UNIT-II

23/12/21
Thursday

DFA AND NFA
– Finite Automata

Overview

- DFA Vs NFA/NDFA
- NFA with ε transitions
- Equivalence between NFA with and without ε transitions
- NFA to DFA Conversion
- minimization of FSM
- Moore and Melay Machines.

DFA Vs NDFA

The following table lists the differences between DFA and NDFA.

DFA	NDFA
1. The transition from a state is to a single particular next state for each input symbol. Hence it is called deterministic.	1. The transition from a state can be to multiple next states for each input symbol. Hence it is called non-deterministic.
2. Empty string transitions are not seen in DFA.	2. NDFA permits empty string transitions.
3. Backtracking is allowed in DFA.	3. In NDFA, backtracking is not always possible.
4. Requires more space	4. Requires less space
5. A string is accepted by a DFA, if it transits to a final state.	5. A string is accepted by a NDFA, if at least one of all possible transitions ends in a final state.

NFA with ε Transitions.

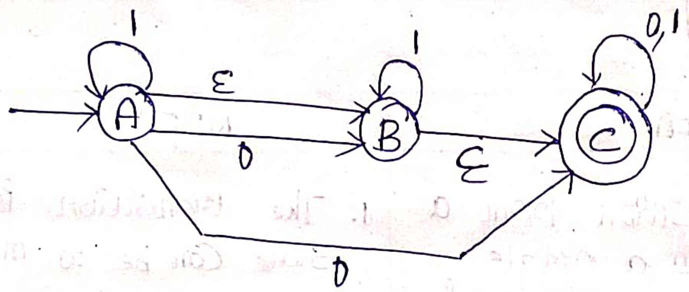
The Class of NFAs by allowing instantaneous (ε) transitions:

⇒ The automaton may be allowed to change its state without reading the input symbol.

⇒ In diagrams, such transitions are depicted by labeling the appropriate arcs with ε.

⇒ Every transition with ε move can have equivalent NFA without ε-moves also.

⇒ Note that this does not mean that ε has become an input symbol. On the contrary, we assume that the symbol ε does not belong to any alphabet.



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

$Q = \{A, B, C\}$

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

$q_0 = A$

$F = \{C\}$

δ	0	1	ε
A	B,C	A	B
B	-	B	C
C	C	C	-

ε is accepted by the automata because as there is ε move from A to B and also B to C, without any input symbol, we can reach the final state.

W = {ε, 0} ; Accepted by the Automata.

W = 1 ; Accepted by the Automata.

$\because A \xrightarrow{\varepsilon} B \xrightarrow{\varepsilon} C \xrightarrow{1} C$

$L = \{(0,1)^n \mid n \geq 0\}$ (or)

$L = \{(0,1)^* \mid n \geq 0\}$

Formal definition of ε-NFA

- A ε-NFA is a quintuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a set of states

Σ is the alphabet of input symbols

$q_0 \in Q$ is the initial state

$F \subseteq Q$ is the set of final states

$\delta : Q \times \Sigma \longrightarrow P(Q)$ is the transition function

$$\left[\begin{array}{l} \because \delta : Q \times \Sigma \rightarrow 2^Q \\ \delta : Q \times \{\Sigma \cup \varepsilon\} \longrightarrow 2^Q \end{array} \right]$$

$$\underbrace{\{\Sigma \cup \varepsilon\}}_{\Sigma_\varepsilon}$$

- Note ε is never a member of Σ
- Σ_ε is defined to be $(\Sigma \cup \varepsilon)$

ε- NFA

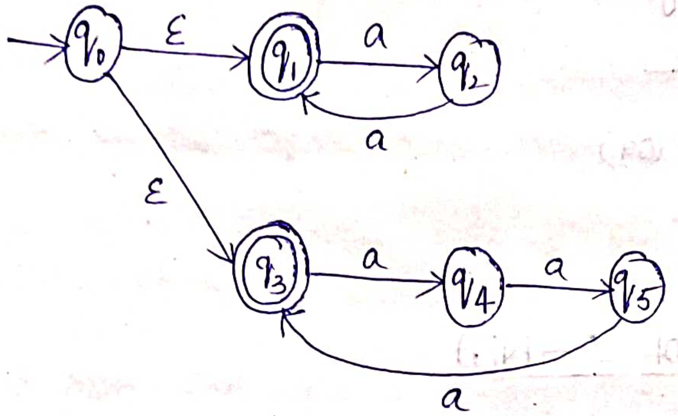
- ε-NFAs add a convenient feature but (in a sense) they bring us nothing new: they do not extend the class of language that can be represented. Both NFA's and ε-NFAs recognize exactly the same languages.

Example:-

$L = \{a^n \mid n \text{ is even or divisible by } 3\}$

$L = \{\varepsilon, aa, aaa, aaaa, aaaaaa, aaaaaaaa, aaaaaaaaa \ldots\ldots\}$

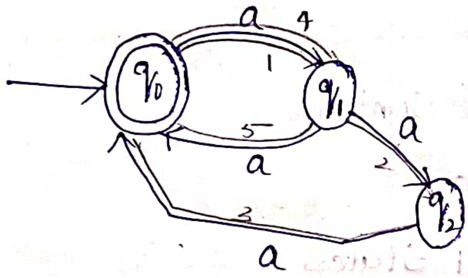
1st Method:-



ε-NFA

(i)

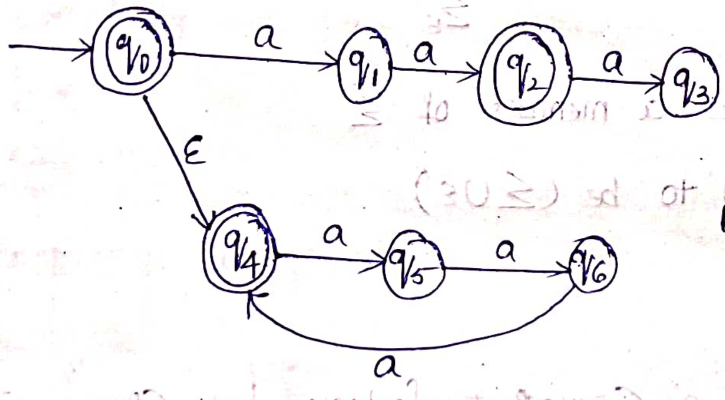
2nd Method:-



(ii)

Here (i) and (ii) are not Equivalent because in (ii) Language accepts 5 a's in our given language there is no 5 a's so it is not equivalent.

3rd Method:-



It is equivalent to (i).

(iii)

We can draw any transition diagram for a given language. i.e., we can draw any type of diagram but it is equivalent to each other.

Equivalent means, if a given language, X_1 draws one type of transition diagram and another X_2 draws one type of transition diagram, both are accepted same languages, if X_1 accepts the language then X_2 also accepts the same language. If X_2 a rejects the language then X_1 also rejects the languagen then we say it is equivalent.

Eliminating ε-Transitions

⇒ NFA with ε can be converted to NFA without ε

⇒ NFA without ε can be converted to DFA.

- To do this, we will use a method, which can remove all the ε transition from given NFA.

1) Find out all the ε-transitions from each state from Q. That will be called as ε-closure$\{q_i\}$ where, $q_i \in Q$.

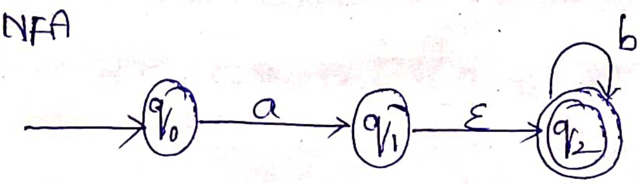
2) Then δ' transitions can be obtained. The δ' transitions mean a ε-closure on δ moves.

3) Repeat Step-2 for each input symbol and each state of given NFA.

4) Using the resultant states, the transition table for equivalent NFA without ε can be built.

Example:-

ε-NFA



This is ε-NFA

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

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

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

$q_0 = q_0$

$F = \{q_2\}$

ε-NFA:

δ	a	b	ε
$\rightarrow q_0$	q_1	-	-
q_1	-	-	q_2
$*q_2$	-	q_2	-

ε - closure $(q_0) = \{q_0\}$

ε - closure $(q_1) = \{q_1, q_2\}$

ε - closure $(q_2) = \{q_2\}$

Step-2:-

$\delta(q_0, a) = q_1$

$\delta(q_1, \varepsilon) = q_2$

$\delta(q_2, b) = q_2$

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

$\delta'(q_0, b) = \phi$

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

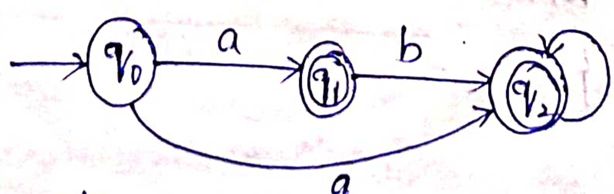
$\delta'(q_1, b) = \{q_2\}$ $\quad \varepsilon b, b\varepsilon, \varepsilon b \varepsilon$

$\delta'(q_2, a) = \phi$

$\delta'(q_2, b) = q_2$

transition Table:- NFA

δ	a	b
$\rightarrow q_0$	q_1, q_2	-
q_1	ϕ	q_2
$*q_2$	ϕ	q_2



THIS IS NFA

For this General procedure:-

$\delta'(q_0, a) = \varepsilon\text{-closure}(\delta(\delta^*(q_0, \varepsilon), a))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), a))$

$= \varepsilon\text{-closure}(\delta(q_0, a))$

$= \varepsilon\text{-closure}(q_1)$

$= \{q_1, q_2\}$

$\delta'(q_0, b) = \varepsilon\text{-closure}(\delta(\delta^*(q_0, \varepsilon), b))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), b))$

$= \varepsilon\text{-closure}(\delta(q_0, b))$ [$\because$ q_0, b there is no move i.e., $\emptyset$]

$= \varepsilon\text{-closure}(\emptyset)$

$= \emptyset$ [$\because$ $\varepsilon\text{-closure}(\emptyset) = \emptyset$]

$\delta'(q_1, a) = \varepsilon\text{-closure}(\delta(\delta^*(q_1, \varepsilon), a))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1), a))$

$= \varepsilon\text{-closure}(\delta(\{q_1, q_2\}, a))$

$= \varepsilon\text{-closure}(\delta(q_1, a) \cup \delta(q_2, a))$

$= \varepsilon\text{-closure}(\emptyset \cup \emptyset)$

$= \emptyset$

$\delta'(q_1, b) = \varepsilon\text{-closure}(\delta(\delta^*(q_1, \varepsilon), b))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1), b))$

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

$= \varepsilon\text{-closure}(\delta(q_1, b) \cup \delta(q_2, b))$

$= \varepsilon\text{-closure}(\emptyset \cup q_2)$

$= \{q_2\}$

$$\delta'(q_2,a) = \varepsilon\text{-closure}(\delta(\delta^*(q_2,\varepsilon),a))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2),a))$$
$$= \varepsilon\text{-closure}(\delta(q_2,a))$$
$$= \varepsilon\text{-closure}(\emptyset)$$
$$= \emptyset$$

$$\delta'(q_2,b) = \varepsilon\text{-closure}(\delta(\delta^*(q_2,\varepsilon),b))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2),b))$$
$$= \varepsilon\text{-closure}(\delta(q_2,b))$$
$$= \varepsilon\text{-closure}(q_2)$$
$$= \{q_2\}$$

Step-3:-

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

$\delta'(q_0,b) = \emptyset$

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

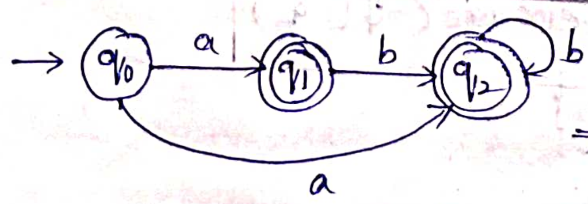
$\delta'(q_1,b) = \{q_2\}$

$\delta'(q_2,a) = \emptyset$

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

States	a	b
$\rightarrow q_0$	$\{q_1,q_2\}$	$\emptyset$
$*q_1$	$\emptyset$	$\{q_2\}$
$*q_2$	$\emptyset$	$\{q_2\}$

Step-4:-



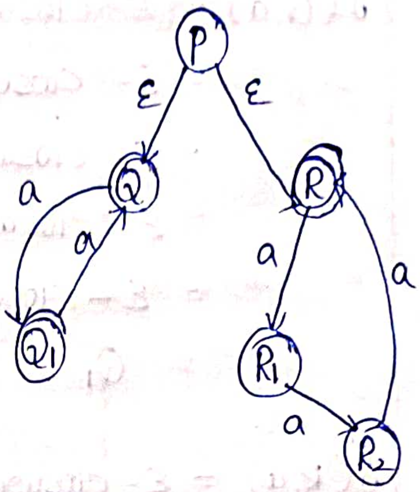
$\Rightarrow$ NFA

28/12/21
Tuesday

Example:-

$L = \{a^n \mid n \text{ is even or divisible by } 3\}$

Convert ε-NFA to NFA



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

$Q = \{P, Q, R, Q_1, R_1, R_2\}$

$\Sigma = \{a\}$

$q_0 = P$

$F = \{Q, R\}$

δ

δ	a	ε
P	–	Q, R
Q	Q_1	–
R	R_1	–
Q_1	Q	–
R_1	R_2	–
R_2	R	–

$\varepsilon\text{-closure}(P) = \{P, Q, R\}$

$\varepsilon\text{-closure}(Q) = \{Q\}$

$\varepsilon\text{-closure}(R) = \{R\}$

$\varepsilon\text{-closure}(Q_1) = \{Q_1\}$

$\varepsilon\text{-closure}(R_1) = \{R_1\}$

$\varepsilon\text{-closure}(R_2) = \{R_2\}$

Step-2:-

$\delta'(P, a) = \varepsilon\text{-closure}(\delta(\delta^*(P, \varepsilon), a))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(P), a))$

$= \varepsilon\text{-closure}(\delta(P, Q, R), a))$

$= \varepsilon\text{-closure}(\delta(P, a) \cup \delta(Q, a) \cup \delta(R, a))$

$= \varepsilon\text{-closure}(\phi \cup Q_1 \cup R_1)$

$= \varepsilon\text{-closure}(Q_1) \cup \varepsilon\text{-closure}(R_1)$

$= Q_1 \cup R_1$

$= \{Q_1, R_1\}$

$\delta'(Q,a) = \varepsilon\text{-closure}(\delta(\delta^*(Q,\varepsilon),a))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(Q),a))$

$= \varepsilon\text{-closure}(\delta(Q,a))$

$= \varepsilon\text{-closure}(\delta(Q,a))$

$= \varepsilon\text{-closure}(Q_1)$

$= Q_1$

$\delta'(R,a) = \varepsilon\text{-closure}(\delta(\delta^*(R,\varepsilon),a))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(R),a))$

$= \varepsilon\text{-closure}(\delta(R,a))$

$= \varepsilon\text{-closure}(R_1)$

$= R_1$

$\delta'(Q_1,a) = \varepsilon\text{-closure}(\delta(\delta^*(Q_1,\varepsilon),a))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(Q_1),a))$

$= \varepsilon\text{-closure}(\delta(Q_1,a))$

$= \varepsilon\text{-closure}(Q)$

$= Q$

$\delta'(R_1,a) = \varepsilon\text{-closure}(\delta(\delta^*(R_1,\varepsilon),a))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(R_1),a))$

$= \varepsilon\text{-closure}(\delta(R_1,a))$

$= \varepsilon\text{-closure}(R_2)$

$= R_2$

$\delta'(R_2,a) = \varepsilon\text{-closure}(\delta(\delta^*(R_2,\varepsilon),a))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(R_2),a))$

$= \varepsilon\text{-closure}(\delta(R_2,a))$

$= \varepsilon\text{-closure}(R)$

= R

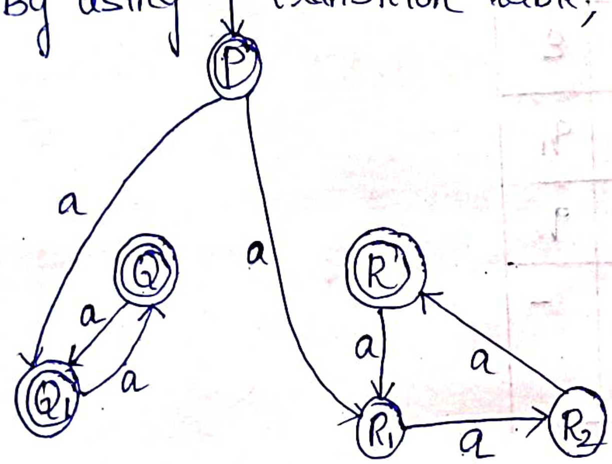
Step-3:-

Table:-

δ'

States	a
$\rightarrow * P$	$\{Q_1, R_1\}$
$*Q$	Q_1
$*R$	R_1
Q_1	Q
R_1	R_2
R_2	R

Step-4:- By using transition table, Construct NFA



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

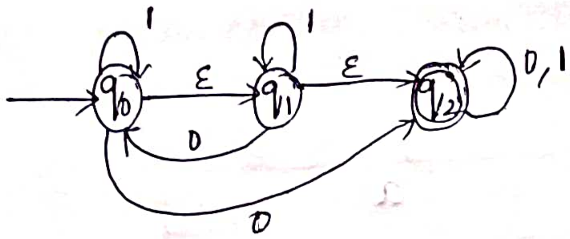
$Q = \{P, Q, R, R_1, R_2, Q_1\}$

$\Sigma = \{a\}$

$q_0 = P$

$F = \{P, Q, R\}$.

Ex:- (2)



Step-1:- Finding ε-closures

Step-2:- Finding δ' transition on each i/p symbol

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

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

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

$q_0 = q_0$

$F = \{q_2\}$

Transition Table:-

δ	0	1	ε
$\rightarrow q_0$	q_2	q_0	q_1
q_1	q_0	q_1	q_2
$*q_2$	q_2	q_2	-

Step-1:- To find the ε-closures of all states.

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

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

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

Step-2:- Finding δ' transition on each i/p symbol

$$\delta'(q_0, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_0, \varepsilon), 0))$$

$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), 0))$$

$$= \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 0))$$

$$= \varepsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_1, 0)) \cup \delta(q_2, 0))$$

$$= \varepsilon\text{-closure}(q_2 \cup q_0 \cup q_2)$$

$= \varepsilon\text{-closure}(q_2) \cup \varepsilon\text{-closure}(q_0) \cup \varepsilon\text{-closure}(q_2)$

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

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

$\delta'(q_0, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_0, \varepsilon), 1))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), 1))$

$= \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 1))$

$= \varepsilon\text{-closure}(\delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1))$

$= \varepsilon\text{-closure}(q_0 \cup q_1 \cup q_2)$

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

$\delta'(q_1, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_1, \varepsilon), 0))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1), 0))$

$= \varepsilon\text{-closure}(\delta(q_1, q_2), 0)$

$= \varepsilon\text{-closure}(\delta(q_1, 0) \cup \delta(q_2, 0))$

$= \varepsilon\text{-closure}(q_0 \cup q_2)$

$= \varepsilon\text{-closure}(q_0) \cup \varepsilon\text{-closure}(q_2)$

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

$= q_0, q_1, q_2\}$

$\delta'(q_1, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_1, \varepsilon), 1))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1), 1))$

$= \varepsilon\text{-closure}(\delta(q_1, q_2), 1))$

$= \varepsilon\text{-closure}(\delta(q_1, 1), \cup \delta(q_2, 1))$

$= \varepsilon\text{-closure}(q_1 \cup q_2)$

$= \varepsilon\text{-closure}(q_1) \cup \varepsilon\text{-closure}(q_2)$

$= \{q_1, q_2\} \cup \{q_2\}$

$= \{q_1, q_2\}$

$\delta'(q_2, 0) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_2, \varepsilon), 0))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2), 0))$

$= \varepsilon\text{-closure}(\delta(q_2), 0))$

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

$= q_2$

$\delta'(q_2, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_2, \varepsilon), 1))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2), 1))$

$= \varepsilon\text{-closure}(\delta(q_2, 1))$

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

$= q_2$

Step-3:- Construct transition Table

Transition Table

δ'	0	1
$\rightarrow * q_0$	q_0, q_2, q_1	q_0, q_1, q_2
$* q_1$	q_0, q_1, q_2	q_1, q_2
$* q_2$	q_2	q_2

Step-4:- Construct NFA by using transition Table

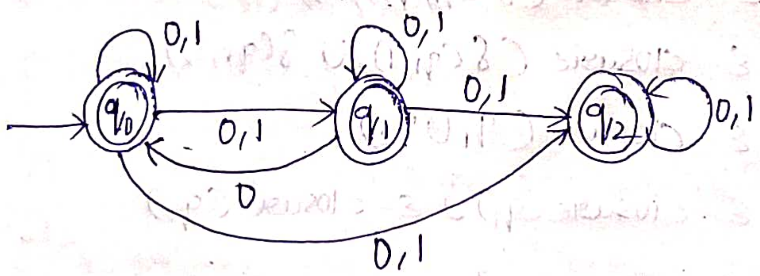
$M' = (Q, \Sigma, \delta', q_0, F)$

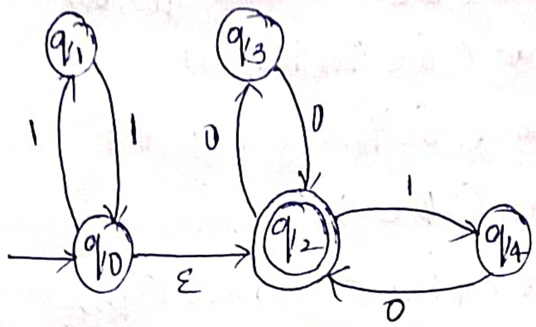
$Q = \{q_0, q_1, q_2\}$

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

$q_0 = q_0$

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





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

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

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

$q_0 = q_0$

$F = \{q_2\}$

Transition Table:-

δ	0	1	ε
$\rightarrow q_0$	–	q_1	q_2
q_1	–	q_0	–
$*q_2$	q_3	q_4	–
q_3	q_2	–	–
q_4	q_2	–	–

Step-1:- Finding ε-closures

ε-closure $(q_0) = \{q_0, q_2\}$

ε-closure $(q_1) = \{q_1\}$

ε-closure $(q_2) = \{q_2\}$

ε-closure $(q_3) = \{q_3\}$

ε-closure $(q_4) = \{q_4\}$

Step-2:- Finding δ' transition on each i/p symbol.

$$\delta'(q_0, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_0,\varepsilon), 0))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), 0))$$
$$= \varepsilon\text{-closure}(\delta(q_0, q_2), 0))$$
$$= \varepsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_2, 0))$$
$$= \varepsilon\text{-closure}(\phi \cup q_3)$$
$$= \varepsilon\text{-closure}(\phi) \cup \varepsilon\text{-closure}(q_3)$$
$$= \phi \cup q_3$$
$$= \{q_3\}$$

$$\delta'(q_0, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_0,\varepsilon), 1))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), 1))$$
$$= \varepsilon\text{-closure}(\delta(q_0, q_2), 1)$$
$$= \varepsilon\text{-closure}(\delta(q_0, 1) \cup \delta(q_2, 1))$$
$$= \varepsilon\text{-closure}(q_1 \cup q_4)$$
$$= \varepsilon\text{-closure}(q_1) \cup \varepsilon\text{-closure}(q_4)$$
$$= \{q_1\} \cup \{q_4\}$$
$$= \{q_1, q_4\}$$

$$\delta'(q_1, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_1,\varepsilon), 0))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1), 0))$$
$$= \varepsilon\text{-closure}(\delta(q_1), 0))$$
$$= \varepsilon\text{-closure}(\phi)$$
$$= \phi$$

$$\delta'(q_1, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_1,\varepsilon); 1))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1), 1))$$
$$= \varepsilon\text{-closure}(\delta(q_1), 1))$$
$$= \varepsilon\text{-closure}(q_0)$$
$$= \{q_0, q_2\}$$

$$\delta'(q_2, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_2,\varepsilon), 0))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2), 0))$$
$$= \varepsilon\text{-closure}(\delta(q_2), 0))$$
$$= \varepsilon\text{-closure}(q_3)$$

$= \{q_3\}$

$\delta'(q_2, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_2, \varepsilon), 1))$
$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2), 1))$
$= \varepsilon\text{-closure}(\delta(q_2, 1))$
$= \varepsilon\text{-closure}(q_4)$
$= \{q_4\}$

$\delta'(q_3, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_3, \varepsilon), 0))$
$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_3), 0))$
$= \varepsilon\text{-closure}(\delta(q_3, 0))$
$= \varepsilon\text{-closure}(q_2)$
$= q_2$

$\delta'(q_4, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_4, \varepsilon), 0))$
$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_4), 0))$
$= \varepsilon\text{-closure}(\delta(q_4, 0))$
$= \varepsilon\text{-closure}(q_2)$
$= q_2$

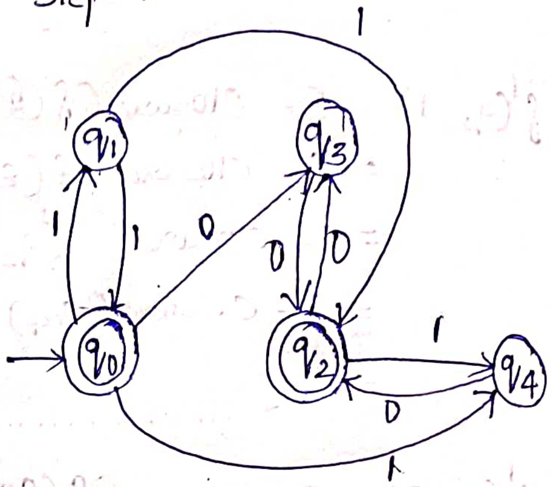
$\delta'(q_3, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_3, \varepsilon), 1))$
$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_3), 1))$
$= \varepsilon\text{-closure}(\delta(q_3, 1))$
$= \varepsilon\text{-closure}(\phi)$
$= \phi$

$\delta'(q_4, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_4, \varepsilon), 1))$
$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_4), 1))$
$= \varepsilon\text{-closure}(\delta(q_4, 1))$
$= \varepsilon\text{-closure}(\phi)$
$= \phi$

Transition State:-

δ'	0	1
$\rightarrow * q_0$	$\{q_3\}$	$\{q_1, q_4\}$
q_1	-	$\{q_0, q_2\}$
$* q_2$	$\{q_3\}$	$\{q_4\}$
q_3	$\{q_2\}$	ϕ
q_4	$\{q_2\}$	ϕ

Step-4:-



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

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

$\Sigma = \{0, 1)$

$q_0 = q_0$

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

NFA to DFA Conversion:-

- Let $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA which accepts the language $L(M)$. There should be equivalent DFA denoted $M' = (Q', \Sigma', q_0', \delta', F')$ such that $L(M) = L(M')$

- Steps for Converting NFA to DFA:

Step-1:- Initially $Q' = \phi$

Step-2:- Add q_0 of NFA to Q'. Then find the transitions from this start state.

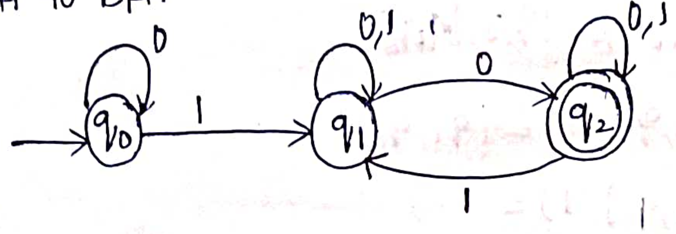
Step-3:- In Q', find the possible set of states for each input symbol. If this set of states is not in Q', then add it to Q'.

Step-4:- In DFA, the final state will be all the states which contain F(final states of NFA)

30/12/21
Thursday

Example:-

NFA to DFA



δ:

State	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$\{q_1, q_2\}$	q_1
$*q_2$	q_2	$\{q_1, q_2\}$

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

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

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

$q_0 = q_0$

$F = \{q_2\}$

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

$q_0' = q_0$

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

Step:- Initially $Q' = \phi$

Step-2:- $Q' = \{q_0\}$

Find the transitions from initial state

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

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

Step-3:- $Q' = \{[q_0], [q_1]\}$; Add q_1 to Q'

$\delta'([q_1], 0) = [q_1, q_2] \Rightarrow$ New state q_2

$\delta'([q_1], 1) = [q_1]$

Step-4:- Add q_2 to Q'

$$Q' = \{[q_0],[q_1],[q_1,q_2]\}$$

$$\delta'([q_1,q_2],0) = [q_1,q_2]$$

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

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

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

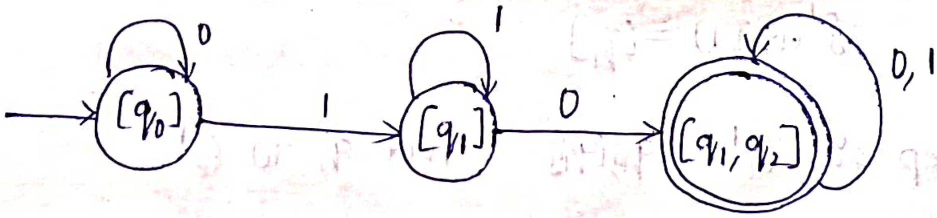
$Q' = \{[q_0],[q_1],[q_1,q_2]\}$

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

$q_0' = q_0$

$F' = \{[q_1,q_2]\}$

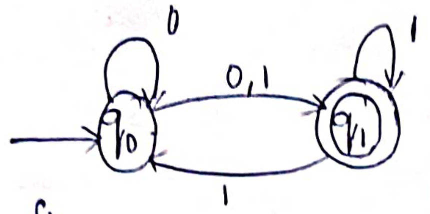
δ'	0	1
$\rightarrow q_0$	$[q_0]$	$[q_1]$
q_1	$[q_1,q_2]$	$[q_1]$
$*[q_1,q_2]$	$[q_2]$	$[q_1,q_2]$



It is our DFA.

$L = \{w \mid w$ is 10 as substring$\}$

Ex: ② NFA to DFA



δ:

State	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	q_1
$* q_1$	ϕ	$\{q_0, q_1\}$

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

$Q = \{q_0, q_1\}$

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

$q_0 = q_0$

$F = \{q_1\}$

We have to find DFA of given NFA

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

$Q' = \{ \}$

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

$q_0' = q_0$

Now

Step-1:- Initially Q add to Q'

$Q' = \{ \}$

Step-2:- Add q_0 of NFA to Q'

$Q' = \{[q_0]\}$

Then find the transitions from this start state.

Step-3:-

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

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

Both are newly generated $[q_1]$, $[q_0, q_1]$ are added to Q'

$$Q' = \{[q_0], [q_1], [q_0, q_1]\}$$

The possible states of q_1

$$\delta'(q_1, 0) = \phi$$

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

The possible states of $[q_0, q_1]$

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

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

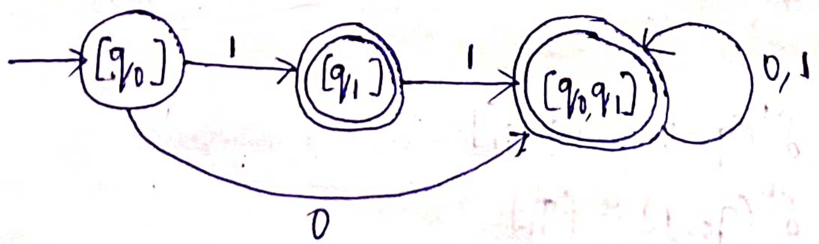
Both are old states

$$Q' = \{[q_0][q_1][q_0, q_1]\}$$

Step-4:- Construct the transition table

δ'	0	1
$[q_0]$	$[q_0, q_1]$	$[q_1]$
$[q_1]$	ϕ	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_1]$

Step -5:- Final states are $[q_0, q_1]$, $[q_1]$



Conversion from ε–NFA to DFA:-

Step-1:- Take ε-closure for starting state of NFA as a starting state of DFA.

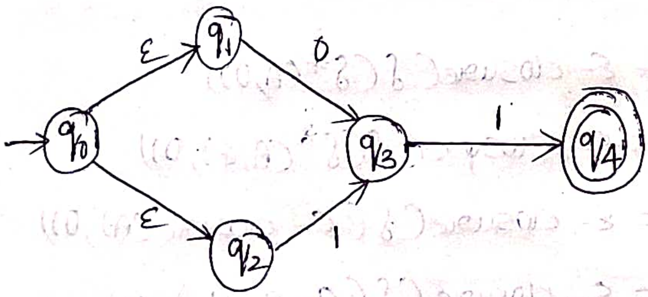
Step-2:- Find the states for each input symbol that can be traversed from the present. i.e., the union of transition value and their closures state of DFA.

Step-3:- If we found a new state, take it as current state and repeat step 2

Step-4:- Repeat step-2 and step 3 until there is no new state present in the transition table of DFA.

Step-5:- Make the states of DFA as a final state which contains the final state of NFA.

Ex:- ① ε–NFA to DFA



For given ε–NFA

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

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

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

$q_0 = q_0$

$F = \{q_4\}$

Transition Table:-

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

Step-1:- Finding ε-closures of all states of ε-NFA

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

$\varepsilon\text{-closure}(q_1) = \{q_1\}$

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

$\varepsilon\text{-closure}(q_3) = \{q_3\}$

$\varepsilon\text{-closure}(q_4) = \{q_4\}$

Step-2:- Let $\varepsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$ be State A.

$Q' = \{A\}$

$\delta'(A, 0) = \varepsilon\text{-closure}(\delta(\delta^*(A, 0))$

$= \varepsilon\text{-closure}(\delta(\delta^*(A, \varepsilon), 0))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(A), 0))$

$= \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 0))$

$= \varepsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_1, 0), \cup \delta(q_2, 0))$

$= \varepsilon\text{-closure}(\phi \cup q_3 \cup \phi)$

$= \varepsilon\text{-closure}(q_3)$

$= \{q_3\}$ $\Rightarrow$ New state

$\delta'(A, 1) = \varepsilon\text{-closure}(\delta(\delta^*(A, \varepsilon), 1))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(A), 1))$

$= \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 1))$

$= \varepsilon\text{-closure}(\delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1))$

$= \varepsilon\text{-closure}(\phi \cup \phi \cup q_3)$

$= \varepsilon\text{-closure}(q_3)$

$= q_3 \Rightarrow B$

Add B to Q'

$Q' = \{A, B\}$

The possible states of B

$\delta'(B, 0) = \varepsilon\text{-closure}(\delta(\delta^*(B, \varepsilon), 0))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(B), 0))$

$= \varepsilon\text{-closure}(\delta(q_3, 0))$

$= \phi$

$\delta'(B, 1) = \varepsilon\text{-closure}(\delta(\delta^*(B, \varepsilon), 1))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(B), 1))$

$= \varepsilon\text{-closure}(\delta(q_3, 1))$

$= q_4 \Rightarrow$ New state $\Rightarrow C$; It is named as C

Add C to Q'

$Q' = \{A, B, C\}$

The possible states of C

$\delta'(C, 0) = \varepsilon\text{-closure}(\delta(\delta^*(C, \varepsilon), 0))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(C), 0))$

$= \varepsilon\text{-closure}(\delta(q_4, 0))$

$= \phi$

$\delta'(C, 1) = \varepsilon\text{-closure}(\delta(\delta^*(C, \varepsilon), 1))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(C), 1))$

$= \varepsilon\text{-closure}(\delta(q_4, 1))$

$= \phi$

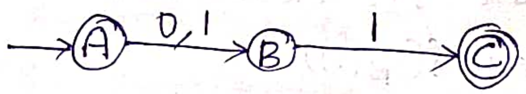
Step-3:- Construct Transition State Table

δ'	0	1
→A	$\{q_3\}$	B
B	ϕ	C
*C	ϕ	ϕ

Step-4:- Make the final states of DFA as a final states which contains the final state of NFA.

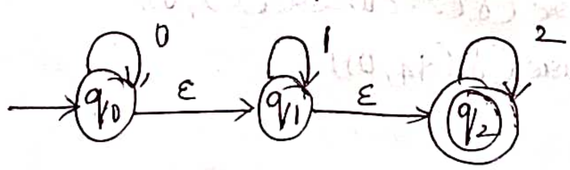
ε – closure $(q_4) = \{q_4\}$

The final state of NFA is q_4 which will be in ε- closure of q_4 is the final state of DFA.



Language which will be accepted by this DFA is $L = \{0, 1, 11\}$

Ex:- 2 ε – NFA to DFA



Given ε-NFA

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

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

$\Sigma = \{0, 1, 2\}$

$q_0 = q_0$

$F = \{q_2\}$

Transition Table:-

δ	0	1	2	ε
q_0	q_0	ϕ	ϕ	q_1
q_1	ϕ	q_1	ϕ	q_2
q_2	ϕ	ϕ	q_2	ϕ

Step-1:- Finding ε-closure of all states of ε-NFA

ε-closure $(q_0) = \{q_0, q_1, q_2\}$

ε-closure $(q_1) = \{q_1, q_2\}$

ε-closure $(q_2) = \{q_2\}$

Step-2:- Take ε-closure for starting state of NFA as a starting state of DFA

$A = \{q_0, q_1, q_2\}$.

$Q' = \{A\}$

The possible states of A

$\delta'(A, 0) = \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 0))$

$= \varepsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$

$= \varepsilon\text{-closure}(q_0 \cup \phi \cup \phi)$

$= \varepsilon\text{-closure}(q_0)$

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

$\delta'(A, 1) = \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 1))$

$= \varepsilon\text{-closure}(\delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1))$

$= \varepsilon\text{-closure}(\delta(\phi \cup q_1 \cup \phi)$

$= \{q_1, q_2\}$ New State

Let $\{q_1, q_2\}$ named as B, and added to Q'

$Q' = \{A, B\}$

The possible states of B

$\delta'(B,0) = \varepsilon\text{-closure}(\delta(q_0,q_2),0))$

$= \varepsilon\text{-closure}(\delta(q_1,0),\cup\,\delta(q_2,0))$

$= \varepsilon\text{-closure}(\delta(q_1,0)\cup\delta(q_2,0))$

$= \varepsilon\text{-closure}(\phi\cup\phi)$

$= \phi$

$\delta'(B,1) = \varepsilon\text{-closure}(\delta(q_0,q_2),1))$

$= \varepsilon\text{-closure}(\delta(q_0,1)\cup\delta(q_2,1))$

$= \varepsilon\text{-closure}(\phi\cup\phi)$

$= \phi$

$Q' = \{A,B\}$

$\delta'(B,2) = \varepsilon\text{-closure}(\delta(B,\varepsilon),2))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(B),2))$

$= \varepsilon\text{-closure}(\delta(q_0,q_2),2))$

$= \varepsilon\text{-closure}(\delta(q_0,2)\cup\delta(q_2,2))$

$= \varepsilon\text{-closure}(\phi\cup q_2)$

$= \{q_2\} \Rightarrow C$

Add C to Q'

$Q' = \{A,B,C\}$

The possible states of C

$\delta'(C,0) = \varepsilon\text{-closure}(\delta(\delta^*(C,\varepsilon),0))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(C),0))$

$= \varepsilon\text{-closure}(\delta(q_2,0))$

$= \varepsilon\text{-closure}(\phi)$

$= \phi.$

$\delta'(C,1) = \varepsilon\text{-closure}(\delta(\delta^*(C,\varepsilon),1))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(C),1))$

$= \varepsilon\text{-closure}(\delta(q_2,1))$

$= \varepsilon\text{-closure}(\phi)$

$= \phi$

$\delta'(C,2) = \varepsilon\text{-closure}(\delta(q_2, 2))$
$= \varepsilon\text{-closure}(q_2)$
$= \{q_2\} \Rightarrow \text{old}$

$Q' = \{A, B, C\}$

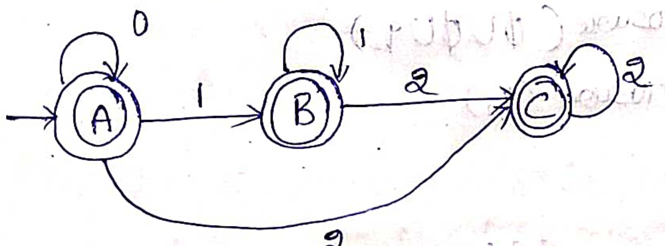
$\delta'(A,2) = \varepsilon\text{-closure}(\delta(\delta^*(A,\varepsilon), 2))$
$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(A), 2))$
$= \varepsilon\text{-closure}(\delta((q_0, q_1, q_2), 2))$
$= \varepsilon\text{-closure}(\delta(q_0,2) \cup \delta(q_1,2) \cup \delta(q_2,2))$
$= \varepsilon\text{-closure}(\phi \cup \phi \cup q_2)$
$= q_2 \Rightarrow \text{old}$

Step-3:- Construct Transition Table.

δ	0	1	2
A	A	B	C
B	–	B	C
C	–	–	C

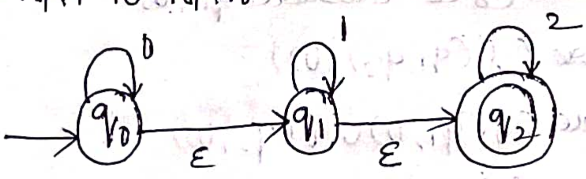
Step-4:- Draw the DFA

Make the final states as A, B, C



01/01/2022
Saturday

ε–NFA to NFA:-



Step-1:- Find all ε-closures

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

Step-2:-

$$\delta'(q_0, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_0,\varepsilon),0))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0),0))$$
$$= \varepsilon\text{-closure}(\delta(q_0,q_1,q_2),0))$$
$$= \varepsilon\text{-closure}(\delta(q_0,0) \cup \delta(q_1,0) \cup \delta(q_2,0))$$
$$= \varepsilon\text{-closure}(q_0 \cup \phi \cup \phi)$$
$$= \varepsilon\text{-closure}(q_0)$$
$$= \{q_0, q_1, q_2\}$$

$$\delta'(q_0, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_0,\varepsilon),1))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0),1))$$
$$= \varepsilon\text{-closure}(\delta(q_0,q_1,q_2),1))$$
$$= \varepsilon\text{-closure}(\delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_2,1))$$
$$= \varepsilon\text{-closure}(\phi \cup q_1 \cup \phi)$$
$$= \varepsilon\text{-closure}(q_1)$$
$$= \{q_1, q_2\}$$

$$\delta'(q_0, 2) = \varepsilon\text{-closure}(\delta(\delta^*(q_0,\varepsilon),2))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0),2))$$
$$= \varepsilon\text{-closure}(\delta(q_0,q_1,q_2),2))$$
$$= \varepsilon\text{-closure}(\delta(q_0,2) \cup \delta(q_1,2), \cup \delta(q_2,2))$$
$$= \varepsilon\text{-closure}(\phi \cup \phi \cup q_2)$$
$$= \varepsilon\text{-closure}(q_2)$$
$$= \{q_2\}$$

$$\delta'(q_1, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_1,\varepsilon),0))$$
$$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1),0))$$
$$= \varepsilon\text{-closure}(\delta(q_1,q_2),0))$$
$$= \varepsilon\text{-closure}(\delta(q_1,0) \cup \delta(q_2,0))$$
$$= \varepsilon\text{-closure}(\phi \cup \phi)$$
$$= \phi$$

$$\delta'(q_1, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_1,\varepsilon),1))$$
$$= \varepsilon\text{-closure}(\delta(\delta\text{-closure}(q_1),1))$$
$$= \varepsilon\text{-closure}(\delta(q_1,q_2),1))$$

$= \varepsilon\text{-closure}(\delta(q_1,1) \cup \delta(q_2,1))$

$= \varepsilon\text{-closure}(q_1 \cup \phi)$

$= \varepsilon\text{-closure}$

$= \{q_1, q_2\}$

$\delta'(q_1, 2) = \varepsilon\text{-closure}(\delta(\delta^*(q_1,\varepsilon), 2))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1), 2))$

$= \varepsilon\text{-closure}(\delta(q_1,q_2), 2))$

$= \varepsilon\text{-closure}(\delta(q_1,2) \cup \delta(q_2,2))$

$= \varepsilon\text{-closure}(\phi \cup q_2)$

$= \{q_2\}$

$\delta'(q_2, 0) = \varepsilon\text{-closure}(\delta(\delta^*(q_2,\varepsilon), 0))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2), 0))$

$= \varepsilon\text{-closure}(\delta(q_2, 0))$

$= \varepsilon\text{-closure}(\phi)$

$= \phi$

$\delta'(q_2, 1) = \varepsilon\text{-closure}(\delta(\delta^*(q_2,\varepsilon), 01))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2), 1))$

$= \varepsilon\text{-closure}(\delta(q_2, 1))$

$= \varepsilon\text{-closure}(\phi)$

$= \phi$

$\delta'(q_2, 2) = \varepsilon\text{-closure}(\delta(\delta^*(q_2,\varepsilon), 2))$

$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-clousure}(q_2), 2))$

$= \varepsilon\text{-closure}(\delta(q_2, 2))$

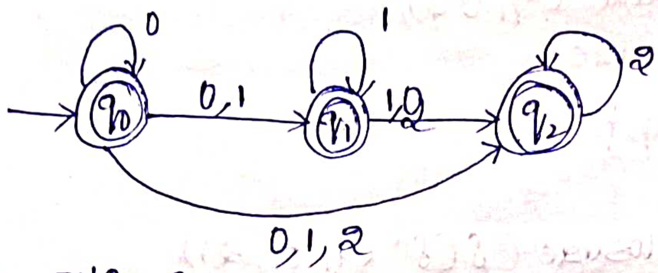
$= \varepsilon\text{-closure}(q_2)$

$= \{q_2\}$

Step-3:- Construct transition table.

δ'	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\{\phi\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	ϕ	ϕ	$\{q_2\}$

Step-4:- Draw a NFA without ε



This is NFA without ε

Now

NFA to DFA

For given NFA

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

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

$\Sigma = \{0, 1, 2\}$

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

$q_0 = q_0$

Transition Table:-

δ	0	1	2
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	$\{q_2\}$

Step-1:- initially $Q' = \phi$, Add ϕ to Q'

Step-2:- Add q_0 of NFA to Q'

$Q' = \{[q_0]\}$

Then find the transitions from this start state

Step-3:-

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

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

$\delta'(q_0, 2) = [q_2]$

All are newly generated states, are added to Q'

$$Q' = \{[q_0], [q_0, q_1, q_2], [q_1, q_2], [q_2]\}$$

The possible states of $[q_0, q_1, q_2]$

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

$$\delta'([q_0, q_1, q_2], 1) = \delta'([q_0], 1) \cup \delta'([q_1], 1) \cup \delta'([q_2], 2)$$
$$= [q_1, q_2] \cup [q_1, q_2] \cup [q_2]$$
$$= [q_1, q_2]$$

$$\delta'([q_0, q_1, q_2], 2) = \delta'([q_0], 2) \cup \delta([q_1], 2) \cup \delta([q_2], 2))$$
$$= [q_2] \cup [q_2] \cup [q_2]$$
$$= q_2$$

The possible states of $[q_1, q_2]$

$$\delta'([q_1, q_2], 0) = \delta'([q_1], 0) \cup \delta'([q_2], 0))$$
$$= \phi \cup \phi$$
$$= \phi$$

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

$$\delta'([q_1, q_2], 2)) = \delta(q_1, 2) \cup \delta(q_2, 2)$$
$$= [q_2] \cup [q_2]$$
$$= q_2$$

The possible states of $[q_2]$

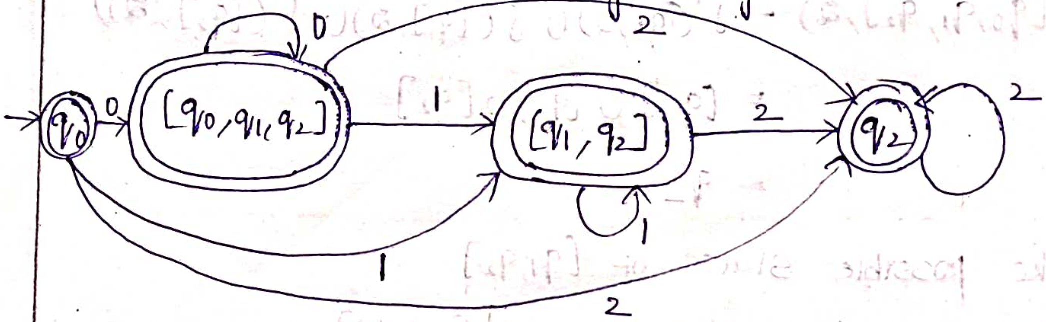
$$\delta'(q_2, 0) = \delta(q_2, 0) = \phi$$
$$\delta'(q_2, 1) = \delta(q_2, 1) = \phi$$
$$\delta'(q_2, 2) = q_2$$

$Q' = \{[q_0], [q_0, q_1, q_2], [q_1, q_2], [q_2]\}$

Step-3:- Construct transition Table

δ	0	1	2
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$[q_1, q_2]$	$[q_2]$
$[q_1, q_2]$	ϕ	$[q_1, q_2]$	$[q_2]$
$[q_2]$	ϕ	ϕ	$[q_2]$
q_0	$[q_0, q_1, q_2]$	$[q_1, q_2]$	$[q_2]$

Step-4:- Construct DFA by using transition Table

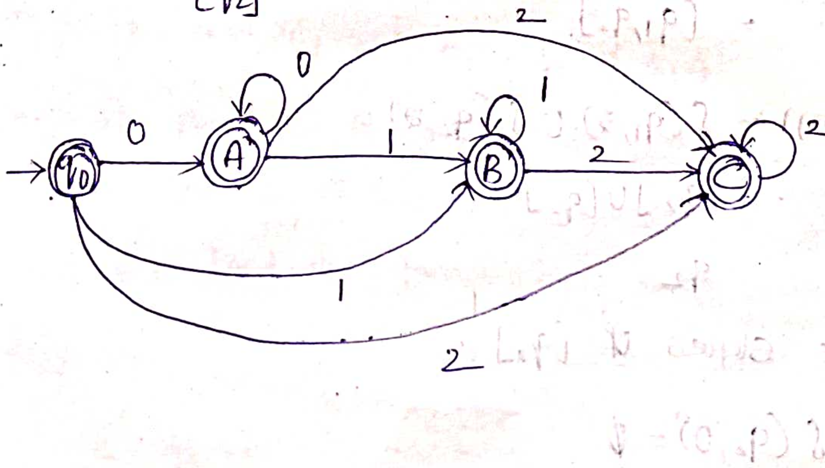


in ε-NFA to DFA

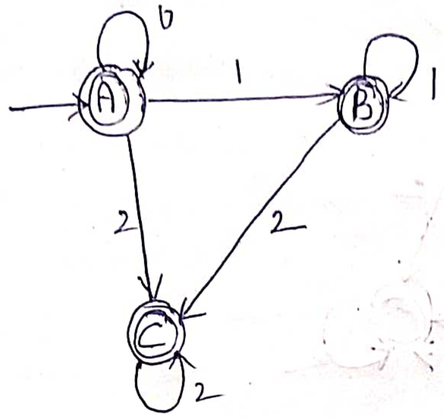
$A = \{q_0, q_1, q_2\}$

$B = \{q_1, q_2\}$

$C = \{q_2\}$



Ex:- ε - NFA to DFA



For given ε-NFA

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

$Q = \{A, B, C\}$

$\Sigma = \{0, 1, 2\}$

$q_0 = A$

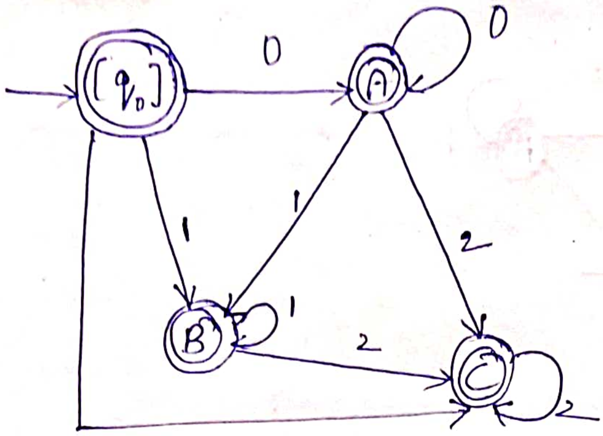
$F = \{A, B, C\}$

Transition Table:-

δ	0	1	2	ε
A	A	B	C	-
B	ϕ	B	C	-
C	ϕ	ϕ	C	-

ε-NFA - NFA - DFA

δ'	0	1	2
$[q_0]^*$	$[q_0, q_1, q_2]$	$[q_1, q_2]$	$[q_2]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$[q_1, q_2]$	$[q_2]$
$[q_1, q_2]$	ϕ	$[q_1, q_2]$	$[q_2]$
$[q_2]$	ϕ	ϕ	$[q_2]$

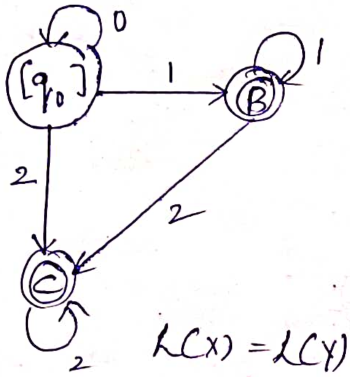


(Y)

Both are not same, but both are Equivalent. X is minimize from for Y.

Suppose at right side, we remove the duplication i.e., $[q_0, q_1, q_2]$

then DFA will be



$L(X) = L(Y)$

3/1/22
Monday.

Minimization of FSM

⇒ Reducing the no. of states from given FA

⇒ Eliminating redundant states from FSM

→ Step-1:- Remove all the states that are unreachable from the initial state via any set of the transition.

→ Step-2:- Draw the transition table for all pair of States.

→ Step-3:- Now split the transition table into two tables T_1 and T_2. T_1 Contains all final states and T_2 Contains non-final states.

→ Step-4:- Find similar row from T_1 such that

1. $\delta(q,a)=p$
2. $\delta(r,a)=p$

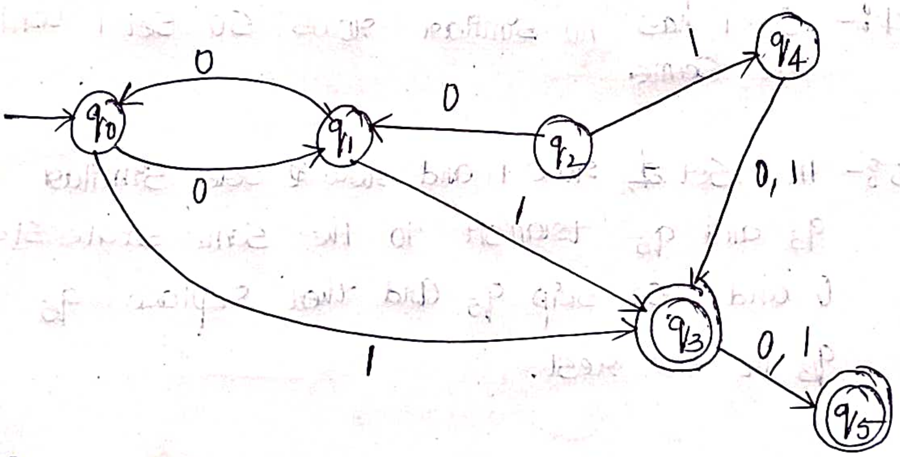
That means, find the two states which have the same value of a and b and remove from one of them.

→ Step-5:- Repeat Step-3 until we find no similar rows available in the transition Table T_1.

→ Step-6:- Repeat Step-3 and Step-4 for table T_2 also.

→ Step-7:- Now Combine the reduced T_1 and T_2 tables. The Combined transition table is the transition table of minimized FSM.

Ex:-



For a given Automata

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

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

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

$q_0=q_0$

$F=\{q_3,q_5\}$

Step-2:-

Transition Table:-

δ	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_1	q_3
$*q_3$	q_5	q_5
$*q_5$	q_5	q_5

Step-3:- Split the transition Table into two tables

$T_1 \rightarrow$ Final States

$T_2 \rightarrow$ Non-Final States

T_1	0	1
$*q_3$	q_5	q_5
$*q_5$	q_5	q_5

Set-1

T_2	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_0	q_3

Set-2

Step-4:- Set-1 has no similar rows so Set 1 will be the same.

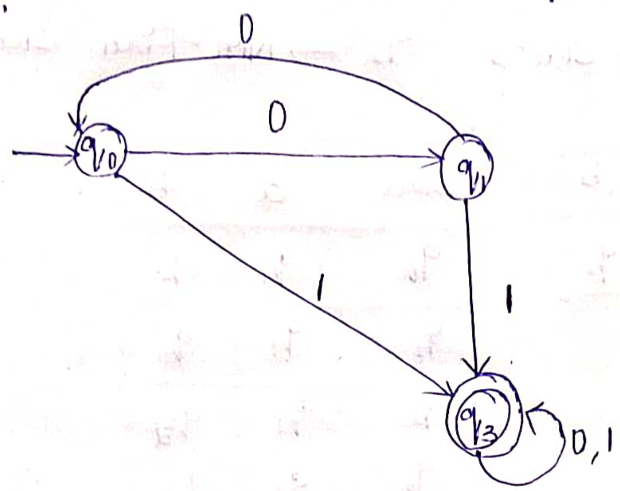
Step-5:- In Set ~~1~~, row 1 and row 2 are similar since q_3 and q_5 transit to the same state state on 0 and 1. So skip q_5 and then replace q_5 by q_3 in the rest.

State	0	1
q_3	q_3	q_3

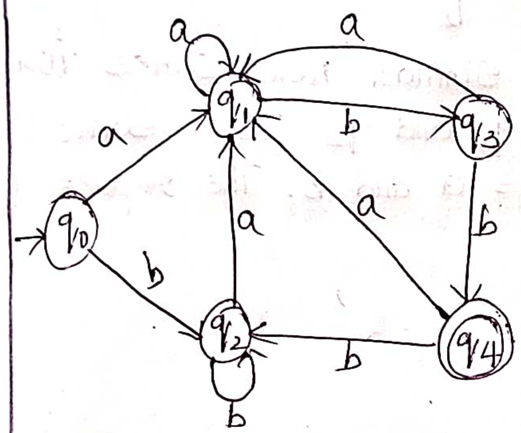
Step-6:- Now Combine set 1 and set 2

State	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_0	q_3
$*q_3$		

Construct Finite Automata by using above Transition Table.



Ex:-2:-



For given Automata

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

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

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

$q_0 = q_0$

$F = \{q_4\}$

Step-1:- There is unreachable states. All are reachable States.

Step-2:- Construct Transition Tables.

δ	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

Step-3:- Now split the Transition tables.

$T_1 \rightarrow$ Final States $\quad T_2 \rightarrow$ Non-Final States

State	a	b
q_4	q_1	q_2

T_1

State	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

T_2

Step-4:- From T_2, To find similar next states for different states. For q_0 and q_2 have same next states for inputs a and b. The replace q_2 by q_0 (or) remove q_2

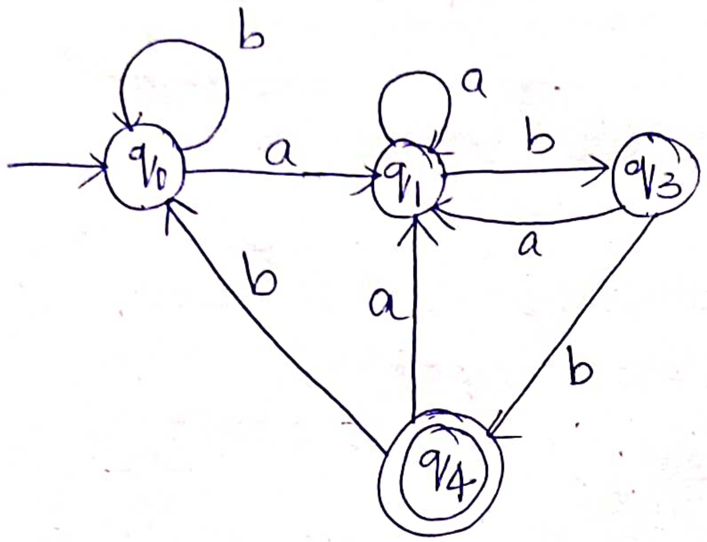
State	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_3	q_1	q_4

T_2

Step-5:- Combine T_1 and T_2 to form the transition table for minimised FSM.

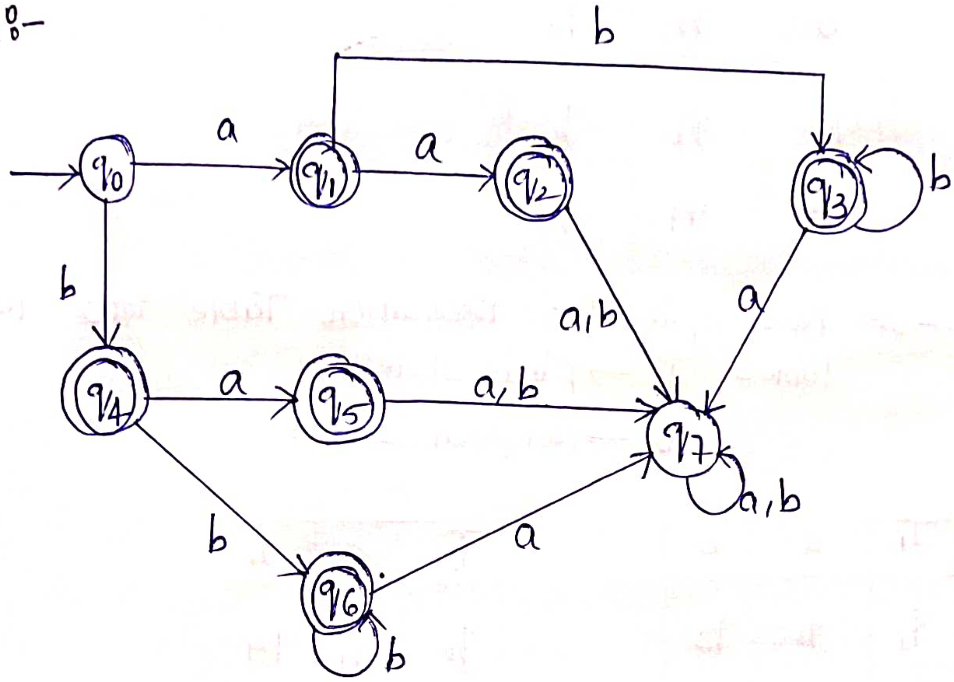
T	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_3
q_3	q_1	q_4
$*q_4$	q_1	q_0

Step-6:- Construct finite Automata by using this table.



b
a
a
b
b
a
a
b
q_0
q_1
q_3
q_4

Ex:-



Ans:- For given Automata

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

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

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

$\delta = ?$

$q_0 = q_0$

$F = \{q_1, q_2, q_3, q_4, q_5, q_6\}$

Step-1:- There is not unreachable states. All are reachable states.

Step-2:- Construct Transition states.

T	a	b
q_0	q_1	q_4
q_1	q_2	q_3
q_2	q_7	q_7

	a	b
q_3	q_7	q_3
q_4	q_5	q_6
q_5	q_7	q_7
q_6	q_7	q_6
q_7	q_7	q_7

Step-3:- Now split the Transition Table into two Tables $T_1 \rightarrow$ Final States
$T_2 \rightarrow$ NonFinal states.

T_1	a	b
q_1	q_2	q_3
q_2	q_7	q_7
q_3	q_7	q_3.
q_4	q_5	q_6
q_5	q_7	q_7
q_6	q_7	q_6

T_2	a	b
q_0	q_1	q_4
q_7	q_7	q_7

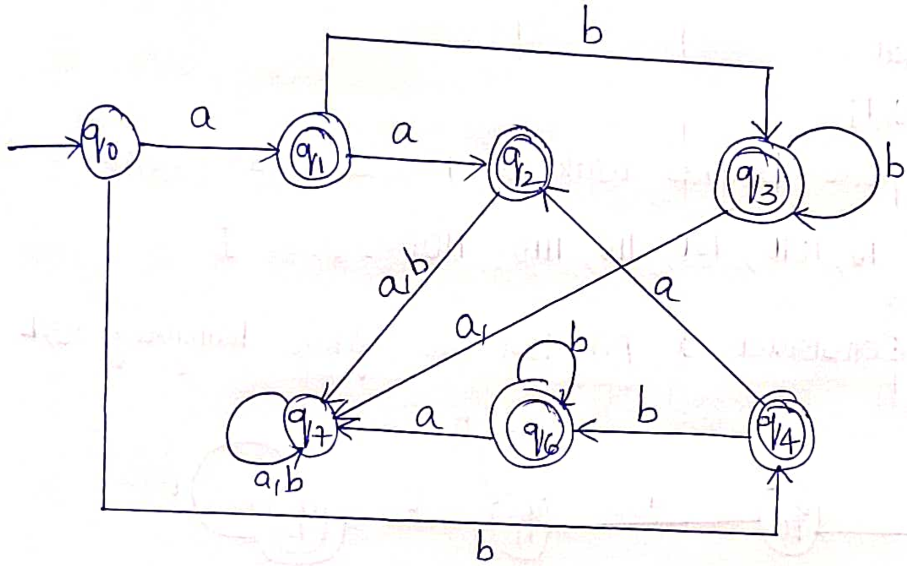
Step-4:- From T_1, find similar rows of same input symbols, q_2 and q_5, remove q_5, and replace q_5 by q_2

T_1	a	b
q_1	q_2	q_3
q_2	q_7	q_7
q_3	q_7	q_3
q_4	q_5	q_6
q_6	q_7	q_6

Step-5:- Combine T_1 and T_2 to form the transition table for minimised FSM.

T	a	b
q_0	q_1	q_4
q_1	q_2	q_3
q_2	q_7	q_7
q_3	q_7	q_3
q_4	q_2	q_6
q_6	q_7	q_6
q_7	q_7	q_7

Step-6:- Construct FA by using this Table.



17/1/22 Mon

Minimization of DFA Using Equivalence Theorem

- Eliminate all the dead states and inaccessible states from the given DFA if any

  Dead state:- All those non-final states which transit to itself for all input symbol in Σ are called as dead states.

  Inaccessible state:- All those states which can never be reached from the initial state are called as inaccessible state.

- Draw a state transition table for the given DFA.
  - Transition table shows the transition of all states on all input symbols in Σ.

- Now, start applying Equivalence Theorem.
  - Take a counter variable K and initialize it with value 0.
  - Divide Q (set of states) into two sets such that one set contains all the non-final states and other set contains all the final states.
  - This partition is called P_0.

- Update partitions
  - increment by $K = 1$
  - Find P_K by partitioning the different sets of P_{K-1}.
  - In each set of P_{K-1}, consider all the possible pair of states within each set and if the two states are distinguishable, partition the set into different sets in P_K.

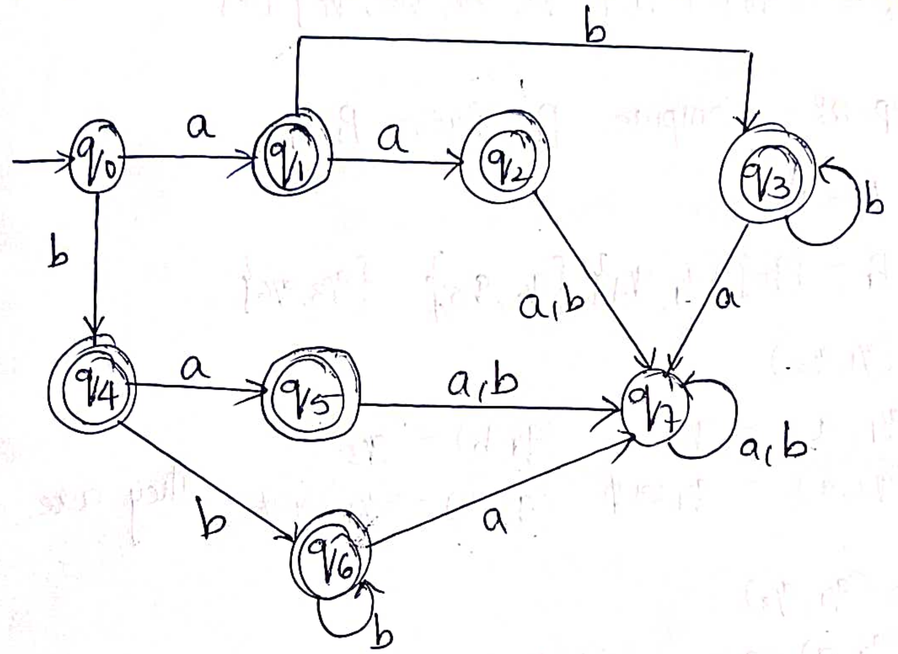
    Two states q_1 and q_2 are distinguishable in partition P_K for any input symbol 'a' if $\delta(q_1, a)$ and $\delta(q_2, a)$ are in different sets in partition P_{K-1}.

- Repeat step-4 until no change in partition occurs i.e., when $P_k = P_{k-1}$

- States which belong to the same set are equivalent. Equivalent states are merged to form a single state in the minimal DFA.

No. of states in Minimal DFA = No. of sets in P_k.

Ex:-



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

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

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

$F = \{q_1, q_2, q_3, q_4, q_5, q_6\}$

$q_0 = q_0$

δ:	a	b
q_0	q_1	q_4
q_1	q_2	q_3
q_2	q_7	q_7
q_3	q_7	q_3
q_4	q_5	q_6
q_5	q_7	q_7
q_6	q_7	q_6
q_7	q_7	q_7

In given FA, There is dead state is q_7 and there are no inaccessible state.

Step-1:- Make final states and non-final states $K=0$ as separates states by remaing dead states

$K=0$

$P_0 = \{q_0\} \{q_1, q_2, q_3, q_4, q_5, q_6\} \{\phi\}$

Step-2:- Compute P_K from P_{K-1}.

$K=1$

$P_1 = \{q_0\} \{q_1, q_4\} \{q_2, q_5\} \quad \{q_3, q_6\}$

(q_1, q_2)

$(q_1, a) = q_2 \qquad (q_1, b) = q_3$

$(q_2, a) = q_7 \text{ or } \phi \qquad (q_2, b) = q_7 \text{ or } \phi$ They are distinguishable

$\Rightarrow (q_1, q_3)$

$(q_1, a) = q_2 \qquad (q_1, b) = q_3$

$(q_3, a) = \phi \qquad (q_3, b) = q_3$ They are distinguishable

$\Rightarrow (q_2, q_3)$

$(q_2, a) = \phi \qquad (q_2, b) = \phi$

$(q_3, a) = \phi \qquad (q_3, b) = q_3$ They are distinguishable

$\Rightarrow (q_1, q_4)$

$(q_1, a) = q_2 \qquad (q_1, b) = q_3$

$(q_4, a) = q_5 \qquad (q_4, b) = q_6$ They are not distinguishable

$\Rightarrow (q_1, q_5)$

$(q_1, a) = q_2 \qquad (q_1, b) = q_3$

$(q_5, a) = \phi \qquad (q_5, b) = \phi$ They are distinguishable

$\Rightarrow (q_2, q_5)$

$(q_2, a) = \phi \qquad (q_2, b) = q_7 \text{ or } \phi$

$(q_5, a) = \phi \qquad (q_5, b) = \phi$ They are non-distinguiskable

$\Rightarrow (q_1, q_6)$

$\Rightarrow (q_1, a) = q_2$ $\quad (q_1, b) = q_3$ $\quad$ are distinguishable

$(q_6, a) = q_7$ $\quad (q_6, b) = q_6$

$\Rightarrow (q_2, q_6)$

$(q_2, a) = \phi$ $\quad (q_2, b) = \phi$ $\quad$ are distinguishable

$(q_6, a) = q_7$ $\quad (q_6, b) = q_6$

$\Rightarrow (q_3, q_6)$

$(q_3, a) = \phi$ $\quad (q_3, b) = q_3$

$(q_6, a) = \phi$ $\quad (q_6, b) = q_6$ $\quad$ are not distinguishable

$\Rightarrow (q_2, q_3)$

$(q_2, a) = \phi$ $\quad (q_2, b) = \phi$ $\quad$ are distinguishable

$(q_3, a) = \phi$ $\quad (q_3, b) = q_3$

$\Rightarrow (q_2, q_4)$

$(q_2, a) = \phi$ $\quad (q_2, b) = q_7$ $\quad$ are distinguishable

$(q_4, a) = q_5$ $\quad (q_4, b) = q_6$

$\Rightarrow (q_2, q_5)$

$(q_2, a) = \phi$ $\quad (q_2, b) = \phi$

$(q_5, a) = \phi$ $\quad (q_5, b) = \phi$ $\quad$ are not distinguishable

$\Rightarrow (q_3, q_4)$

$(q_3, a) = \phi$ $\quad (q_3, b) = q_3$ $\quad$ are distinguishable

$(q_4, a) = q_5$ $\quad (q_4, b) = q_6$

$\Rightarrow (q_3, q_5)$

$(q_3, a) = \phi$ $\quad (q_3, b) = q_3$ $\quad$ are distinguishable

$(q_5, a) = \phi$ $\quad (q_5, b)\ q_7$

$\Rightarrow (q_3, q_6)$

$(q_3, a) = q_7$ $\quad (q_3, b) = q_3$

$(q_6, a) = q_7$ $\quad (q_6, b) = q_6$ $\quad$ are not distinguishable

$\Rightarrow (q_4, q_5)$

$(q_4, a) = q_5$ $(q_4, b) = q_6$
$(q_5, a) = q_7$ $(q_5, b) = q_7$ are distinguishable

$\Rightarrow (q_4, q_6)$

$(q_4, a) = q_5$ $(q_4, b) = q_6$
$(q_6, a) = q_7$ $(q_6, b) = q_6$ are distinguishable

$\Rightarrow (q_5, q_6)$

$(q_5, a) = q_7$ $(q_5, b) = q_7$
$(q_6, a) = q_7$ $(q_6, b) = q_6$ are distinguishable

$P_1 = \{q_0\} \{q_1, q_4\} \{q_2, q_5\} \{q_3, q_6\}$

Step-3:- Compute P_2 from P_1

$P_2 =$

$\Rightarrow (q_1, q_4)$

$(q_1, a) = q_2$ $(q_1, b) = q_3$
$(q_4, a) = q_5$ $(q_4, b) = q_6$ are not distinguishable

$\Rightarrow (q_2, q_5)$

$(q_2, a) = \emptyset$ $(q_2, b) = \emptyset$
$(q_5, a) = \emptyset$ $(q_5, b) = \emptyset$ are not distinguishable

$\Rightarrow (q_3, q_6)$

$(q_3, a) = q_7$ $(q_3, b) = q_3$
$(q_6, a) = q_7$ $(q_6, b) = q_6$ are not distinguishable

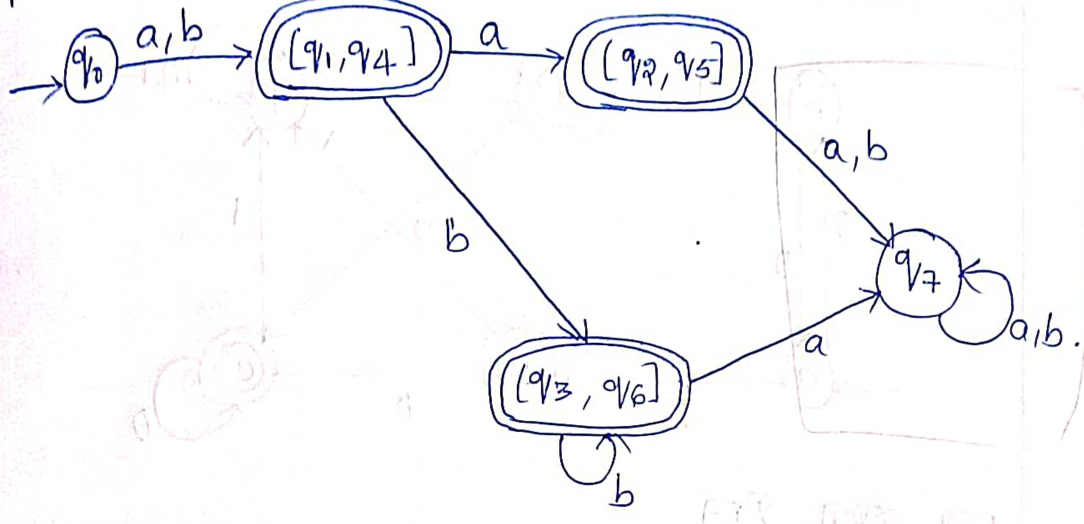
$P_{k-1} = P_2$

$P_{2-1} = P_2$

$P_1 = P_2$

$\{q_0, [q_1, q_4], [q_2, q_5], [q_3, q_6]\}$

Step-4:- Draw the diagram



20/1/22
Thursday

DFA Minimization using MyhillNerode Theorem

Step-1:- Draw a table for all pairs of states (Q_i, Q_j) not necessarily connected directly [All are unmarked initially]

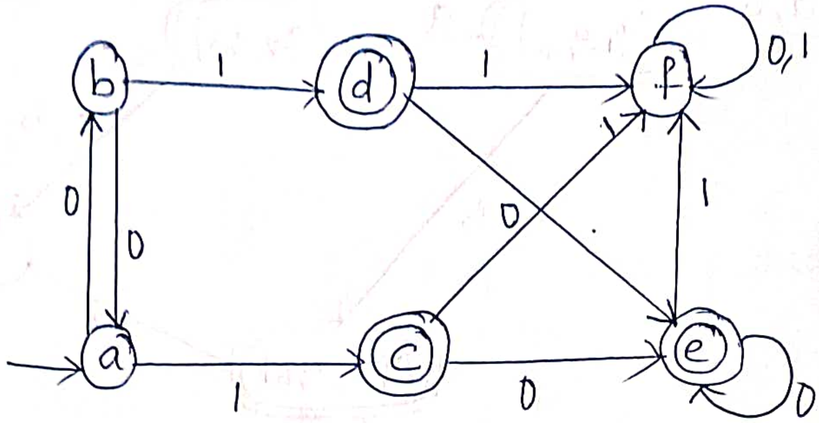
Step-2:- Consider every state pair (Q_i, Q_j) in the DFA where $Q_i \in F$ and $Q_j \notin F$ or vice versa and mark them. [Here F is the set of final states]

Step-3:- Repeat this step until we cannot mark anymore states.

- If there is an unmarked pair (Q_i, Q_j) mark it if the pair $\{\delta(Q_i, A), \delta(Q_j, A)\}$ is marked for some input alphabet.

Step-4:- Combine all the unmarked pair (Q_i, Q_j) and make them a single state in the reduced DFA.

Example:-



For given DFA

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

$Q = \{a, b, c, d, e, f\}$

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

$q_0 = a$

$F = \{c, d, e\}$

Step-1:- We draw a table for all pairs of states

(a, b)
(a, c) (b, c)
(a, d) (b, d) (c, d)
(a, e) (b, e) (c, e), (d, e)
(a, f) (b, f) (c, f) (d, f) (e, f)

	a	b	c	d	e	f
a	-	-	-	-	-	-
b		-	-	-	-	-
c			-	-	-	-
d				-	-	-
e					-	-
f						-

Step-2:- Consider every pair (Q_i, Q_j) of DFA where $Q_i \in F$ and $Q_j \notin F$ are vice versa and mark them

$(a,b) \rightarrow$ unmark ; $a \notin F$, $b \notin F$
$(a,c) \rightarrow$ Mark ; $a \notin F$, $c \in F$
$(a,d) \rightarrow$ Mark ; $a \notin F$, $d \in F$
$(a,e) \rightarrow$ Mark ; $a \notin F$, $e \in F$
$(a,f) \rightarrow$ umark ; $a \notin F$, $f \notin F$
$(b,c) \rightarrow$ Mark ; $b \notin F$, $c \in F$
$(b,d) \rightarrow$ Mark ; $b \notin F$, $d \in F$
$(b,e) \rightarrow$ Mark ; $b \notin F$, $e \in F$
$(b,f) \rightarrow$ Unmark ; $b \notin F$, $f \notin F$
$(c,d) \rightarrow$ unmark ; $c \in F$, $d \in F$
$(c,e) \rightarrow$ unmark ; $c \in F$, $e \in F$
$(c,f) \rightarrow$ Mark ; $c \in F$, $f \notin F$
$(d,e) \rightarrow$ Unmark ; $d \in F$, $e \in F$
$(d,f) \rightarrow$ Mark ; $d \in F$, $f \notin F$
$(e,f) \rightarrow$ Mark ; $e \in F$, $f \notin F$

	a	b	c	d	e	f
a	–	–	–	–	–	–
b		–	–	–	–	–
c	✓	✓	–	–	–	–
d	✓	✓		–	–	–
e	✓	✓			–	–
f			✓	✓	✓	–

Step-3:- Repeat step 2 until we cannot mark anymore states

In the above table

* (a,b) are unmarked , then

$(a,0) = \begin{bmatrix} b \\ a \end{bmatrix}$ $\quad (a,1) = \begin{bmatrix} c \\ d \end{bmatrix}$
$(b,0)$ $\quad (b,1)$

↓ unmarked ↓ unmarked

⟹ If anyone of the State is marked then (a,b) is marked.

[b,a] and [c,d] are unmarked. Hence [a,b] is unmarked. (a,b) → unmarked.

*(a,f) are unmarked, then

$(a,0) = \begin{bmatrix} b \\ f \end{bmatrix}$ $\quad (a,1) = \begin{bmatrix} c \\ f \end{bmatrix}$
$(f,0)$ $\quad (f,1)$

↓ unmarked ↓ marked

[b,f] is unmarked and [c,f] is marked. Hence [a,f] is marked. ⟹ [a,f] → ✓

* (b,f) are unmarked, then

$(b,0) = \begin{bmatrix} a \\ f \end{bmatrix}$ $\quad (b,1) = \begin{bmatrix} d \\ f \end{bmatrix}$
$(f,0)$ $\quad (f,1)$

↓ unmarked ↓ marked

[a,f] is unmarked and [d,f] is marked. Hence [b,f] is marked. ⟹ [b,f] → ✓

* (c,d) are unmarked, then

$(c,0) = \begin{bmatrix} e \\ e \end{bmatrix}$ $\quad (c,1) = \begin{bmatrix} f \\ f \end{bmatrix}$
$(d,0)$ $\quad (d,1)$

[e,e] and [f,f] both are unmarked.

Then [c,d] → unmarked.

* (c,e) is unmarked

(c,0) = e $\quad$ (c,1) = f
(e,0) = e $\quad$ (e,1) = f

Both are unmarked

Then (c,e) → unmarked

* (d,e) is unmarked, then
  (d,0) = e   (d,1) = f
  (e,0) = e   (e,1) = f

Both are unmarked
Then (d,e) ⟶ unmarked.

	a	b	c	d	e	f
a	–	–	–	–	–	–
b	unmarked	–	–	–	–	–
c	✓	✓	–	–	–	–
d	✓	✓	U	–	–	–
e	✓	✓	U	U	–	–
f	✓	✓	✓	✓	✓	–

U ⟶ unmarked

Step 4:-

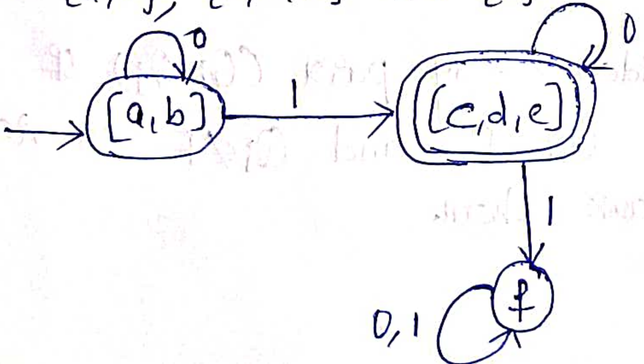
(a,b) (c,d) (c,e) (d,e) are unmarked. Leave it as unmarked states.

Now
(a,b) ⟹ a and b are equivalent then it make as only one state. [a,b]

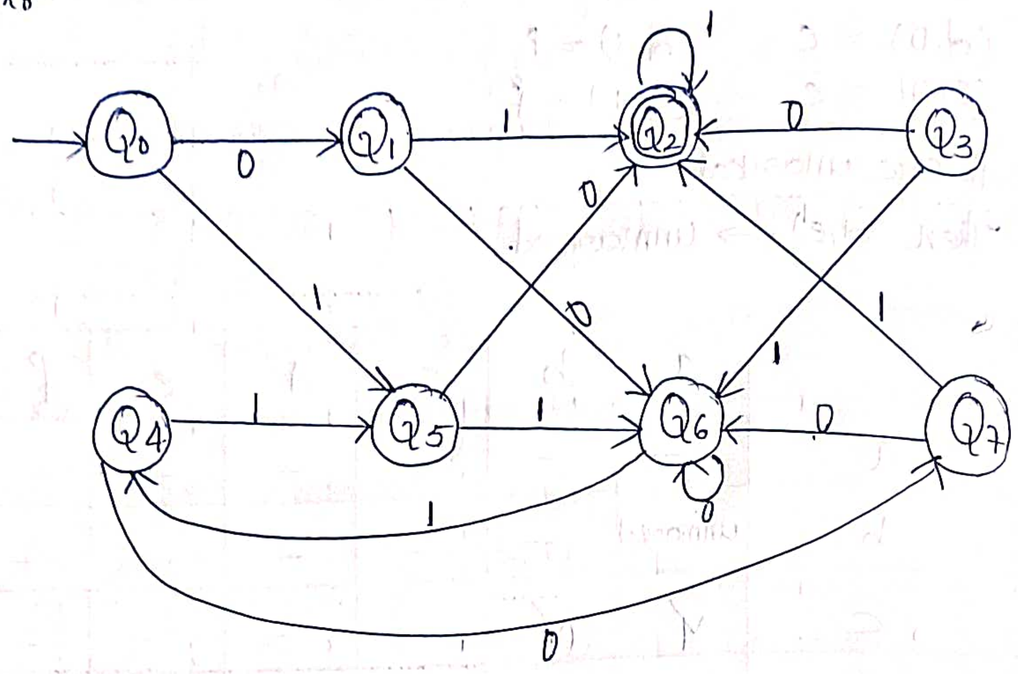
(c,e) (d,e) (c,e) are equivalent then it make as only one state

Hence we got two combined states as {a,b} and {c,d,e}

So, the final minimized DFA will contain three states {a,b}, {c,d,e} and {f}



Ex:-



For given DFA

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

$Q = \{Q_0, Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7\}$

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

$q_0 = Q_0$

$F = \{Q_2\}$

Step-1:- We draw a table for all pairs of states

(Q_0, Q_1)

(Q_0, Q_2) (Q_1, Q_2)

(Q_0, Q_3) (Q_1, Q_3) (Q_2, Q_3)

(Q_0, Q_4) (Q_1, Q_4) (Q_2, Q_4) (Q_3, Q_4)

(Q_0, Q_5) (Q_1, Q_5) (Q_2, Q_5) (Q_3, Q_5) (Q_4, Q_5)

(Q_0, Q_6) (Q_1, Q_6) (Q_2, Q_6) (Q_3, Q_6) (Q_4, Q_6) (Q_5, Q_6)

(Q_0, Q_7) (Q_1, Q_7) (Q_2, Q_7) (Q_3, Q_7) (Q_4, Q_7) (Q_5, Q_7) $(Q_6,$

Step-2:- Consider every pair (Q_i, Q_j) of DFA where $Q_i \in F$ and $Q_j \notin F$ or vice versa and mark them.

	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
Q_0	–	–	–	–	–	–	–	–
Q_1		–	–	–	–	–	–	–
Q_2	✓	✓	–	–	–	–	–	–
Q_3			✓	–	–	–	–	–
Q_4			✓		–	–	–	–
Q_5			✓			–	–	–
Q_6			✓				–	–
Q_7			✓					–

Step-3:- Repeat step2 untill we cannot mark any-more states.

* (Q_0, Q_1) are unmarked, then

$(Q_0, 0) = Q_1$ $(Q_0, 1) = Q_5$

$(Q_1, 0) = Q_6$ $(Q_1, 1) = Q_2$

$[Q_1, Q_6]$ is unmarked and $[Q_5, Q_2]$ is marked.

Hence $[Q_0, Q_1] \rightarrow$ marked

* (Q_0, Q_3) are unmarked, then

$(Q_0, 0) = Q_1$ $(Q_0, 1) = Q_5$

$(Q_3, 0) = Q_2$ $(Q_3, 1) = Q_6$

$[Q_1, Q_2]$ is marked and $[Q_5, Q_6]$ is unmarked

Hence $[Q_0, Q_3] \rightarrow$ marked.

* (Q_0, Q_4) are unmarked, then

$(Q_0, 0) = Q_1$ $(Q_0, 1) = Q_5$

$(Q_4, 0) = Q_7$ $(Q_4, 1) = Q_5$

$[Q_1, Q_7]$ is unmarked and $[Q_5, Q_5]$ is unmarked.

Then $[Q_0, Q_4]$ is unmarked.

* (Q_0, Q_5) is unmarked, then

$(Q_0, 0) = Q_1$ $(Q_0, 1) = Q_5$

$(Q_5, 0) = Q_2$ $(Q_5, 1) = Q_6$

$[Q_1, Q_2]$ is marked and $[Q_5, Q_6]$ is unmarked. Hence $[Q_0, Q_5]$ is marked.

* (Q_0, Q_6) is unmarked, then

$(Q_0, 0) = Q_1$ $(Q_0, 1) = Q_5$

$(Q_6, 0) = Q_6$ $(Q_6, 1) = Q_4$

$[Q_1, Q_6]$ is unmarked and $[Q_5, Q_4]$ is unmarked. Hence $[Q_0, Q_6]$ is unmarked.

* (Q_0, Q_7) is unmarked, then

$(Q_0, 0) = Q_1$ $(Q_0, 1) = Q_5$

$(Q_7, 0) = Q_6$ $(Q_7, 1) = Q_2$

$[Q_1, Q_6]$ is unmarked and $[Q_5, Q_2]$ is marked. Hence $[Q_0, Q_7]$ is marked.

* (Q_1, Q_3) is unmarked, then

$(Q_1, 0) = Q_6$ $(Q_1, 1) = Q_2$

$(Q_3, 0) = Q_2$ $(Q_3, 1) = Q_6$

$[Q_6, Q_2]$ is marked and $[Q_2, Q_6]$ is marked. Hence $[Q_1, Q_3]$ is marked.

* (Q_1, Q_4) is unmarked, then

$(Q_1, 0) = Q_6$ $(Q_1, 1) = Q_2$

$(Q_4, 0) = Q_7$ $(Q_4, 1) = Q_5$

$[Q_6, Q_7]$ is unmarked and $[Q_2, Q_5]$ is marked. Hence $[Q_1, Q_4]$ is marked.

* (Q_1, Q_5) is unmarked, then

$(Q_1, 0) = Q_6$ $\quad$ $(Q_1, 1) = Q_2$

$(Q_5, 0) = Q_2$ $\quad$ $(Q_5, 1) = Q_6$

$[Q_6, Q_2]$ is marked and $[Q_2, Q_6]$ is marked.
Then $[Q_1, Q_5]$ is marked.

* (Q_1, Q_6) is unmarked, then

$(Q_1, 0) = Q_6$ $\quad$ $(Q_1, 1) = Q_2$

$(Q_6, 0) = Q_6$ $\quad$ $(Q_6, 1) = Q_4$

$[Q_6, Q_6]$ is unmarked and $[Q_2, Q_4]$ is marked.
Then $[Q_1, Q_6]$ is marked.

* (Q_1, Q_7) is unmarked, then

$(Q_1, 0) = Q_6$ $\quad$ $(Q_1, 1) = Q_2$

$(Q_7, 0) = Q_6$ $\quad$ $(Q_7, 1) = Q_2$

$[Q_6, Q_6]$ is unmarked and $[Q_2, Q_2]$ is unmarked.
Then $(Q_1, Q_7]$ is unmarked.

* (Q_3, Q_4) is unmarked, then

$(Q_3, 0) = Q_2$ $\quad$ $(Q_3, 1) = Q_6$

$(Q_4, 0) = Q_7$ $\quad$ $(Q_4, 1) = Q_5$

$[Q_2, Q_7]$ is marked and $[Q_6, Q_5]$ is unmarked
Then $[Q_3, Q_4]$ is marked.

* (Q_3, Q_5) is unmarked, then

$(Q_3, 0) = Q_2$ $\quad$ $(Q_3, 1) = Q_6$

$(Q_5, 0) = Q_2$ $\quad$ $(Q_5, 1) = Q_6$

$[Q_2, Q_2]$ is unmarked and $[Q_6, Q_6]$ is unmarked.
Then $[Q_3, Q_5]$ is unmarked.

* (Q_3, Q_6) is unmarked, then

$(Q_3, 0) = Q_2$ $\quad$ $(Q_3, 1) = Q_6$

$(Q_6, 0) = Q_6$ $\quad$ $(Q_6, 1) = Q_4$

$[Q_2, Q_6]$ is marked and $[Q_6, Q_4]$ is unmarked. Hence $[Q_3, Q_6]$ is marked.

* (Q_3, Q_7) is unmarked, then

$(Q_3, 0) = Q_2$ $\quad$ $(Q_3, 1) = Q_6$

$(Q_7, 0) = Q_6$ $\quad$ $(Q_7, 1) = Q_2$

$[Q_2, Q_6]$ is marked and $[Q_6, Q_2]$ is marked. Then $[Q_3, Q_7]$ is marked.

* (Q_4, Q_5) is unmarked, then

$(Q_4, 0) = Q_7$ $\quad$ $(Q_4, 1) = Q_5$

$(Q_5, 0) = Q_2$ $\quad$ $(Q_5, 1) = Q_6$

$[Q_7, Q_2]$ is marked and $[Q_5, Q_6]$ is unmarked. Then $[Q_4, Q_5]$ is marked.

* (Q_4, Q_6) is unmarked, then

$(Q_4, 0) = Q_7$ $\quad$ $(Q_4, 1) = Q_5$

$(Q_6, 0) = Q_6$ $\quad$ $(Q_6, 1) = Q_4$

$[Q_7, Q_6]$ is unmarked and $[Q_5, Q_6]$ is unmarked. Then $[Q_4, Q_6]$ is unmarked.

* (Q_4, Q_7) is unmarked, then

$(Q_4, 0) = Q_7$ $\quad$ $(Q_4, 1) = Q_5$

$(Q_7, 0) = Q_6$ $\quad$ $(Q_7, 1) = Q_2$

$[Q_7, Q_6]$ is unmarked. and $[Q_5, Q_2]$ is marked. Then $[Q_4, Q_7]$ is marked.

* (Q_5, Q_6) is unmarked, then

$(Q_5, 0) = Q_2$ $\quad$ $(Q_5, 1) = Q_6$

$(Q_6, 0) = Q_6$ $\quad$ $(Q_6, 1) = Q_4$

$[Q_2, Q_6]$ is marked and $[Q_6, Q_4]$ is unmarked. Then $[Q_5, Q_6]$ is marked.

* (Q_5, Q_7) is unmarked, then

$(Q_5, 0) = Q_2$ $\quad$ $(Q_5, 1) = Q_6$

$(Q_7, 0) = Q_6$ $\quad$ $(Q_7, 1) = Q_2$

$[Q_2, Q_6]$ is marked and $[Q_6, Q_2]$ is marked. Then $(Q_5, Q_7]$ is marked.

* $[Q_6, Q_7)$ is unmarked, then

$(Q_6, 0) = Q_6$ $\quad$ $(Q_6, 1) = Q_4$

$(Q_7, 0) = Q_6$ $\quad$ $(Q_7, 1) = Q_2)$

$[Q_6, Q_6]$ is unmarked and $[Q_4, Q_2]$ is marked. Then $(Q_6, Q_7]$ is marked.

	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
Q_0	–	–	–	–	–	–	–	–
Q_1	✓	–	–	–	–	–	–	–
Q_2	✓	✓	–	–	–	–	–	–
Q_3	✓	✓	✓	–	–	–	–	–
Q_4	U	✓	✓	✓	–	–	–	–
Q_5	✓	✓	✓	U	✓	–	–	–
Q_6	U	✓	✓	✓	✓	✓	–	–
Q_7	✓	U	✓	✓	✓	✓	✓	–

Step-4:-

(Q_0, Q_4) (Q_0, Q_6) (Q_1, Q_7) (Q_3, Q_5) (Q_4, Q_6)

* (Q_0, Q_4) is unmarked then

$(Q_0, 0) = Q_1$ $\quad$ $(Q_0, 1) = Q_5$

$(Q_4, 0) = Q_7$ $\quad$ $(Q_4, 1) = Q_5$

(Q_1, Q_7) is unmarked then (Q_0, Q_4) is unmarked.

(Q_0, Q_6) is unmarked, then

$(Q_0, 0) = Q_1$ $\quad$ $(Q_0, 1) = Q_5$

$(Q_6, 0) = Q_6$ $\quad$ $(Q_6, 1) = Q_4$

(Q_1, Q_6) is marked and (Q_5, Q_4) is unmarked. Then

(Q_0, Q_6) is marked

* (Q_1, Q_7) is unmarked, then

$(Q_1, 0) = Q_6$ $\quad$ $(Q_1, 1) = Q_2$

$(Q_7, 0) = Q_6$ $\quad$ $(Q_7, 1) = Q_2$

(Q_6, Q_6) is unmarked and (Q_2, Q_2) is unmarked. Then

$[Q_1, Q_7]$ is unmarked.

* (Q_3, Q_5) is unmarked, then

$(Q_3, 0) = Q_2$ $\quad$ $(Q_3, 1) = Q_6$

$(Q_5, 0) = Q_2$ $\quad$ $(Q_5, 1) = Q_6$

(Q_2, Q_2) is unmarked and (Q_6, Q_6) is unmarked. Then

(Q_3, Q_5) is unmarked.

* (Q_4, Q_6) is unmarked, then

$(Q_4, 0) = Q_7$ $\quad$ $(Q_4, 1) = Q_5$

$(Q_6, 0) = Q_6$ $\quad$ $(Q_6, 1) = Q_4$

~~Both~~ (Q_7, Q_6) is marked and then (Q_4, Q_6) is marked.

	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
Q_0	—	—	—	—	—	—	—	—
Q_1	✓	—	—	—	—	—	—	—
Q_2	✓	✓	—	—	—	—	—	—
Q_3	✓	✓	✓	—	—	—	—	—
Q_4	✓	✓	✓	✓	—	—	—	—
Q_5	✓	✓	✓	U	✓	—	—	—
Q_6	✓	✓	✓	✓	✓	✓	—	—
Q_7	✓	U	✓	✓	✓	✓	✓	—

Again Compare

* (Q_0, Q_4) is unmarked, then

$(Q_0, 0) = Q_1$ $(Q_0, 1) = Q_5$

$(Q_4, 0) = Q_7$ $(Q_4, 1) = Q_5$

Both are unmarked then $(Q_0, Q_4) \rightarrow$ unmarked

* (Q_1, Q_7)

$(Q_1, 0) = Q_2$ $(Q_1, 1) = Q_2$

$(Q_7, 0) = Q_6$ $(Q_7, 1) = Q_2$

Both are unmarked then $(Q_1, Q_7) \rightarrow$ unmarked

* (Q_3, Q_5)

$(Q_3, 0) = Q_2$ $(Q_3, 1) = Q_6$

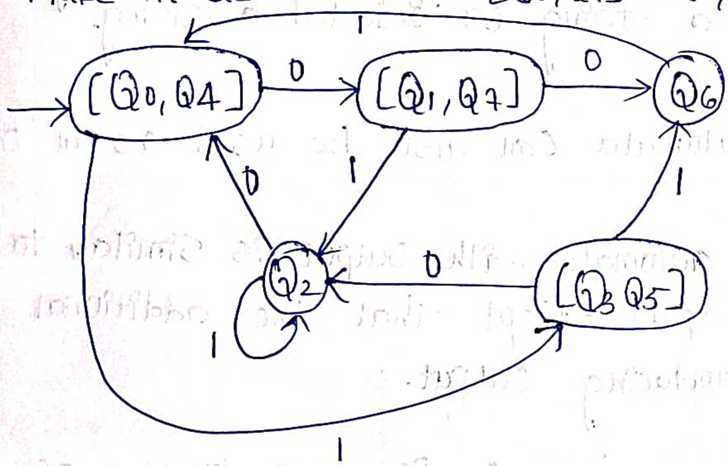
$(Q_5, 0) = Q_2$ $(Q_5, 1) = Q_6$

Both are unmarked then $(Q_3, Q_5) \rightarrow$ unmarked

Step-5:- Combine all unmarked States

(Q_0, Q_1) (Q_1, Q_7) (Q_3, Q_5)

Make it as one state $[Q_0, Q_1]$. $[Q_1, Q_7]$



24/1/22
Monday

EQUIVALENCE OF FINITE AUTOMATA

⇒ Two automata M_0 and M_1 are said to be equivalent to each other if both accept exactly the same set of input strings. Formally, if two automata M_0 and M_1 are equivalent then.

⇒ If there is a path from the start state of M_0 to a final state of M_0 labeled $M_{01} M_{02} \ldots M_{0K}$, there is a path from the start state of M_1 to a final state of M_1 labeled $M_{11}, M_{12}, \ldots M_{1K}$

⇒ If there is a path from the start state of M_1 to a final state of M_1 labeled $M_{11}, M_{12} \ldots M_{1K}$, there is a path from the start state of M_0 to a final state of M_0 labeled $M_{01}, M_{02} \ldots M_{0K}$.

Finite Automata with Output

- Finite automata have a limited capacity of either accepting a string or rejected a string.
- Finite automata can also be used as an output device.
- A finite automata with output is similar to finite automata (FA) except that the additional capability of producing output.
- In a formal way it is also known as Finite State Machine (FSM) or Transducer.
- Two types of FSM
  - Moore Machine
  - Mealy Machine

Definition – FSM

Finite automata with output machine M is defined by 6-tuples are as follows;

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

Where

Q: Finite non-empty set of states

Σ: Finite non--empty set of input alphabet.

δ: State transition function.

Δ: Output alphabet

λ: Output function. denotes as $Q \rightarrow \Delta$

q_0: Initial state (which may be fixed or variable depends on machine behavior).

δ: Takes the current state from Q and an input alphabet from Σ and returns the new set of output alphabets and the next state. Therefore, it can be seen as a function which maps on ordered sequence, or of input alphabets into a corresponding sequence, or set, of output events.

$Q \times \Sigma \rightarrow Q$ is the next-state function.

Characteristics

- Finite automata with output machines do not have final states / state.

- Machine generates an output on every input. The value of the output is a function of current state and the current input. λ output λ $q \in Q$, ip

- Finite automata with output machines are characterized by two behaviors;

  – State transaction function (δ) [State transition function (δ) is also known as STF]

  – Output Function (λ) [Output function is also known as Machine Function (MAF)]

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

$\lambda : \Sigma \times Q \rightarrow O$ [For Mealy Machine]

$\lambda : Q \rightarrow O$ [For Moore Machine]

Mealy Machine: It depends output on Current State and input Symbol.

Moore Machine: It depends output only Current State.

Moore Machine

- Moore Machine is a finite state machine in which the next state is decided by the Current State and Current input Symbol.
- The output symbol at a given time depends only on the present state of the machine.

$$\lambda : Q \rightarrow O$$

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

Q : finite set of states

q_0 : initial state of machine

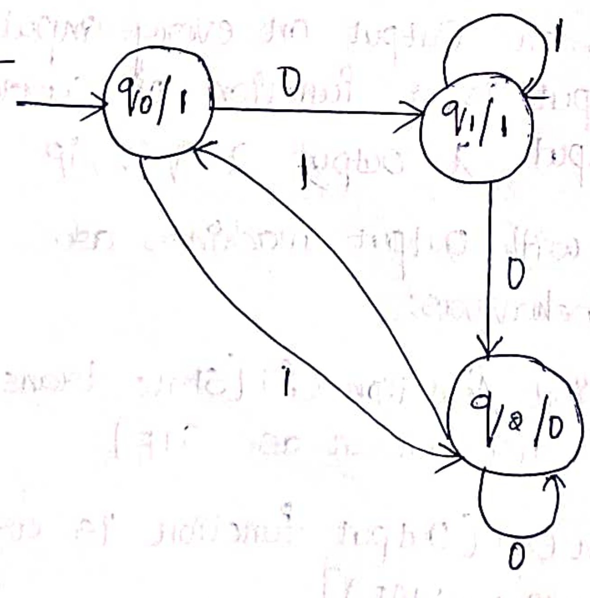
Σ : finite set of input symbols.

O : Output alphabet [Δ or O]

δ : transition function where $Q \times \Sigma \rightarrow Q$

λ : Output function, $Q \rightarrow O$

Ex:-



For this

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

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

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

$q_0 = q_0$

$O = \{0,1\}$

δ λ:

Current State	Next State (δ)		Output λ
	0	1	
q_0	q_1	q_2	1
q_1	q_2	q_1	1
q_2	q_2	q_0	0

- Output is represented with each input state separated by /
- Output ~~is~~ length for a Moore machine is greater than input by 1.

Input:- 010

Transition: $q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_2$

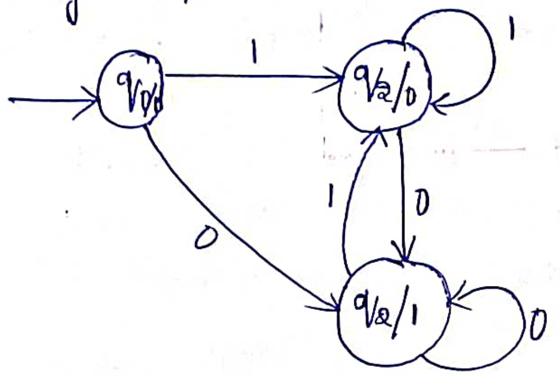
1 1 1 0

Output:- 1110 [1 for q_0, 1 for q_1, again 1 for q_1, 0 for q_2]

Ex:- Design a Moore Machine to generate 1's Complement of a given binary number.

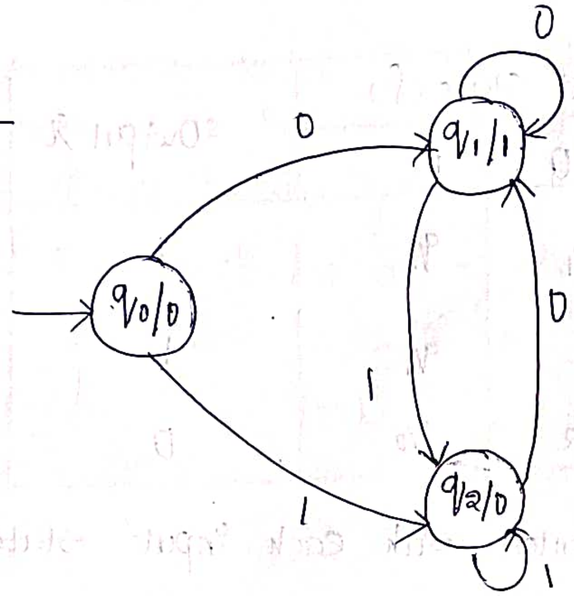
To generate 1's Complement of a given binary number the simple logic is that if the input is 0 then the output will be 1 and if the input is 1 then the output will be 0. That means there are three states. One state is start state. The second state is for taking 0's as input and produces output as 1.

The third State is for taking 1's as input and producing output as 0.



Ex:- 110

Ex:-



For given

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

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

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

$q_0 = q_0$

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

$\delta : \lambda$

δ:

Current State	Next State 0	Next State 1	(λ) Output
$\rightarrow q_0$	q_1	q_2	0
q_1	q_1	q_2	1
q_2	q_1	q_2	0

Input		1	0	1	1
State	q_0	q_2	q_1	q_2	q_2
Output	0	0	1	0	0

Input: 1011

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

0 0 1 0 0

Ex:- Construct a Moore machine that determines whether an input string contains an even or odd number's of 1's. The machine should give 1 as output if an even number of 1's are in the string and 0 otherwise.

In the process of taken input string, the total no. of one's is in even number then it treat as 1 otherwise 0. See one Example

If input is 1010

1 0 1 0

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

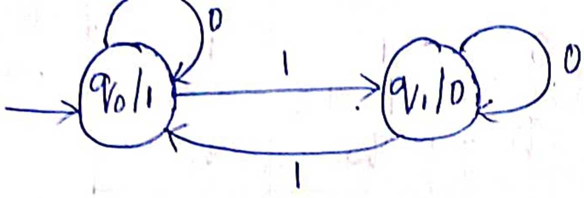
1 0 0 1 1

Here we first take 1 as input there is only one '1' so output is 0 and take 0 as input there is only one '1' in the process so output is 0 and take 1 as input here two '1's that means in even number so we get 1 and 0 taken as input already even no. of 1's we get 1 as output. Finally 100011 0 is first we taken as like that.

1 0 0 0 1

↓ ↓ ↓ ↓ ↓

1 0 0 0 0 1



Input: 11001

↓↓↓↓↓↓

101110

We Expected

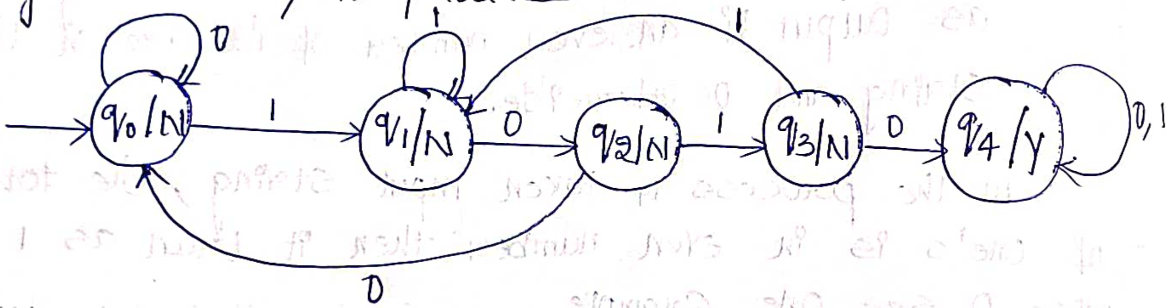
$q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1$

1 0 1 1 1 0

We get

Ex:-

Design a Moore machine with the input alphabet {0,1} and output alphabet {Y,N}, which produces Y as output if input sequence contains 1010 as a substring otherwise, it produces N as output.



101101

↓↓↓↓↓↓

NNNNNNN

Because we can't see substring as 1010

1011010101011

↓↓↓↓↓↓↓↓↓↓↓↓↓

NNNNNNNNNYYYY

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

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

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

$\Delta = \{Y, N\}$

$q_0 = q_0$

$\delta : \lambda$

Current State	Next State		Output(λ)
	0	1	
q_0	q_0	q_1	N
q_1	q_2	q_1	N
q_2	q_0	q_3	N
q_3	q_4	q_1	N
q_4	q_4	q_4	Y

n ip

more machine $1+n+1$

mealy machine $1+2(n)$.

Columns Here $n = 2$, For More machine $1+2+1$
$= 4$

$1+n+1$

Present State, Next State, Output

For mealy machine

Present	Next State		Output	
	0	1	0	1

27/1/22
Thursday

Mealy Machine

- A machine in which output symbol depends upon the present input symbol and present state of the machine.
- The output is represented with each input symbol for each state separated by /. The mealy machine can be described by 6 tuples $(Q, q_0, \Sigma, O, \delta, \lambda')$

Where

Q: finite set of States

q_0: Initial State of machine

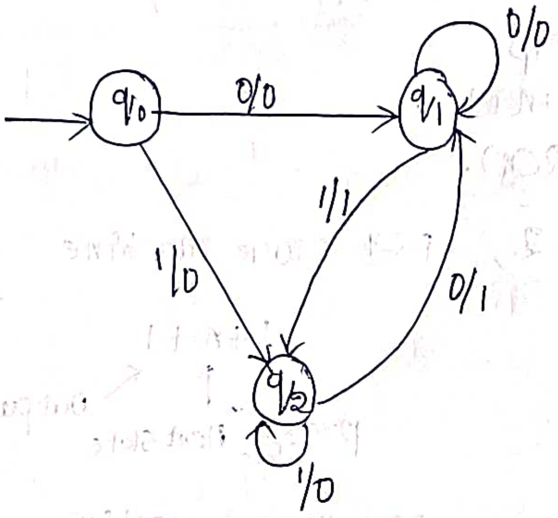
Σ: Finite Set of input alphabet

O: Output alphabet

δ: transition function where $Q \times \Sigma \rightarrow Q$

λ: Output function Where $Q \times \Sigma \rightarrow O$

Ex:-



$M = (Q, \Sigma, q_0, O, \delta, \lambda)$

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

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

$q_0 = q_0$

$O = \{0, 1\}$

δ and λ

	Input:0		Input:1	
present State	Next State	Output	NextState	Output
q_0	q_1	0	q_2	0
q_1	q_1	0	q_2	1
q_2	q_1	1	q_2	0

• Output is represented with each input symbol for each state separated by /

• The length of output is equal to the length of input

Input: 11

Transition: $q_0 \xrightarrow{1} q_2 \xrightarrow{1} q_2$

0 0

Output : 00 [q_0 to q_2 transition has output 0 and q_2 to q_2 transition also has output 0]

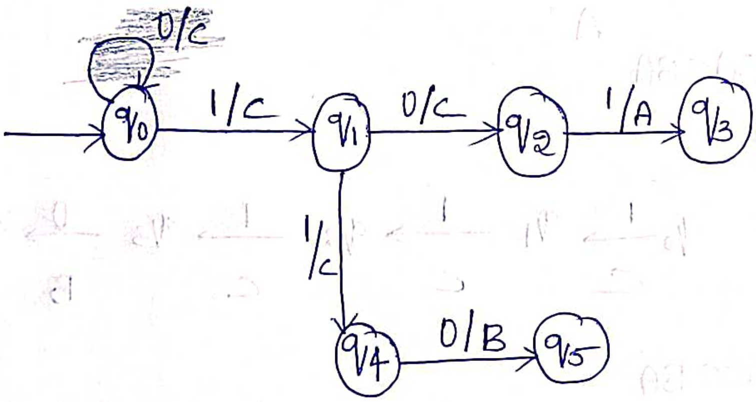
Input: 0101

Transition: $q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{1} q_2$

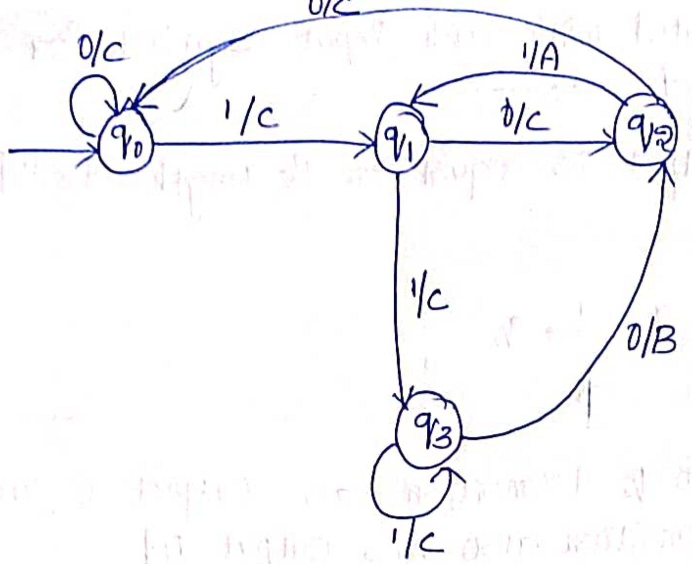
0 1 1 1

Output: 0111

Ex:- Design a Mealy Machine for a binary input sequence such that if it has a substring 101, the machine ~~110~~, ~~it inputs B~~ ~~o~~ output A, if the input has substring 110, it ~~in~~ outputs B otherwise it outputs C.



For designing such a machine, we will check two conditions, and those are 101 and 110. If we get 101, the output will be A. If we recognize 110, the output will be B. For other strings the output will be C.



Input:- 1010101

Transition:- $q_0 \xrightarrow[C]{1} q_1 \xrightarrow[C]{0} q_2 \xrightarrow[A]{1} q_1 \xrightarrow[C]{0} q_2 \xrightarrow[A]{1} q_1 \xrightarrow[C]{0} q_2 \xrightarrow[A]{1} q_1$

CCACACA

Output:- CCACACA

Input:- 101101

Transition:- $q_0 \xrightarrow[C]{1} q_1 \xrightarrow[C]{0} q_2 \xrightarrow[A]{1} q_1 \xrightarrow[C]{1} q_3 \xrightarrow[B]{0} q_2 \xrightarrow[A]{1} q_1$

Output:- CCACBA

Input:- 11101

transition:- $q_0 \xrightarrow[C]{1} q_1 \xrightarrow[C]{1} q_3 \xrightarrow[C]{1} q_3 \xrightarrow[B]{0} q_2 \xrightarrow[A]{1} q_1$

Output:- CCCBA

$M = (Q, \Sigma, q_0, O, \lambda, \delta)$

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

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

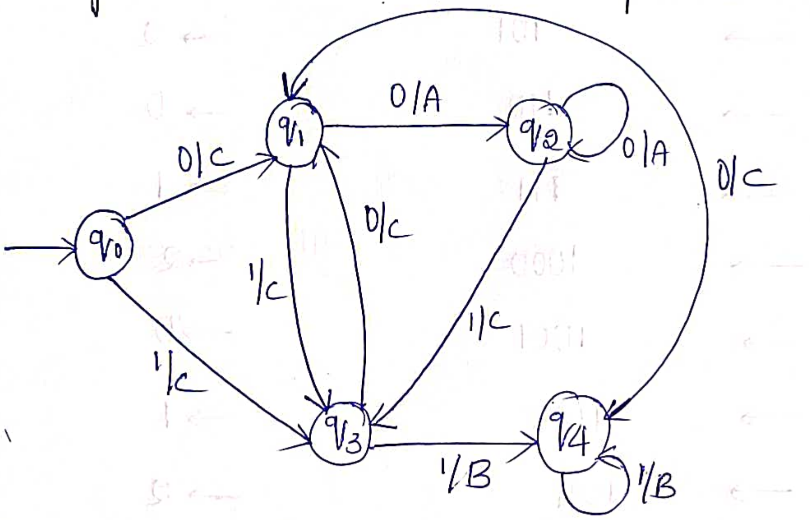
$q_0 = q_0$

$O = \{0,1\}$

δ and λ

Present State	Input: 0		Input: 1	
	Next State	Output	Next State	Output
$\rightarrow q_0$	q_0	C	q_1	C
q_1	q_2	C	q_3	C
q_2	q_0	C	q_1	A
q_3	q_2	B	q_3	C

Ex:- Design a Mealy machine that scans sequence of input of 0 and 1 and generates output 'A' if the input string terminates in 00, output 'B' if the string terminates in 11, and output 'C' otherwise.



$M = (Q, \Sigma, \delta, \lambda, O, q_0)$

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

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

$q_0 = q_0$

$O = \{A, B, C\}$

Present State	Input: 0		Input: 1	
	Next State	Output	Next State	Output
q_0	q_1	C	q_3	C
q_1	q_2	A	q_3	C
q_2	q_2	A	q_3	C
q_3	q_1	C	q_4	B
q_4	q_1	C	q_4	B

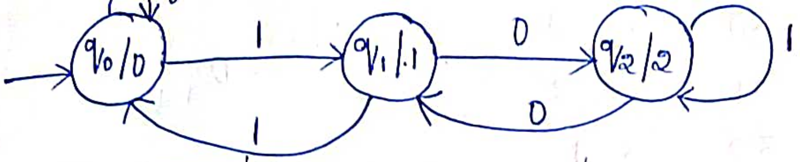
Construct Moore machine which takes binary String as input and produce mod 3 recidue as output.

$\Sigma = \{0,1\}$ $O = \{0,1,2\}$

Dec i/p		binary string	mod-3
0	→	0	→ 0
1	→	1	→ 1
2	→	10	→ 2
3	→	11	→ 0
4	→	100	→ 1
5	→	101	→ 2
6	→	110	→ 0
7	→	111	→ 1
8	→	1000	→ 2
9	→	1001	→ 0
10	→	1010	→ 1
11	→	1011	→ 2
12	→	1100	→ 0
13	→	1101	→ 1
14	→	1110	→ 2
15	→	1111	→ 0
16	→	10000	→ 1
17	→	10001	→ 2
18	→	10010	→ 0
19	→	10011	→ 1

Input:- 101011

Output:- 122101



present state	δ: Next state		Output
	0	1	
q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2

$M = (Q, \Sigma, O, \delta, \lambda, q_0)$

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

$O = \{0, 1, 2\}$

$q_0 = q_0$

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

Construct a Moore Machine which takes base 4 input and produces residue modulo5 as o/p.

$\Sigma = \{0,1,2,3\}$ $O = \{0,1,2,3,4\}$

Base '4' Dec		'4' Base		%5
0	$\rightarrow$	0	$\rightarrow$	0
1	$\rightarrow$	1	$\rightarrow$	1
2	$\rightarrow$	2	$\rightarrow$	2
3	$\rightarrow$	3	$\rightarrow$	3
4	$\rightarrow$	10	$\rightarrow$	4
5	$\rightarrow$	11	$\rightarrow$	0
6	$\rightarrow$	12	$\rightarrow$	1
7	$\rightarrow$	13	$\rightarrow$	2
8	$\rightarrow$	20	$\rightarrow$	3
9	$\rightarrow$	21	$\rightarrow$	4
10	$\rightarrow$	22	$\rightarrow$	0
11	$\rightarrow$	23	$\rightarrow$	1

$4 \underline{|4}$ $1 - 0 \rightarrow (10)_4$

$4 \underline{|5}$ $1 - 1 \rightarrow (11)_4$

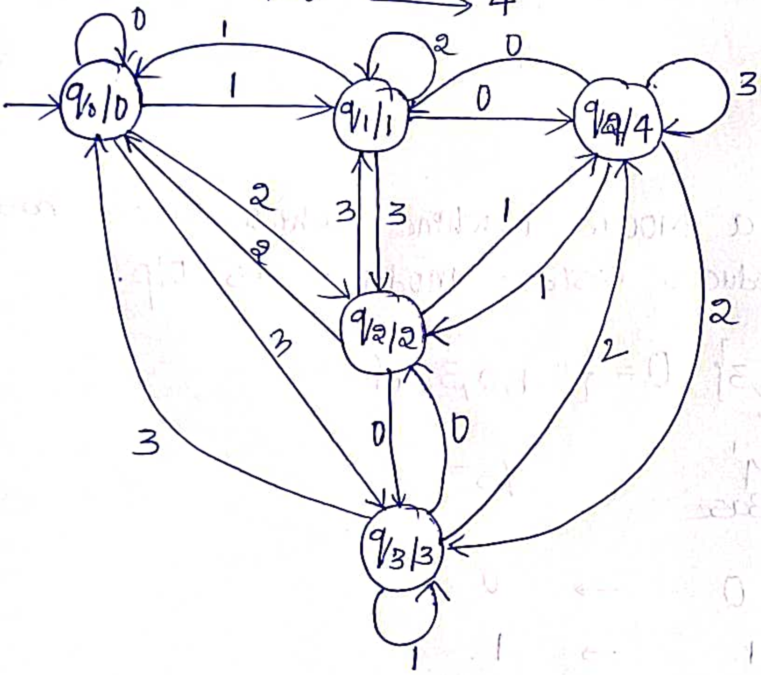
$4 \underline{|6}$ $1 - 2$ $(12)_4$

$4 \underline{|10}$ $2 - 2 \rightarrow (22)_4$

12	$\longrightarrow$	30	$\longrightarrow$	2
13	$\longrightarrow$	31	$\longrightarrow$	3
14	$\longrightarrow$	32	$\longrightarrow$	4
15	$\longrightarrow$	33	$\longrightarrow$	0
16	$\longrightarrow$	100	$\longrightarrow$	1
17	$\longrightarrow$	101	$\longrightarrow$	2
18	$\longrightarrow$	102	$\longrightarrow$	3
19	$\longrightarrow$	103	$\longrightarrow$	4
20	$\longrightarrow$	110	$\longrightarrow$	0
21	$\longrightarrow$	111	$\longrightarrow$	1
22	$\longrightarrow$	112	$\longrightarrow$	2
23	$\longrightarrow$	113	$\longrightarrow$	3
24	$\longrightarrow$	120	$\longrightarrow$	4

$4 \lfloor 12$
$3 - 0$

$4 \lfloor 17$
$4 \lfloor 4 - 1$
$1 - 0 \Rightarrow (101)_4$



$M = (Q, \Sigma, \delta, \lambda, O, q_0)$

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

$\Sigma = \{0, 1, 2, 3\} \Rightarrow 'n'$

$O = \{0, 1, 2, 3, 4\}$

$q_0 = q_0$

$n = 4$

Present state	Next state				Output
	0	1	2	3	
q_0	q_0	q_1	q_2	q_3	0
q_1	q_4	q_0	q_1	q_2	1

	0	1	2	3	
q_2	q_3	q_0	q_0	q_1	2
q_3	q_2	q_3	q_4	q_0	3
q_4	q_1	q_2	q_3	q_4	4

29/1/22
Saturday

Conversion from Mealy machine to Moore machine

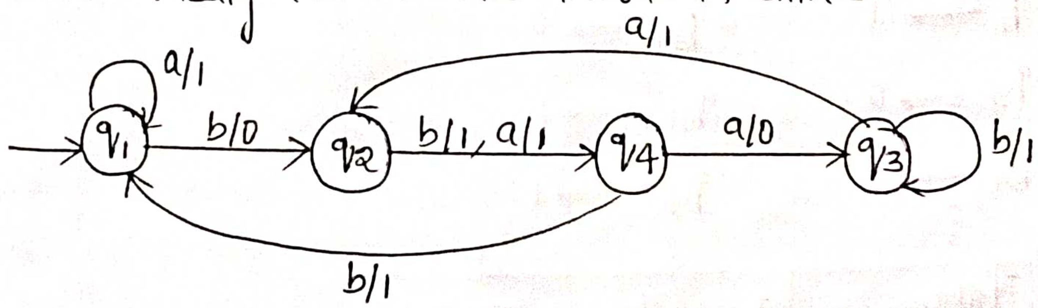
- Create a separate state for every new output symbol and according to incoming and outgoing edges are distributed.

Step-1:- For each state (Q_i), calculate the number of different outputs that are available in the transition table of the Mealy machine.

Step-2:- Copy state Q_i, if all the outputs of Q_i are the same. Break q_i into n states as Q_{in}, if it has n distinct outputs where $n = 0, 1, 2, \ldots$

$\Rightarrow$ We see the output symbol from state transition consider.

Convert Mealy Machine into Moore Machine



For given

$M = (Q, \Sigma, O, q_0, \delta, \lambda)$

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

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

$O = \{0, 1\}$

$q_0 = q_1$

Present State	Next-State			
	I/p: a		I/p: b	
	State	O/p(λ)	State	O/p(λ)
q_1	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	1	q_3	1
q_4	q_3	0	q_1	1

Here q_1 has two incoming edges one is from q_1 itself with Output '1' and another one is from q_4 with Output '1' so here the Output is same i.e., 1 so need not to split the states.

Now q_2 have two incoming edges one is from q_1 with output '0' and another one is from q_3 with Output '1'. so here to split the states because the symbol of output is different. i.e., $q_{20} \rightarrow 0$ and $q_{21} \rightarrow 1$

Ily q_3 and q_4 also check the process.

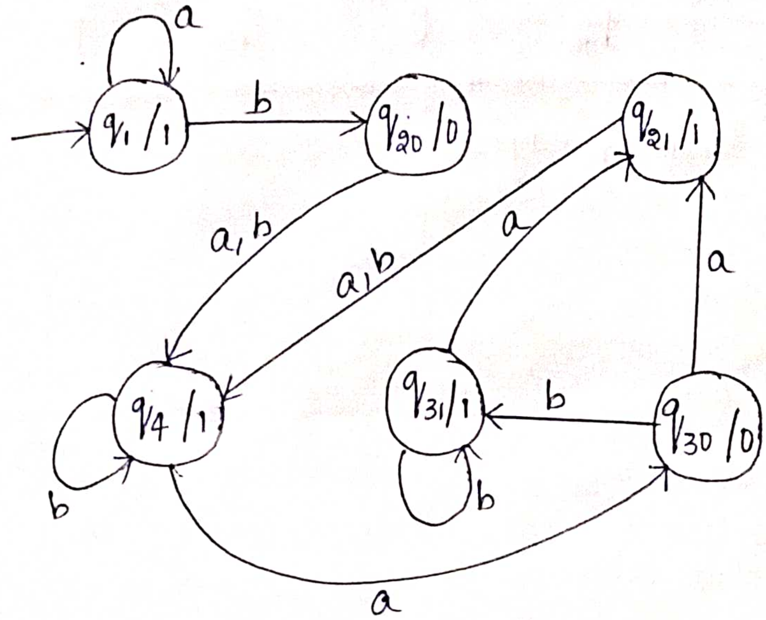
$q_1 \longrightarrow 1$ $\qquad$ $q_{30} \longrightarrow 0$

$q_{20} \longrightarrow 0$ $\qquad$ $q_{31} \longrightarrow 1$

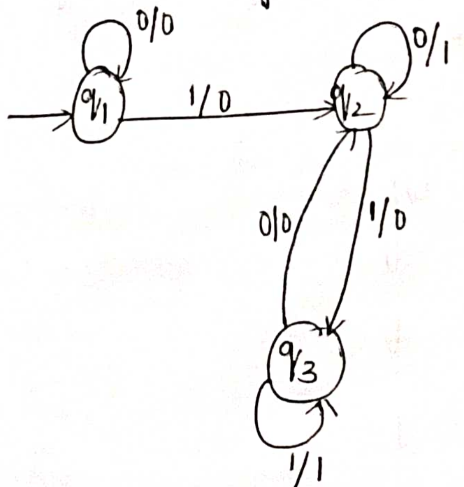
$q_{21} \longrightarrow 1$ $\qquad$ $q_4 \longrightarrow 1$

Table	a (δ)	b	Output (λ)
q_1	q_1	q_{20}	1
q_{20}	q_4	q_4	0
q_{21}	q_4	q_4	1
q_{30}	q_{21}	q_{31}	0
q_{31}	q_{21}	q_{31}	1
q_4	q_{30}	q_1	1

Moore Machine



Ex: 2 Convert Mealy machine to Moore machine



For given Mealy Machine

$M = (Q, \Sigma, O, \delta, \lambda, q_0)$

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

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

$O = \{0, 1\}$

$q_0 = q_1$

Present state	Next state			
	I/P: 0		I/P: 1	
	State	O/P (λ)	State	O/P (λ)
q_1	q_1	0	q_2	0

q_2	q_2	1	q_3	0
q_3	q_2	0	q_3	1

Check the incoming edges

$q_1 \longrightarrow 0$

$q_{20} \longrightarrow 0$

$q_{21} \longrightarrow 1$

$q_{30} \longrightarrow 0$

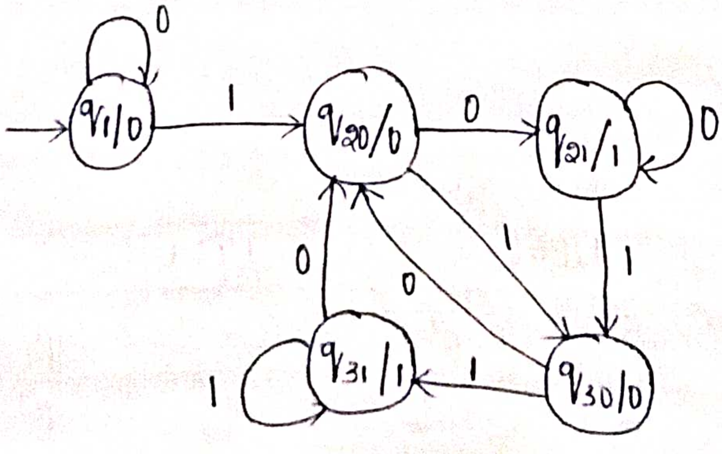
$q_{31} \longrightarrow 1$

Note:- When we see the incoming edges on transition State we see only output symbol. They need not to how many symbols.

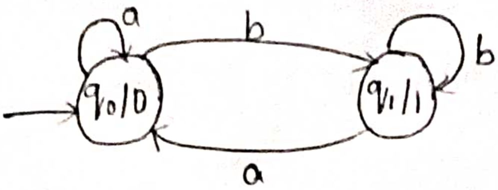
Transition Table:-

T	0	1	O/P (λ)
q_1	q_1	q_{20}	0
q_{20}	q_{21}	q_{30}	0
q_{21}	q_{21}	q_{30}	1
q_{30}	q_{20}	q_{31}	0
q_{31}	q_{20}	q_{31}	1

Moore Machine



Convert Moore Machine to Mealy Machine



For given Moore Machine

$M = (Q, \Sigma, O, \delta, \lambda, q_0)$

$Q = \{q_0, q_1\}$

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

$O = \{0, 1\}$

$q_0 = q_0$

Present State	Next State δ		Output (λ)
	a	b	
q_0	q_0	q_1	0
q_1	q_0	q_1	1

λ' is output of the State

We need to find the every combination of present State and Current input.

$\lambda'(q_0, a) = \lambda(\delta(q_0, a))$
$= \lambda(q_0)$
$= 0$

$\lambda'(q_0, b) = \lambda(\delta(q_0, b))$
$= \lambda(q_1)$
$= 1$

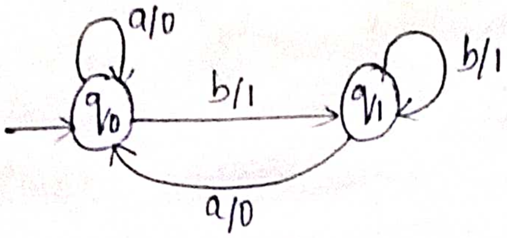
$\lambda'(q_1, a) = \lambda(\delta(q_1, a))$
$= \lambda(q_0)$
$= 0$

$\lambda'(q_1, b) = \lambda(\delta(q_1, b))$
$= \lambda(q_1)$
$= 1$

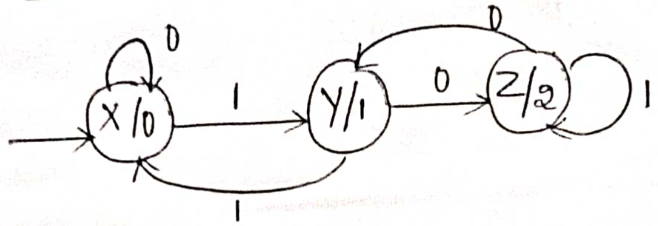
Transition Table

Present State	a		b	
	State	O/P	State	O/P
q_0	q_0	0	q_1	1
q_1	q_0	0	q_1	1

This transition Table is Equivalent to Mealy Machine



Ex:-2 Convert Moore Machine to Mealy Machine



For given Moore Machine

$M = (Q, \Sigma, O, \delta, \lambda, q_0)$

$Q = \{x, y, z\}$

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

$O = \{0, 1, 2\}$

$q_0 = x$

Present state	Next State (δ)		Output(λ)
	0	1	
x	x	y	0
y	z	x	1
z	y	z	2

$\lambda'(x,0) = \lambda(\delta(x,0))$
$= \lambda(x)$
$= 0$

$\lambda'(x,1) = \lambda(\delta(x,1))$
$= \lambda(y)$
$= 1$

$\lambda'(y,0) = \lambda(\delta(y,0))$
$= \lambda(z)$
$= 2$

$\lambda'(y,1) = \lambda(\delta(y,1))$
$= \lambda(x)$
$= 0$

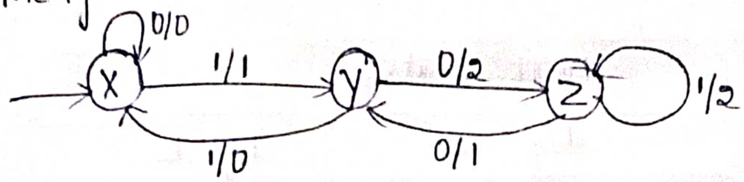
$\lambda'(z,0) = \lambda(\delta(z,0))$
$= \lambda(y)$
$= 1$

$\lambda'(z,1) = \lambda(\delta(z,1))$
$= \lambda(z)$
$= 2$

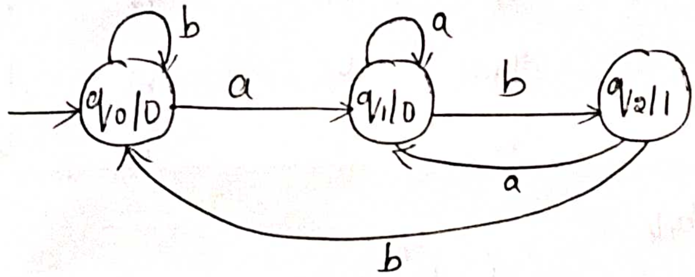
Transition Table

Present State	Next State			
	I/p: 0		I/p: 1	
X	State	o/p λ	State	o/p (λ)
X	X	0	Y	1
Y	Z	2	X	0
Z	Y	1	Z	2

Mealy Machine



Ex:- 3 Convert Moore Machine to Mealy Machine



For given Moore Machine

$M = (Q, \Sigma, O, \delta, \lambda, q_0)$

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

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

$O = \{0, 1\}$

$q_0 = q_0$

Present State	Next State		Output
	a	b	
q_0	q_1	q_0	0
q_1	q_1	q_2	0
q_2	q_1	q_0	1

$\lambda'(q_0, a) = \lambda(\delta(q_0, a))$
$= \lambda(q_1)$
$= 0$

$\lambda'(q_0, b) = \lambda(\delta(q_0, b))$
$= \lambda(q_0)$
$= 0$

$\lambda'(q_1, a) = \lambda(\delta(q_1, a))$
$= \lambda(q_1)$
$= 0$

$\lambda'(q_1, b) = \lambda(\delta(q_1, b))$
$= \lambda(q_2)$
$= 1$

$\lambda'(q_2, a) = \lambda(\delta(q_2, a))$
$= \lambda(q_1)$
$= 0$

$\lambda'(q_2, b) = \lambda(\delta(q_2, b))$
$= \lambda(q_0)$
$= 0$

Transition Table:-

Present State	Next State			
	I/p: a		I/p: b	
	State	O/p(λ)	State	O/p(λ)
q_0	q_1	0	q_0	0
q_1	q_1	0	q_2	1
q_2	q_1	0	q_0	0