-By Srinkhala Maddhesia

Date ____
Page ____

Theory of Automata & Formal Language

UNIT→1

- Introduction to Theory of Computations / Automata :-

As word suggest 'TOC' is the study of 'mathematical' machines or systems called automata.

① Theory of automata is a theoretical branch of computer science & mathematical.

② It is the study of abstract machines & the computation problems that can be solved using these machines. The abstract machine is called the automata.

③ This automation consist of states & transitions. The state is represented by circles & the transition is represented by arrows.

④ Automata is the kind of machine which takes some string as input & this input goes through a finite no. of states & may enter in the final state.

⑤ These are basic terminologies that are important & frequently used in automat:

ⓐ Symbols :- Symbols are an entity or individual objects, which can be any letter, alphabet or any picture.

Ex→ 1, a, b, #

(b) **Alphabets :-** Alphabets are a finite set of symbols. It is denoted by 'Σ'.

Ex → $\Sigma = \{a, b\}$
$\Sigma = \{A, B, C\}$
$\Sigma = \{0, 1, 2\}$
$\Sigma = \{\#, \beta, \Delta\}$

(c) **String :-** It is a finite collection of symbols from the alphabet. The string is denoted by 'w'.

Ex :- $\Sigma = \{a, b\}$

Various string that can be generated from Σ are
$\{a, b, ab, ba, bb, aa, aaa, bbb, bab, aba \ldots\}$

Note :- A string with zero occurrences of symbols is known as an empty string. It is represented as ϵ.

The no. of symbols in a string 'w' is called the length of string. It is denoted by $|w|$.

Ex → $w = 010$
no. of string $|w| = 3$

(d) **Language :-** A language is a collection of appropriate string. A language which is formed over Σ can be finite or infinite.

Ex → Language (L) = { set of string of length 2 }
$\Sigma = \{a, b\}$ $\quad L = \{aa, ab, ba, ba\}$ → Finite Language

Date ____
Page ____

Ex → $L = \{$ set of all string with 'a' $\}$, $\Sigma = \{a, b\}$
$L = \{$ aa, ab, aba, aaa, abba ..... $\}$ → Infinite language

- Finite Automata :-

A F.A is a model that has a finite set of states (represented in the figure by circles) & its control move from one state to another state in response to external inputs (represented by arrow).

F.A can be broadly classified into two types :-

Finite Automata

- F.A without Output
  - → DFA
  - → NFA
  - → NFA with ε
- Finite Automata with output
  - → Moore Machine
  - → Mealy Machine

- DFA :-

→ DFA stand for Deterministic Finite Automata
→ For each symbolic representation of the alphabet, there is only one transition state.
→ DFA can't use empty string transition.
→ A DFA is a set of five tuples represented as,

$$M = \{Q, \Sigma, \delta, q_0, F\}$$

where,

$Q \rightarrow$ A non-empty finite set of states $(q_0, q_1, \dots)$

$\Sigma \rightarrow$ input alphabet $(\Sigma = \{0,1\}, \Sigma = \{a,b\} \dots)$

$\delta \rightarrow$ Transition Function $[\delta : Q \times \Sigma \rightarrow Q]$

$q_0 \rightarrow$ initial state

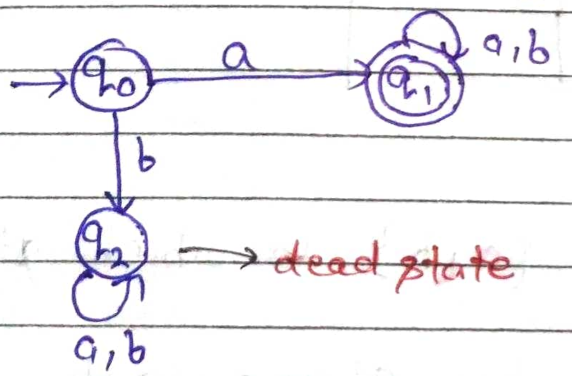
$F \rightarrow$ Final state

Example :-

① Construct a DFA, $\Sigma = \{a,b\}$ in which string start from 'a'.

$\rightarrow$ $L = \{a, ab, aa, abab, aaa, aaab, \dots\}$

Draw Transition Diagram :-



no. of states = 3

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

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

$q_0 = \{q_0\}$

$F = \{q_1\}$

Transition Function δ as shown by the following table:-

Present State	Input alphabet	
	a	b
$\rightarrow q_0$	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_1\}$	$\{q_1\}$
q_2	$\{q_2\}$	$\{q_2\}$

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

$\begin{matrix} q_0 \\ q_1 \\ q_2 \end{matrix} \times \begin{matrix} a \\ b \end{matrix} \rightarrow$

$\delta(q_0, a) \rightarrow q_1$

$\delta(q_0, b) \rightarrow q_2$

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

$\delta(q_1, b) \rightarrow q_1$

$\delta(q_2, a) \rightarrow q_2$

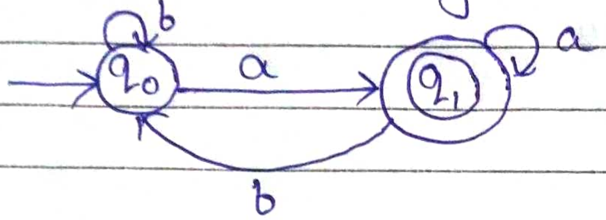
$\delta(q_2, b) \rightarrow q_2$

Date ______
Page ______

(2) Construct a DFA, $\Sigma = \{a, b\}$ in which string end with 'a'.

→ $L = \{a, ba, aa, baba, aaba, abbba \ldots\}$

Draw Transition Diagram :-



Draw Transition Table :-

Present state	input alphabet a	b
$\rightarrow q_0$	$\{q_1\}$	$\{q_0\}$
q_1	$\{q_1\}$	$\{q_0\}$

$Q \rightarrow \{q_0, q_1\}$

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

$q_0 \rightarrow \{q_0\}$

$F \rightarrow \{q_1\}$

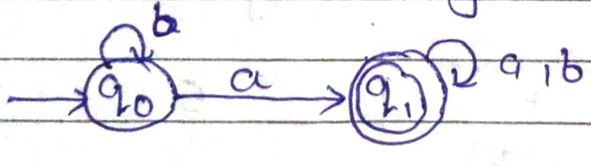
$\delta : Q \times \Sigma \rightarrow Q$

$\begin{matrix} q_0 \\ q_1 \end{matrix} \times \begin{matrix} a \\ b \end{matrix} \rightarrow$
$(q_0, a) \rightarrow q_1$
$(q_0, b) \rightarrow q_0$
$(q_1, a) \rightarrow q_1$
$(q_1, b) \rightarrow q_0$

(3) Construct a DFA, $\Sigma = \{a, b\}$ in which string that contains 'a'.

→ $L = \{a, ba, ab, aba, abaab, aab \ldots\}$

Draw Transition Diagram :-

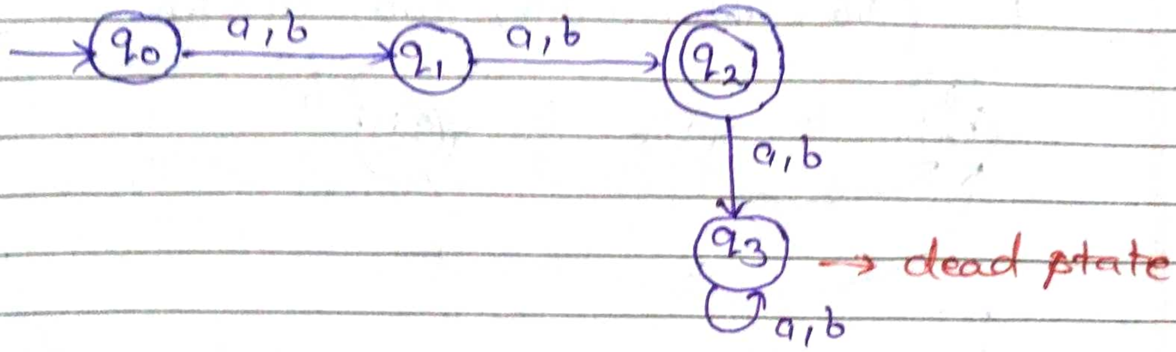


(4) Construct a DFA, $\Sigma = \{a, b\}$ & length of strings

(a) Exact 2

→ $L = \{aa, ab, ba, bb\}$

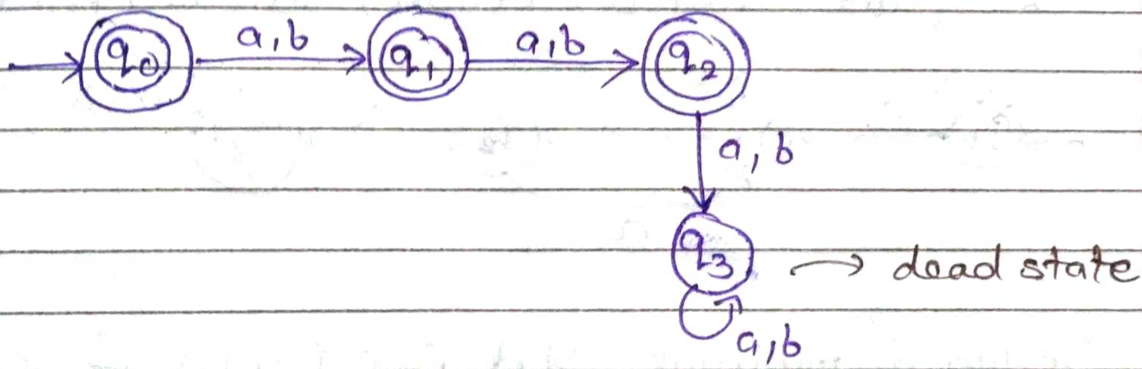
Draw Transition Diagram :-



(b) atmost 2 :-

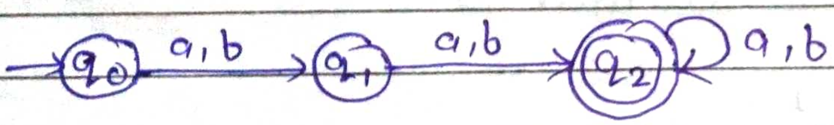
means (0, 1, 2)

L = { ε, a, b, aa, ab, ba, bb }



(c) atleast 2 :-

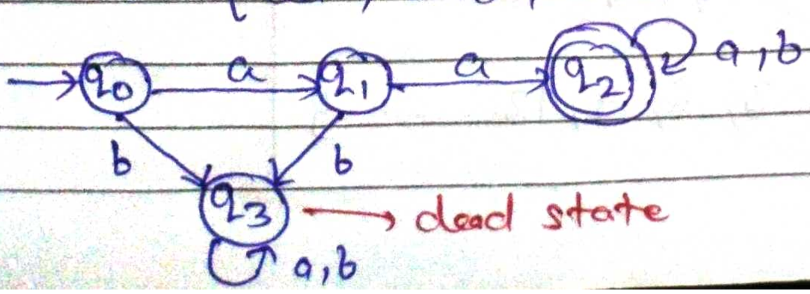
L = { aa, ab, ba, bb, aba, aab, baba, abba --- }



(5) Construct a DFA, Σ = {a, b} & length of string

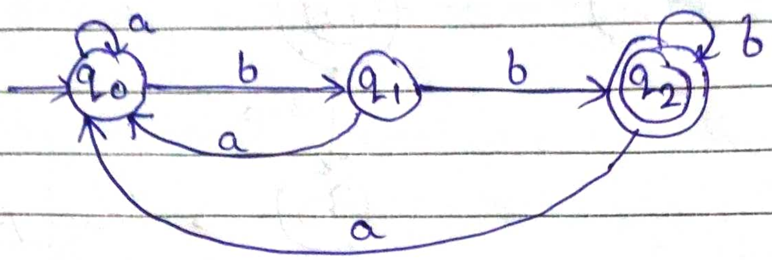
(a) start with aa

→ L = { aa, aab, aaa, aabab ---- }



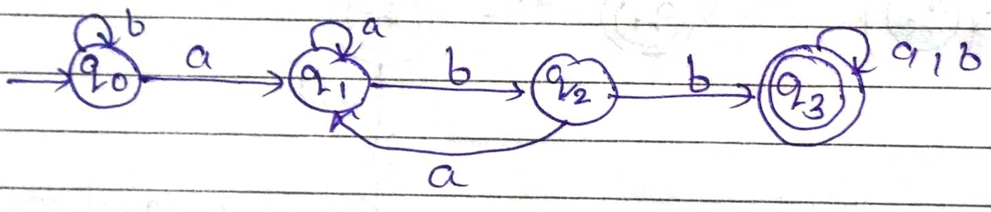
(b) End with bb

$L = \{bb, abb, aabb, bbb, ababb \ldots\}$



(6) Construct a minimal DFA, $\Sigma = \{a, b\}$ & length of strings containing 'abb' as substring

$L = \{abb, ababb, aabba, bbabb, baabba \ldots\}$

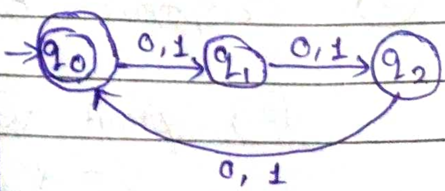


(7) Design a minimal DFA, $\Sigma = \{0, 1\}$ & length of strings

(a) $n(w) = 0 \bmod 3$

	8	4	2	1	
0 →	0	0	0	0	0/3 → 0
3 →	0	0	1	1	3/3 → 0
6 →	0	1	1	0	6/3 → 0
9 →	1	0	0	1	9/3 → 0
12 →	1	1	0	0	12/ → 0

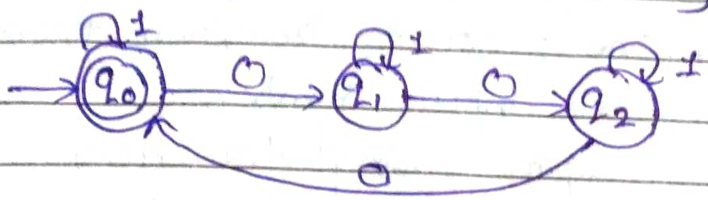
Draw Transition Diagram:-



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

(b) $n_0(w) = 0 \bmod 3$

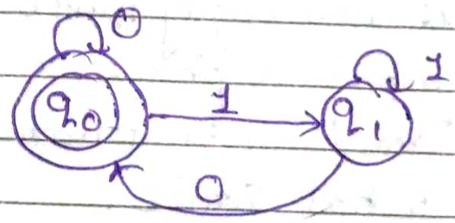
$L = \{0, 3, 6, 9 \dots\}$



(8) Construct a minimal DFA, $\Sigma = \{0, 1\}$ & length of all binary strings

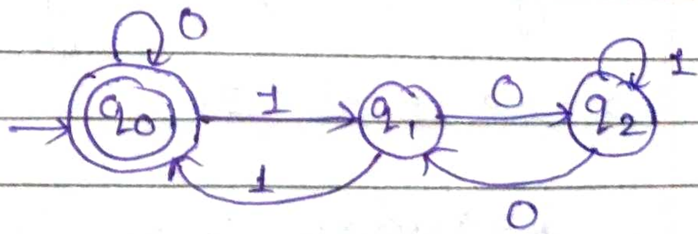
(a) divisible by 2

8 4 2 1
0 0 0 0 → 0
0 0 1 0 → 2
0 1 0 0 → 4
0 1 1 0 → 6
1 0 0 0 → 8



(b) divisible by 3

8 4 2 1
0 → 0 0 0 0
3 → 0 0 1 1
6 → 0 1 1 0
9 → 1 0 0 1
12 → 1 1 0 0



• NFA :-

→ NFA stand for Non-Deterministic Finite Automata
→ A finite automata is said to be NFA, if there is more than one possible transition from one state on the same input symbol.
→ NFA can use empty string transition
→ A NFA is also a set of 5 tuples & represented as,

$$M = \{Q, \Sigma, \delta, q_0, F\}$$

where,

Q → A non-empty finite set of states $(q_0, q_1, q_2 \cdots)$
Σ → input alphabet
δ → Transition Function $[\delta : Q \times \Sigma \rightarrow 2^Q]$
q_0 → initial state
F → Final state

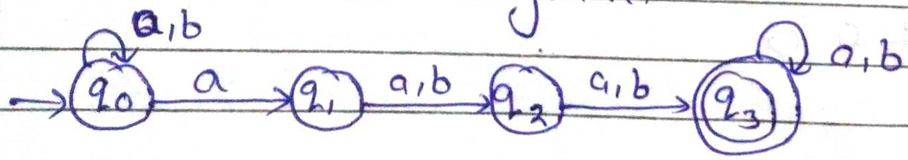
Example :-

① Construct a NFA for the language 'L' which accept all the strings in which the 3rd symbol from right end is always a over $\Sigma = \{a, b\}$

→ $\underline{a,b}$ $\underline{a}$ $\underline{a,b}$ $\underline{a,b}$

$L = \{aaaa, baaa, aabb, babb, aaabb, \cdots\}$

Draw Transition diagram :-



$Q \rightarrow \{q_0, q_1, q_2, q_3\}$	$q_0 \rightarrow \{q_0\}$
$\Sigma \rightarrow \{a, b\}$	$F \rightarrow \{q_3\}$

Date ______
Page ______

$\delta : Q \times \Sigma \rightarrow 2^Q$

$\begin{matrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{matrix} \times \begin{matrix} a \\ b \end{matrix} \rightarrow$

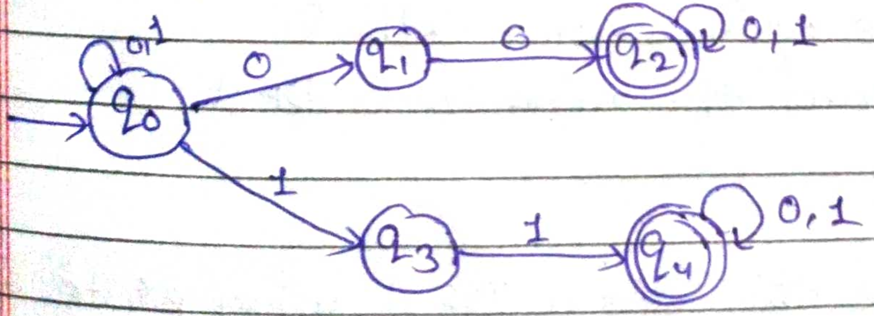
$\delta(q_0, a) \rightarrow \{q_0, q_1\}$
$\delta(q_0, b) \rightarrow \{q_0\}$
$\delta(q_1, a) \rightarrow \{q_2\}$
$\delta(q_1, b) \rightarrow \{q_2\}$
$\delta(q_2, a) \rightarrow \{q_3\}$
$\delta(q_2, b) \rightarrow \{q_3\}$
$\delta(q_3, a) \rightarrow \{q_3\}$
$\delta(q_3, b) \rightarrow \{q_3\}$

Transition Table :-

State	Input alphabet	
	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	$\{q_3\}$	$\{q_3\}$

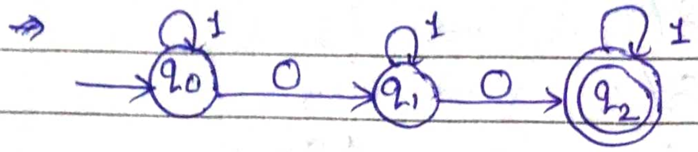
(2) Design a NFA for the language 'L' which accepts all string over (0,1) that have atleast two consecutive 0's or 1's.

$\rightarrow$ $L = \{00, 11, 000, 111, 110, 001, \ldots\}$

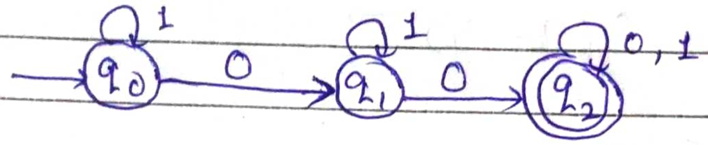


③ Construct a NFA for the language 'L' which $\Sigma = \{0, 1\}$ & no. of 0's in a string is

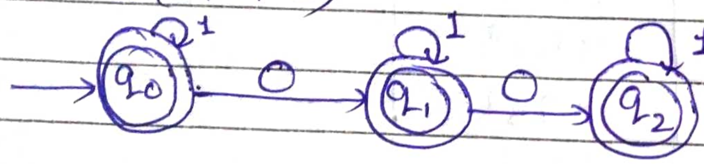
ⓐ Exact 2



ⓑ Atleast 2



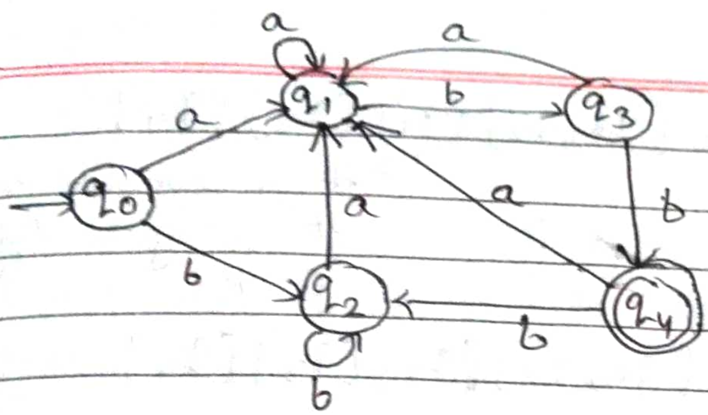
ⓒ Atmost 2
means (0, 1, 2)



- Minimization of DFA (State Equivalence Method or Partition Method) :-

Rule of minimization of DFA :-
① Delete unreachable states
② Draw state Transition Table
③ Apply state equivalence Method
④ Now, construct the minimized DFA

Example :-
① Minimize the given DFA using state equivalence method / partition method



→ Steps :-

① Do not have any unreachable states

② Draw state Transition Table :-

state	a	b
$\rightarrow q_0$	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_1\}$	$\{q_3\}$
q_2	$\{q_1\}$	$\{q_2\}$
q_3	$\{q_1\}$	$\{q_4\}$
q_4	$\{q_1\}$	$\{q_2\}$

③ Apply state equivalence Method :-

Until the two state equivalences become the same, we will continue to create state equivalence & seperate the final & non-final states.

For 0-equivalence = $[\underbrace{q_0, q_1, q_2, q_3}_{\text{Non-Final state}}]\ [\underbrace{q_4}_{\text{Final state}}]$

For 1-equivalence = $[q_0, q_1, q_2][q_3]\ [q_4]$

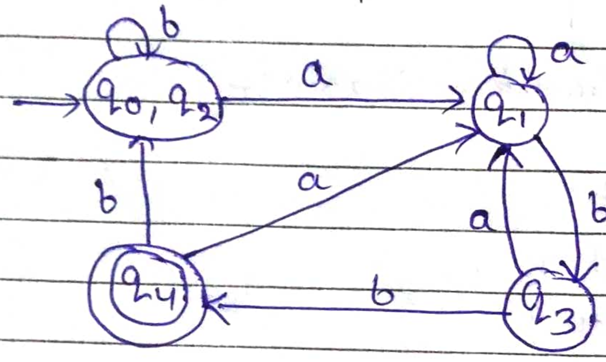
↳ it is seperated because it belong to different group

For 2- equivalence = $[q_0, q_2]\ [q_1]\ [q_3]\ [q_4]$

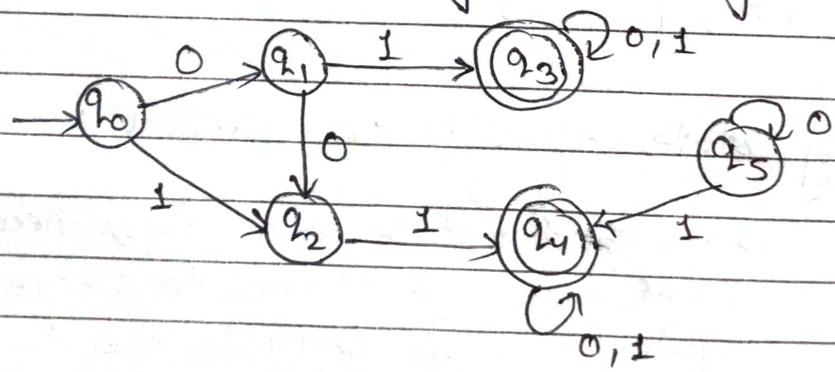
↳ it is seperated because it belong to different group

For 3- equivalence = $[q_0, q_2]\ [q_1]\ [q_3]\ [q_4]$

(4) Now, construct the minimized DFA



(2) Minimize the no. of states of the following DFA.



→ Steps :-

(1) 'q_5' is a unreachable state

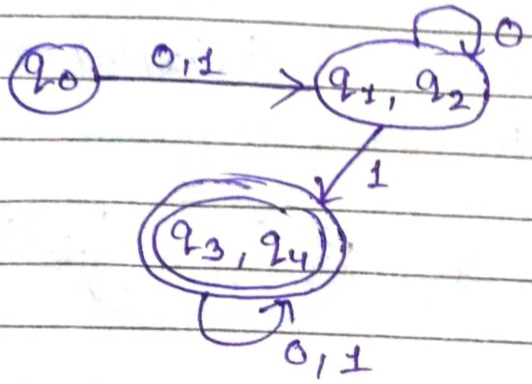
(2) Draw State Transition Table

State	0	1
→ q_0	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_2\}$	$\{q_3\}$
q_2	$\{q_2\}$	$\{q_4\}$
q_3	$\{q_3\}$	$\{q_3\}$
q_4	$\{q_4\}$	$\{q_4\}$

(3) Applying state equivalence Method -

For 0-equivalence = $[q_0, q_1, q_2]\ [q_3, q_4]$
For 1-equivalence = $[q_1, q_2]\ [q_0]\ [q_3, q_4]$
For 2-equivalence = $[q_0]\ [q_1, q_2]\ [q_3, q_4]$

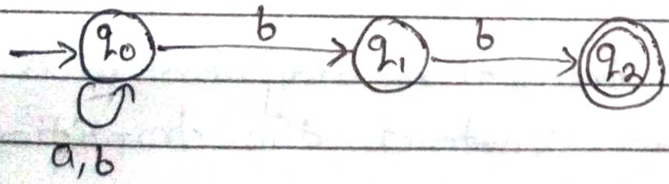
(4) Now, construct the minimized DFA,



- Conversion of NFA to DFA / Conversion of NFA to equivalent-DFA :-

Example :-

(1) Convert the NFA for the following DFA



→ Step :-

(1) For the given transition diagram we will first construct the transition table.

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

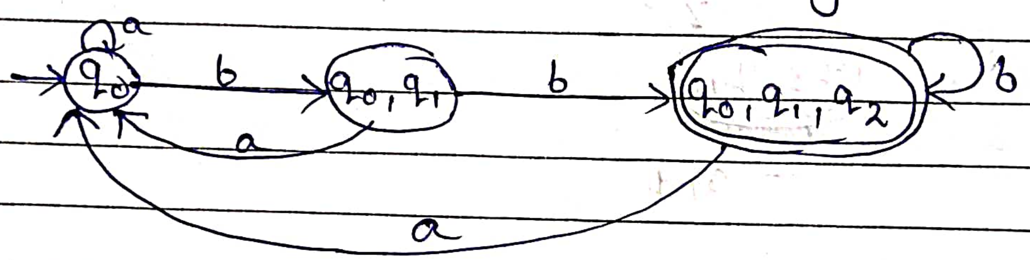
Date
Page

② The transition table for the constructed DFA will be:

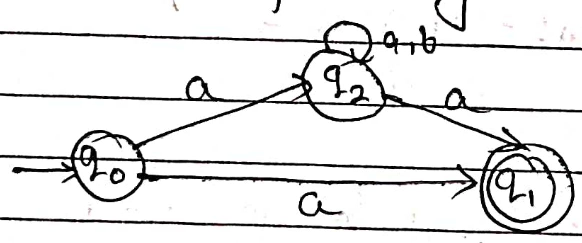
State	a	b
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_0, q_1	q_0	$\{q_0, q_1, q_2\}$
q_0, q_1, q_2	q_0	$\{q_0, q_1, q_2\}$

~~$\delta : Q \times \Sigma \rightarrow Q$~~
~~$\delta(q_0, a) \rightarrow q_0$~~
~~$\delta(q_0, b) \rightarrow q_0, q_1$~~
~~$\delta(q_1, a) \rightarrow \phi$~~
~~$\delta(q_1, b) \rightarrow q_2$~~
~~$\delta(q_2, a) \rightarrow \phi$~~
~~$\delta(q_2, b) \rightarrow \phi$~~

③ Now, construct the transition diagram will be



② Convert the following NFA to DFA



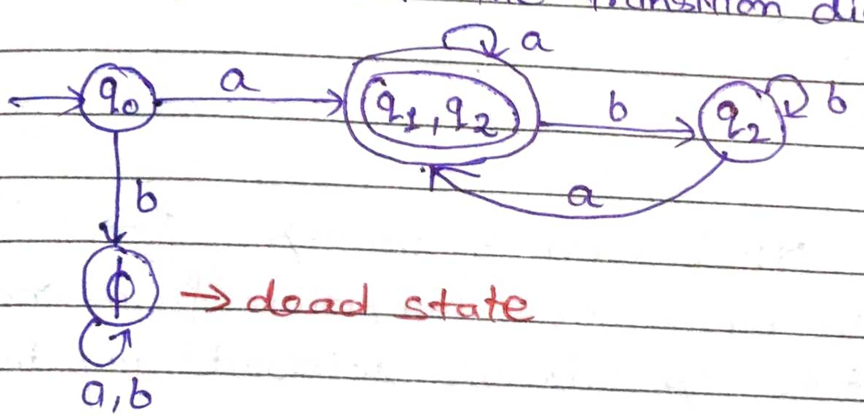
→ ① For the given transition diagram will be first conver the construct the transition table

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

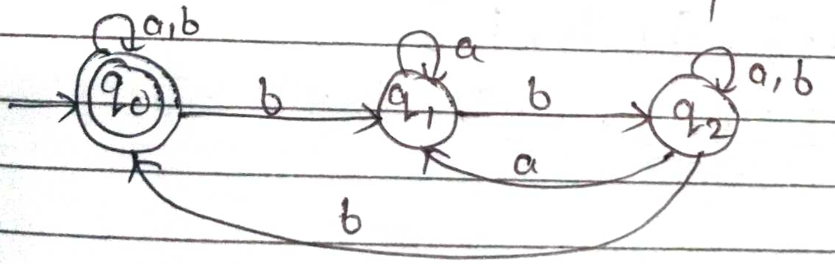
② The transition table for the constructed DFA will be

state	a	b
$\rightarrow q_0$	$\{q_1, q_2\}$	ϕ
(q_1, q_2)	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\{q_1, q_2\}$	$\{q_2\}$
ϕ	ϕ	ϕ

③ Now, construct the transition diagram will be



③ Construct a minimal DFA for the following NFA



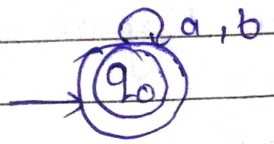
→ ① Construct the transition table -

State	a	b
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_1\}$	$\{q_2\}$
q_2	$\{q_1, q_2\}$	$\{q_0, q_2\}$

(2) The transition table for the constructed DFA will be

State	a	b
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_0, q_1	$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$
q_0, q_1, q_2	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

(3) Now, construct the transition diagram will be



- **ϵ-NFA / NFA with ϵ moves :-**

The NFA in which the transition from one state to another state is allowed without any input symbol ie, empty string 'ϵ'.

ϵ-NFA has 5 tuple -

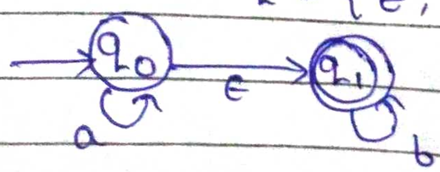
$$M = \{Q, \Sigma, \delta, q_0, F\}$$

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

Example :-

(1) $L = a^n b^m \mid n, m \geq 0$

$n = 0, 1, 2, \ldots$
$m = 0, 1, 2, \ldots$
$L = \{\epsilon, ab, abbb, aabbbb, \ldots\}$

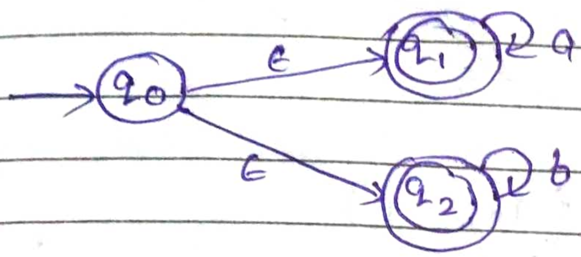


Date
Page

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

$n = 0, 1, 2, \ldots$

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



- **ε - closure :-**

If 'q' is any state in ε-NFA then set of all the states which are at '0' distance from state 'q' is known as ε-closure of 'q'.

Note :-

① Every state is zero distance from itself

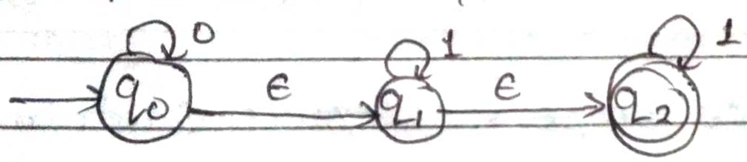
② $\epsilon\text{-closure}(q) \neq \phi$

③ $\epsilon\text{-closure}(\phi) = \phi$

④ $\epsilon\text{-closure}(q_0, q_1, q_2, q_3 \ldots q_n) = \bigcup_{i=0}^{n} \epsilon\text{-closure}(q_0, q_1, q_2 \ldots q_n)$

Example :-

① Find ε - closure



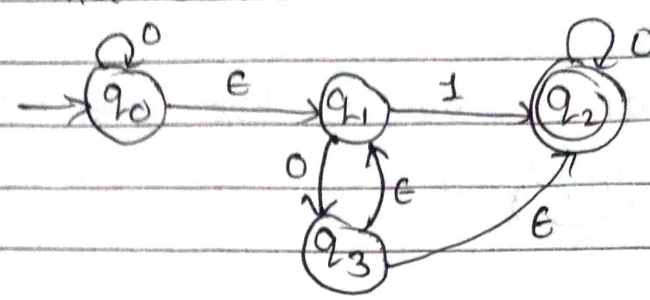
→ $\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

$\epsilon\text{-closure}(q_1) = \{q_1, q_2\}$

$\epsilon\text{-closure}(q_2) = \{q_2\}$

Date
Page

(2) Find ϵ-closure



→ ϵ-closure $(q_0) = \{q_0, q_1\}$
ϵ-closure $(q_1) = \{q_1\}$
ϵ-closure $(q_2) = \{q_2\}$
ϵ-closure $(q_3) = \{q_1, q_2, q_3\}$

- Conversion of ϵ-NFA to NFA :-

Algorithm :-
Let $M = \{Q, \Sigma, \delta, q_0, F\} \rightarrow \epsilon$-NFA
$M' = \{Q', \Sigma', \delta', q_0', F'\} \rightarrow$ NFA

(1) Initial state $(q_0) = q_0'$
(2) Construction of δ'

$$\boxed{\delta'(q, x) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}\{q\}, x))}$$

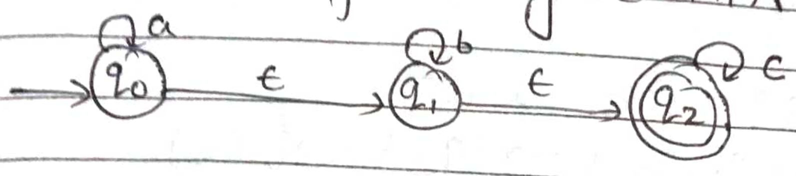
(3) Every state whose ϵ-closure contain final state of ϵ-NFA is final state in NFA

Note :- (1) No change in initial state
(2) No change the total no. of states
(3) May be change in the final no. of states.

Date
Page

Example :-

① Convert the following ϵ-NFA to NFA



→ ① Transition Table of ϵ-NFA :-

δ	a	b	c	ϵ
$\rightarrow q_0$	$\{q_0\}$	ϕ	ϕ	$\{q_1\}$
q_1	ϕ	$\{q_1\}$	ϕ	$\{q_2\}$
q_2	ϕ	ϕ	$\{q_2\}$	ϕ

② Find ϵ-closure of each states

ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$
ϵ-closure $(q_1) = \{q_1, q_2\}$
ϵ-closure $(q_2) = \{q_2\}$

③ Find transition function -

$\delta'(q_0, a) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}\{q_0\}, x))$
$= \epsilon\text{-closure}(\delta(q_0, q_1, q_2, a))$
$= \epsilon\text{-closure}(q_0)$
$= \{q_0, q_1, q_2\}$

OR

$\delta'(q_0, a) = \epsilon^* \longrightarrow a \longrightarrow \epsilon^*$
$= \{q_0, q_1, q_2\} \longrightarrow \{q_0\} \longrightarrow \{q_0, q_1, q_2\}$

$\delta'(q_0, b) = \epsilon^* \rightarrow b \rightarrow \epsilon^*$
$= \{q_0, q_1, q_2\} \rightarrow \{q_1\} \rightarrow \{q_1, q_2\}$

$\delta'(q_0, c) = \epsilon^* \rightarrow c \rightarrow \epsilon^*$
$= \{q_0, q_1, q_2\} \rightarrow \{q_2\} \rightarrow \{q_2\}$

$\delta'(q_1, a) = \epsilon^* \rightarrow a \rightarrow \epsilon^*$
$= q_1, q_2 \rightarrow \phi \rightarrow \phi$

$\delta'(q_1, b) = \epsilon^* \rightarrow b \rightarrow \epsilon^*$
$= q_1, q_2 \rightarrow q_1 \rightarrow q_1, q_2$

$\delta'(q_1, c) = \epsilon^* \rightarrow c \rightarrow \epsilon^*$
$= q_1, q_2 \rightarrow q_2 \rightarrow q_2$

$\delta'(q_2, a) = \epsilon^* \rightarrow a \rightarrow \epsilon^*$
$= q_2 \rightarrow \phi \rightarrow \phi$

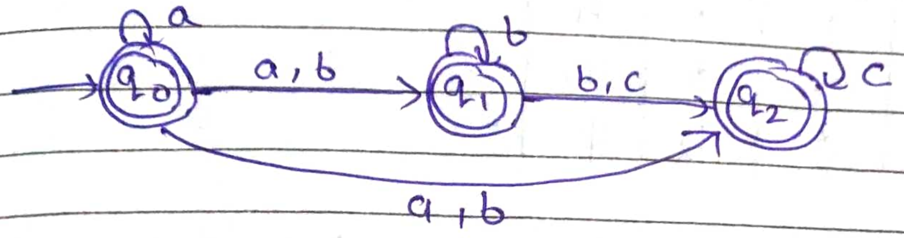
$\delta'(q_2, b) = \epsilon^* \rightarrow b \rightarrow \epsilon^*$
$= q_2 \rightarrow \phi \rightarrow \phi$

$\delta'(q_2, c) = \epsilon^* \rightarrow c \rightarrow \epsilon^*$
$= q_2 \rightarrow q_2 \rightarrow q_2$

Construct the transition table of NFA will be

δ'	a	b	c
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	ϕ	$\{q_1, q_2\}$	$\{q_2\}$
q_2	ϕ	ϕ	$\{q_2\}$

(4) Now, construct the transition diagram of NFA



- **Conversion of ε-NFA to DFA :-**

Algorithm :-

Let $M = \{Q, \Sigma, \delta, q_0, F\} \rightarrow$ ε-NFA

$M' = \{Q', \Sigma, \delta', q_0', F'\} \rightarrow$ DFA

(1) For initial state

$q_0' = \epsilon\text{-closure}(q_0)$

(2) Construction of δ'

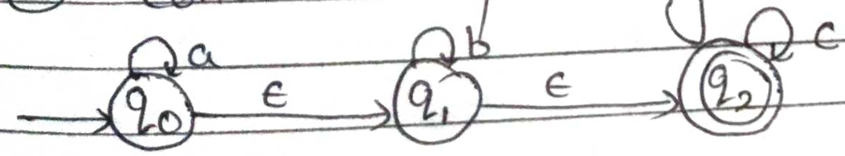
$$\boxed{\delta'(q, x) = \epsilon\text{-closure}(\delta(q, x))}$$

Start the construction of δ' with initial state & continue for every new state

(3) Final state (Every subset which contain final state of ε-NFA is the final state of DFA

Example :-

(1) Convert the following ε-NFA to DFA



→ (1) Draw Transition Table of ε-NFA

Date ______
Page ______

δ	a	b	c	ϵ
$\rightarrow q_0$	q_0	ϕ	ϕ	q_1
q_1	ϕ	q_1	ϕ	q_2
q_2	ϕ	ϕ	q_2	ϕ

(2) Find ϵ-closure of each states

~~ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$~~

~~ϵ-closure $(q_1) = \{q_1, q_2\}$~~

ϵ-closure $(q_2) = \{q_2\}$

(3) Find transition function

~~$\delta'(q_0, q_1, q_2\}$~~

$\delta'((q_0, q_1, q_2), a) = \epsilon\text{-closure}(\delta\{q, x\})$
$= \epsilon\text{-closure}(\delta\{q_0, q_1, q_2, a\})$
$= \epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

$\delta'((q_0, q_1, q_2), b) = \epsilon\text{-closure}(\delta(q_0, q_1, q_2, b))$
$= \epsilon\text{-closure}(\delta q_1) = \{q_1, q_2\}$

$\delta'((q_0, q_1, q_2), c) = \epsilon\text{-closure}(\delta(q_0, q_1, q_2, c))$
$= \epsilon\text{-closure}(q_2) = \{q_2\}$

$\delta'((q_1, q_2), a) = \epsilon\text{-closure}(\delta(q_1, q_2, a))$
$= \epsilon\text{-closure}(\phi) = \phi$

$\delta'((q_1, q_2), b) = \epsilon\text{-closure}(\delta(q_1, q_2, b))$
$= \epsilon\text{-closure}(q_1) = \{q_1, q_2\}$

$\delta'((q_1, q_2), c) = \epsilon\text{-closure}(\delta(q_1, q_2, c))$
$= \epsilon\text{-closure}(q_2) = \{q_2\}$

$\delta'(q_2, a) = \epsilon\text{-closure}(\delta(q_2, a))$
$= \epsilon\text{-closure}(\phi) = \phi$

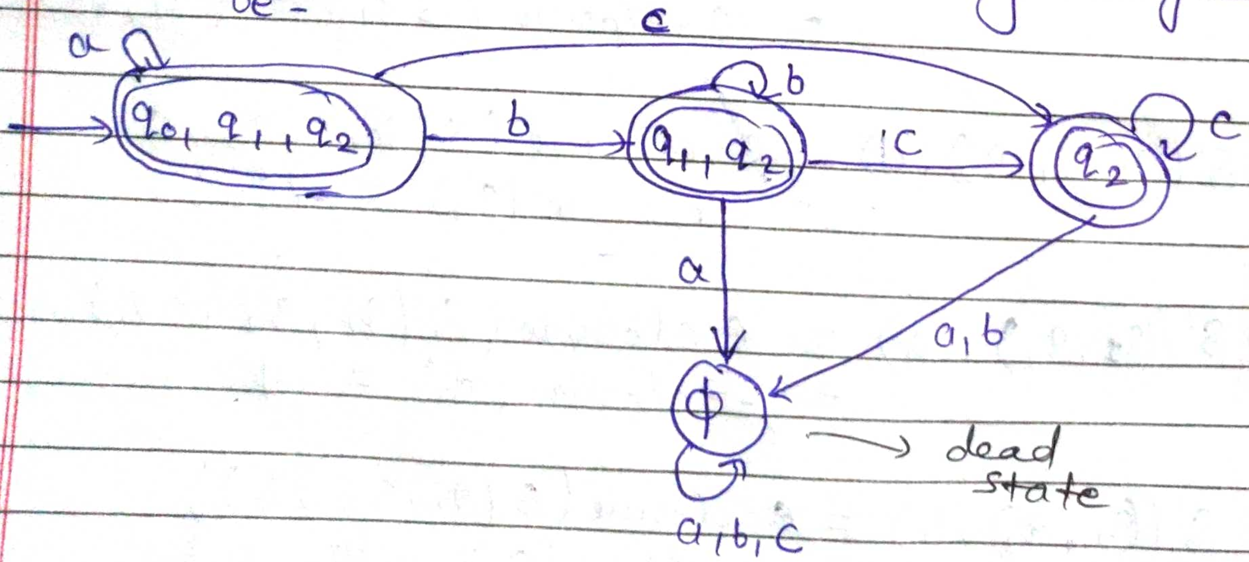
$\delta'(q_2, b) = \epsilon\text{-closure}(\delta(q_2, b))$
$= \epsilon\text{-closure}(\phi) = \phi$

$\delta'(q_2, c) = \epsilon\text{-closure}(\delta(q_2, c))$
$= \epsilon\text{-closure}(q_2) = \{q_2\}$

Construct the transition table of DFA will be

δ'	a	b	c
$\rightarrow$ q_0, q_1, q_2	q_0, q_1, q_2	q_1, q_2	q_2
q_1, q_2	ϕ	q_1, q_2	q_2
q_2	ϕ	ϕ	q_2
ϕ	ϕ	ϕ	ϕ

(4) Now, construct the transition diagram of DFA will be -



Date
Page

Finite Automata with output Mealy & Moore Machine :-

* Moore Machine :-

M.M is a finite state machine in which the next state is decided by the current state & current input symbol.

In M.M, output depend on present state only.

Moore Machine can be described by 6 tuples :- $(Q, q_0, \Sigma, \Delta, \delta, \lambda)$ where,

$Q \rightarrow$ finite set of states
$\Sigma \rightarrow$ finite set of input symbol (input alphabet)
$\Delta \rightarrow$ output alphabet
$\delta \rightarrow$ Transition Function $[\delta : Q \times \Sigma \rightarrow Q]$
$\lambda \rightarrow$ output transition Function $[\lambda : Q \rightarrow \Delta]$
$q_0 \rightarrow$ Initial state

Example :- Design a Moore Machine

Q)

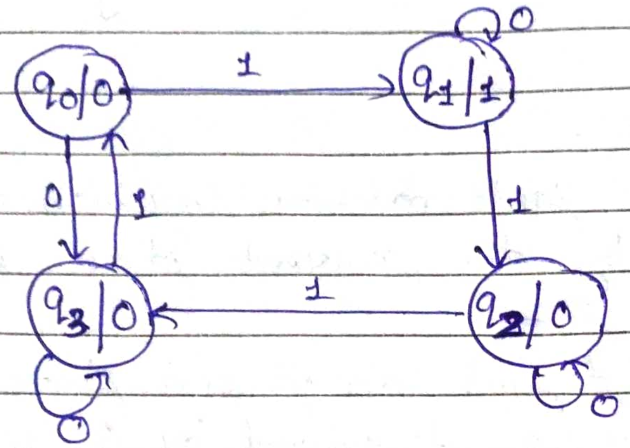
Present state (Q)	Next state 0	Next state 1	Output λ
$\rightarrow q_0$	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

$\rightarrow$

$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{0, 1\}$
$\Delta = \{0, 1\}$
$q_0 = q_0$

$\lambda(q_0) \rightarrow 0$
$\lambda(q_1) \rightarrow 1$
$\lambda(q_2) \rightarrow 0$
$\lambda(q_3) \rightarrow 0$

Now, construct the transition diagram will be



* Mealy Machine :-

A Mealy Machine is a machine in which output symbol depends upon the present input symbol & present state of the machine.

A.MM can be described as 6 tuple $(Q, \Sigma, \Delta, \delta, \lambda', q_0)$

where,

$Q \rightarrow$ finite set of states

$\Sigma \rightarrow$ ~~initial state~~ input alphabet

$\Delta \rightarrow$ output alphabet

$\delta \rightarrow$ Transition Function $[\delta : \Sigma \times Q \rightarrow Q]$

$\lambda' \rightarrow$ output transition Function $[\lambda : Q \times \Sigma \rightarrow \Delta]$

$q_0 \rightarrow$ initial state

Ex :- (1) Design Mealy Machine

Present state (Q)	Input alphabet 0		1	
	State	Output	State	Output
$\rightarrow q_1$	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

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

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

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

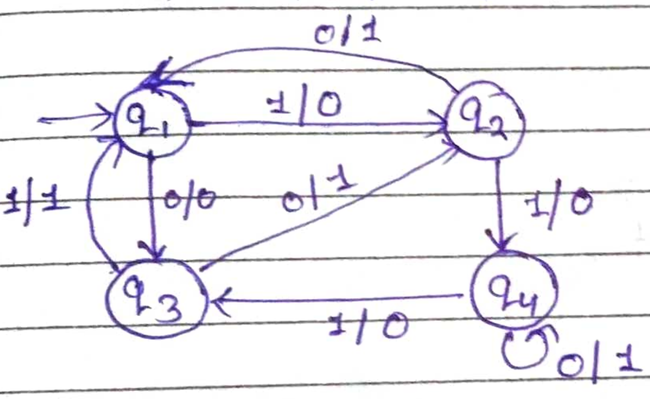
$\lambda'(q_1, 0) \to 0$ | $\lambda'(q_2, 0) \to 1$ | $\lambda'(q_3, 0) \to 1$

$\lambda'(q_1, 1) \to 0$ | $\lambda'(q_2, 1) \to 0$ | $\lambda'(q_3, 1) \to 1$

$\lambda'(q_4, 0) \to 1$

$\lambda'(q_4, 1) \to 0$

Now, construct the transition diagram will be



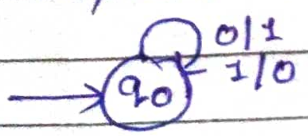
(2) Design Mealy Machine which print 1's complement of input bit string over alphabet $\Sigma = \{0, 1\}$

→ Input = binary string

Input = 10011000

1's complement = 01100111

Construct the transition diagram will be



$Q = \{q_0\}$

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

$q_0 = q_0$

$\lambda'(q_0, 0) \to 1$

$\lambda'(q_0, 1) \to 0$

③ Design Mealy Machine which print 2's complement of input bit string over alphabet $\Sigma = \{0, 1\}$

→ Input = binary string

Input = 1 1 1 0 0

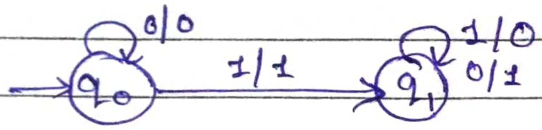
1's complement = 0 0 0 1 1

$$
\begin{array}{r}
00011 \\
+\ 1 \\
\hline
00100
\end{array}
$$

2's complement = 0 0 1 0 0

Input = 1 1 $\boxed{1}$ 0 0

Output = 0 0 $\boxed{1}$ 0 0



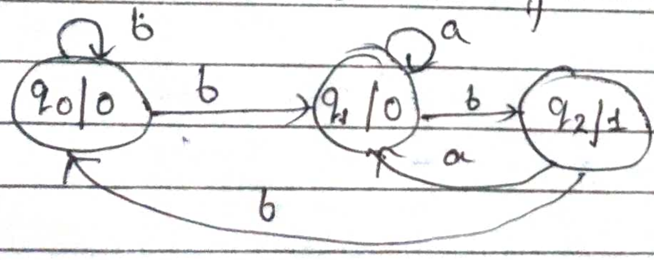
- Conversion from Moore to Mealy Machine :–

① If the transition diagram is given then convert the transition diagram to trasition table.

② Then, in the 2nd step conver the transition table of Moore Machine to Transition diagram of Mealy Machine where the output will be associated with the transition

Note → no. of states always same

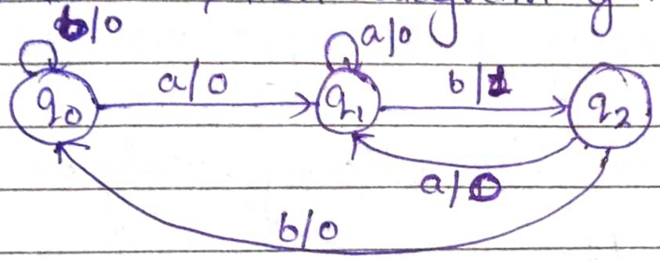
Example :– ① Convert the following moore to Mealy Machine



→ Construct the transition table of Moore Machine -

Present state	Next State a	Next State b	Output (Δ)
→ q_0	q_1	q_0	0
q_1	q_1	q_2	0
q_2	q_1	q_0	1

Draw transition diagram of Mealy Machine -



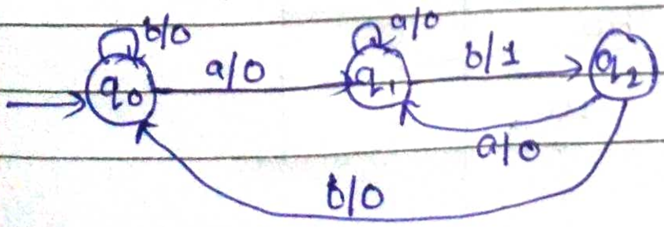
② Convert the following Moore to Mealy Machine

Present State	Next State a	Next State b	output (Δ)
→ q_0	q_1	q_0	0
q_1	q_1	q_2	0
q_2	q_1	q_0	1

→ Draw Transition Table of Mealy Machine

Present State	Next State a: State	a: output	b: State	b: Output
→ q_0	q_1	0	q_0	0
q_1	q_1	0	q_2	1
q_2	q_1	0	q_0	0

→ Draw Transition diagram of Mealy Machine



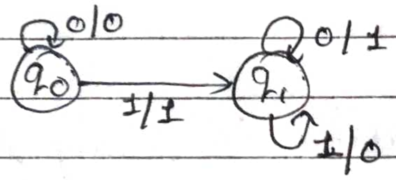
Conversion from Mealy to Moore Machine :-

① If a transition diagram is given then convert the transition diagram into transition table

② From the transition table of Mealy Machine, find out such kind of states which are showing more than one output.

③ Then, in such cases, divide those states into more states depending on the output associated with them.

Example :- ① Convert the following Mealy to Moore Machine.



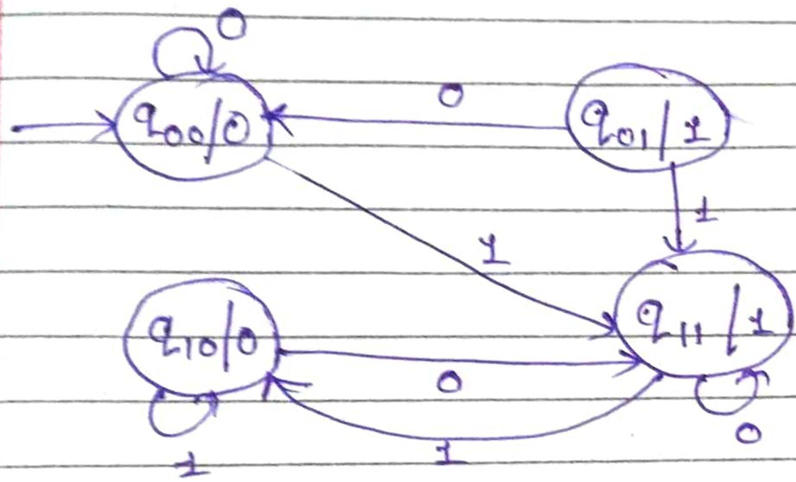
→ ① Construct the Transition ~~diagram~~ table of Mealy Machine

Present State	Next state			
	0	Δ	1	Δ
→ q_0	q_0	0	q_1	1
q_1	q_1	1	q_1	0

② Now, construct the transition ~~diagram~~ table of Moore Machine

Present state	Input = 0 Next state	Input = 1 Next state	Output
q_{00}	q_{00}	q_{11}	0
q_{01}	q_{00}	q_{11}	1
q_{10}	q_{11}	q_{10}	0
q_{11}	q_{11}	q_{10}	1

(3) Draw the transition diagram of Moore Machine



Prepared By — Srinkhala Maddhesia