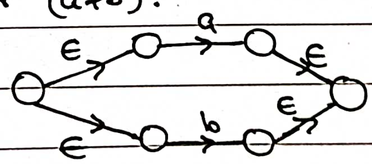
~~Q.11 Se~~

→ Regular Expression to Finite Automata

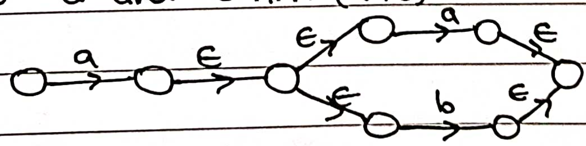
R.E	ε-NFA
① ε	
② a	
③ a+b	
④ ab	
⑤ a^*	
⑥ a^+	

Q. Draw NFA with ε moves for regular expression a·(a+b)·b

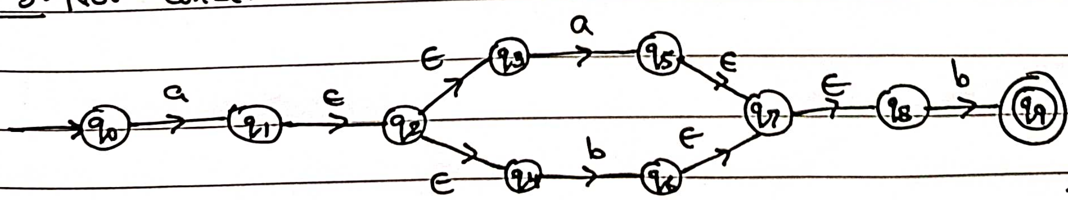
Ans. Step 1: Drawing ε-NFA for (a+b):



Step 2: Concatenate a and ε-NFA (a+b):

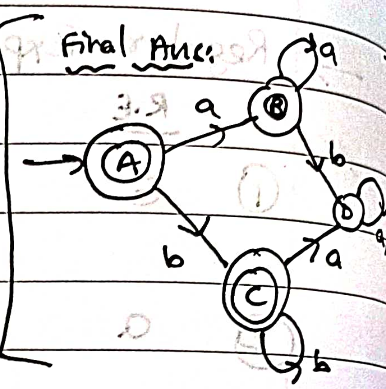


Step 3: Now concatenate a·(a+b) with b:



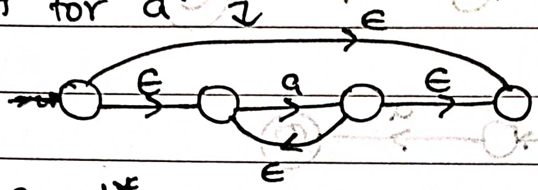
Q. Draw DFA for regular expression $a^* + b^*$

Final Ans:

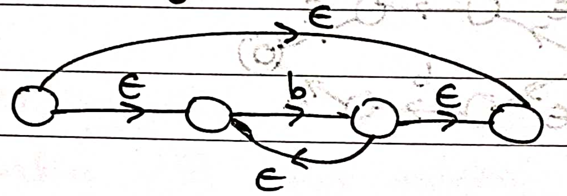


Ans. First converting to ε-NFA:

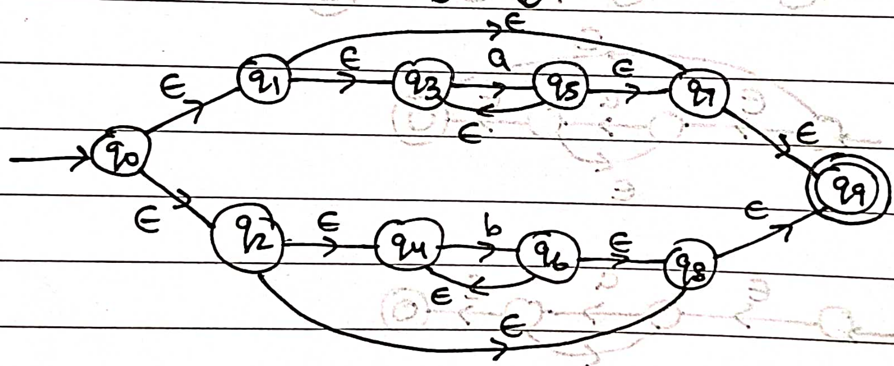
Step 1: ε-NFA for a^* ↴



Step 2: ε-NFA for b^* ↴



Step 3: ε-NFA for $a^* + b^*$ ↴

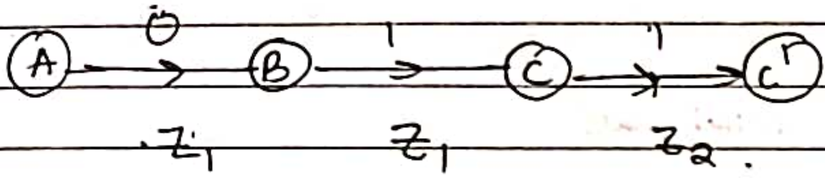


Step 4: Converting ε-NFA to NFA ↴

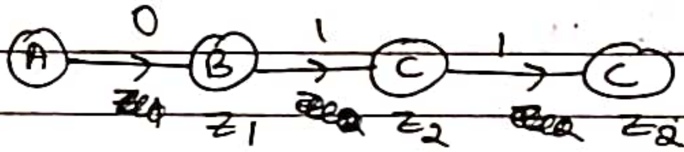
Simulation:

(i) Moore:

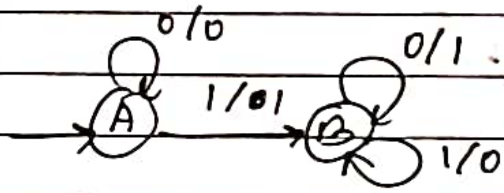
~~δ(A, 0010) ⊢ δ(B~~



(ii) Mealy:

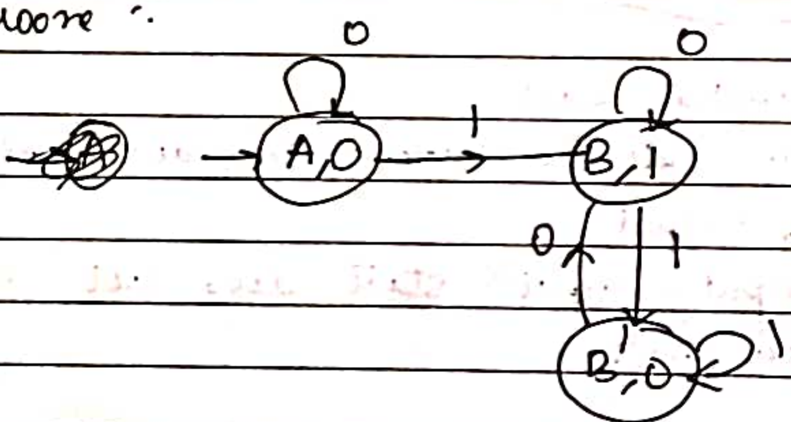


Q

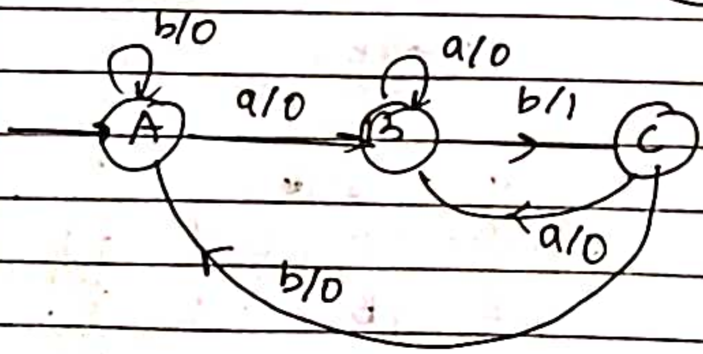


Mealy

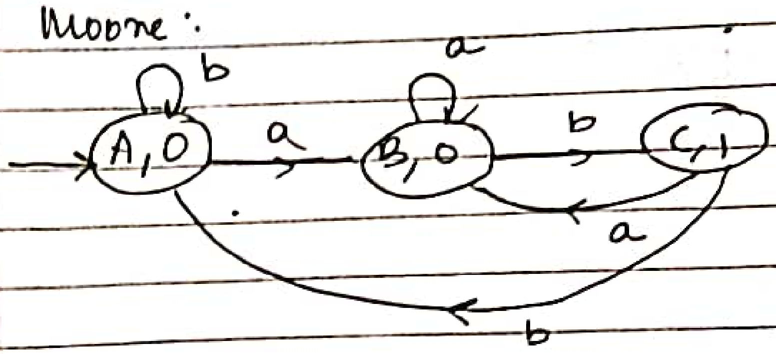
Moore:



Q



Moore :

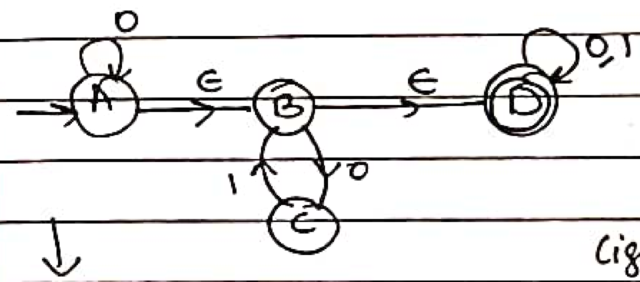


* ε NFA (Epsilon NFA)

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

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

Q



(ignore dead state - if any)

Conversion (ENFA to NFA) :

ϵ^*	0	ϵ^*
A — A	A — A	A, B, D
A — B	C	C
A — D	D	D

ϵ^*	0	ϵ^*
C — C	ϕ	

ϵ^*	1	ϵ^*
A — A	ϕ	
A — B	ϕ	
A — D	D	D

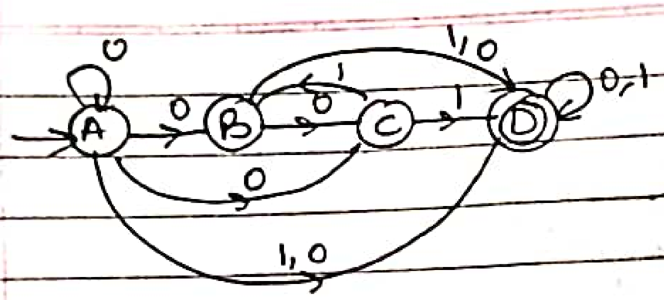
ϵ^*	0	ϵ^*
B — D	D	D
B — B	C	C

ϵ^*	1	ϵ^*
B — D	D	D
B — B	ϕ	

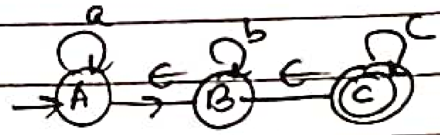
ϵ^*	1	ϵ^*
C — C	B	D, B

ϵ^*	0	ϵ^*
D — D	D	D

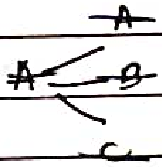
ϵ^*	1	ϵ^*
D — D	D	D



Q



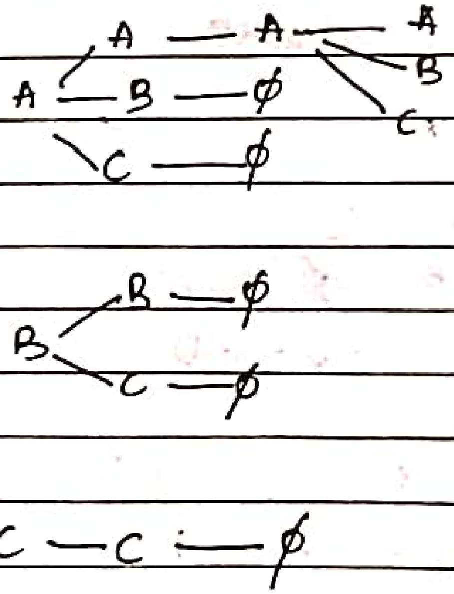
~~ε*~~ ~~a~~ ~~ε*~~

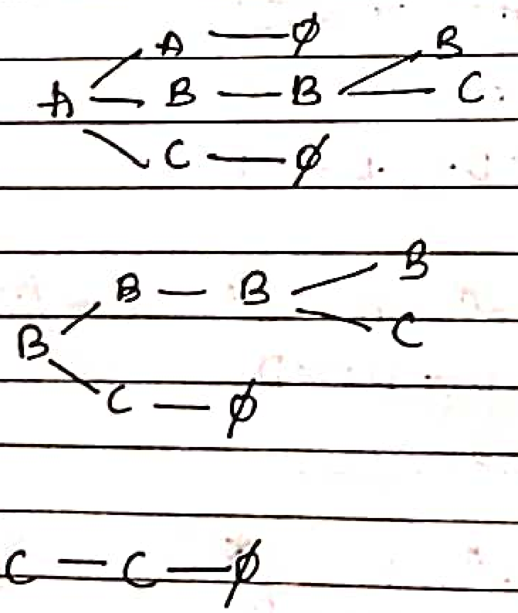


~~x ignore~~

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

ε* a ε* ε* b ε*





Date

ϵ^* $\quad c \quad$ ϵ^*

A — A — ϕ

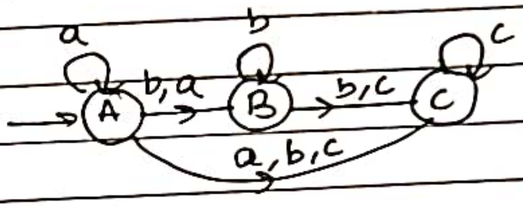
A — B — ϕ

A — c — C — C

B — B — ϕ

B — c — C — C

C — C — C — C.



☆ (Practice)

Q. Draw ε-NFA for : (or even NFA/DFA)

(i) $(ab/ba)^* aa (ab/ba)^*$

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

(iii) $(a+bb)^* (ba^* + \epsilon)$

(iv) $(00+1)^* (10)^*$

(v) $((0+1)^* 10 + (00)^* (11)^*)^*$

(vi) $1 (011)^* 0$

(vii) $(1(00)^* 1 + 01^* 0)^*$

(viii) $(00+11)^* (10)^*$

(ix) $(0+\epsilon)(10)^* (\epsilon+1)$

- Arden's Theorem :

Let 'P', 'Q' and 'R' be regular expression and

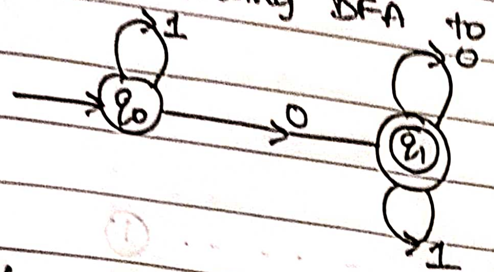
→ if $R = P + RQ$ or $R = RQ + P$
then R can be simplifired as

$$\boxed{R = PQ^*}$$

→ if $R = P + QR$ or $R = QR + P$
then R can be simplifired as

$$\boxed{R = Q^* P}$$

Q. Convert following DFA to regular expression.



Ans. Finding expressions:

$q_0 = q_0 1 + \epsilon$ ------ ①

$q_1 = q_0 0 + q_1 0 + q_1 1$ ---- ②

Applying Arden's Theorem on ①,

$q_0 = q_0 1 + \epsilon$ $\quad\therefore R = q_0,\ P = \epsilon,\ Q = 1$

$\therefore R = PQ^*$

$q_0 = \epsilon 1^* = 1^*$

Using ②

$q_1 = q_0 0 + q_1 0 + q_1 1$

Substituting $q_0 = 1^*$ in ②,

$q_1 = 1^* 0 + q_1 (0+1)$ ---- ③

Apply Arden's Theorem on ③,

$R = q_1,\ P = 1^* 0,\ Q = (0+1)$

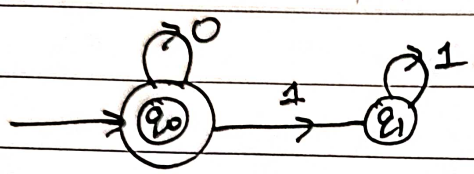
$\therefore R = PQ^*$

$\therefore q_1 = 1^* 0 (0+1)^*$

$RE = q_0 + q_1$

$\therefore$ Regular Expression: $1^* 0 (0+1)^*$

Q. Convert ~~DFA~~ FA to RE.

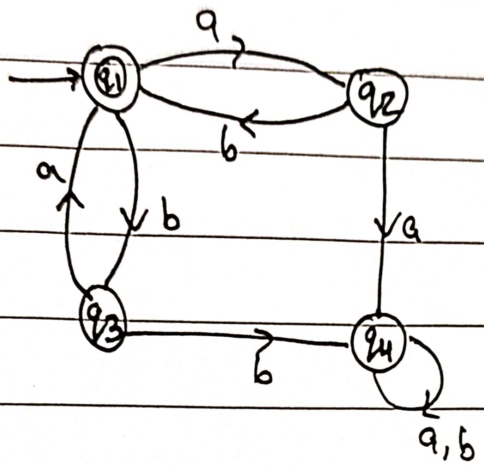


Ans. finding expressions: $q_0 = q_0 0 + \epsilon$ .... (1)

$q_1 = q_0 1 + q_1 1$ .... (2)

Q. Convert FA to RE:

(i)



(ii) set of all strings over $\{1,0\}$ that ends with 1 and has no substrings 00. (~~convert~~ make DFA, then RE)

Date: 6/3/23

- **Pumping Lemma** ↓ (Read from notes)

Example –

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

Step 1: Suppose L is regular

Let n be number of states in FA accepting L

Step 2: Choose a string $z \in L$ such that $|z| > n$.

z can be written in the form $z = uvw$ with $|uv| \leq n$ and $|v| \geq 1$ such that $uv^iw \in L$ for $i > 0$.

Consider $a^m b^m$, $m > n$

aaaa . . . aabbb . . . bb

Using Pumping Lemma, this can be written as $z = uvw$, such that $|uv| \leq n$ and $|v| \geq 1$

As $m > n$, and $|uv| \leq n$,

$$\underbrace{aaaaaa \ldots aaa}_{uv}\underbrace{bbb \ldots bbbb}_{w}$$

'v' does not fall in "b".

Step 3: Let $v = a^p$, $u = a^q$, $w = a^r b^m$

$\therefore p + q + r = m$

Let $z = a^q (a^p) a^r b^m$

$\therefore$ Pumping 'v' gives: $z = a^q (a^p)^i (a^r b^m)$

So many strings, we can get where number of a's are not equal to number of b's :— e.g. if $i = 2$, $2p + q + r \neq m$

$\therefore$ There is a contradiction!

Thus, not regular.