Properties of Transition Functions

Property 1: $\delta^*(q, \lambda) = q$ in a finite automaton.

$|\lambda| = 0$ λ is NULL string

i.e, State of the system can be changed only by an input symbol

Property 2: For all string $w \in \Sigma^*$ and input symbol $a \in \Sigma$

$$\delta^*(q, wa) = \delta(\delta^*(q, w), a)$$

gives the state after the automaton consumes first symbol of string wa

recursive

OR $\delta^*(q, aw) = \delta^*(\delta(q, a), w)$

gives the state after the automaton consumes a prefix of the string wa

NOTE1: transition function $\delta : Q \times \Sigma \longrightarrow Q$

Extended transition function $\delta^* : Q \times \Sigma^* \longrightarrow Q$

δ (state, char) = nextstate

δ^* (state, string) $\overset{Nar}{=}$ state

goto page no 13

NOTE2: Σ is a set of alphabet (finite set.)

Σ^* denotes the set of all strings (including λ the empty string) over the alphabet set Σ

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

Example 1 Prove that for any transition function δ and for any two input alphabet ~~string~~ a, and b.

if $\delta(q_0, a) = q_1$ → (a)

and $\delta(q_1, b) = q_2$ → (b)

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

Proof According to transition function property 2,

$$\delta^*(q_0, ab) = \delta(\delta^*(q_0, a), b) \rightarrow (I)$$

but $\delta^*(q_0, a) = \delta(\delta^*(q_0, \lambda), a) \rightarrow (II)$

According to part 1

$$\delta^*(q_0, \lambda) = q_0 \rightarrow (III)$$

from eq (II)

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

$$= q_1 \quad \text{Given. (from eq (a))}$$

Substituting this in eq^n (I)

$$\delta^*(q_0, ab) = \delta(q_1, b)$$

$$\delta^*(q_0, ab) = q_2 \quad \text{given in eq (b)}$$

Hence Proved.

Acceptability of a Language by a Finite Automata

The Language accepted by a dFA $M = (Q, \Sigma, \delta, q_0, F)$ is the set of all strings on Σ^* accepted by M.

In formal ~~language~~ notation

$$L(M) = \{ w \in \Sigma^* : \delta^*(q_0, w) \in F \}$$

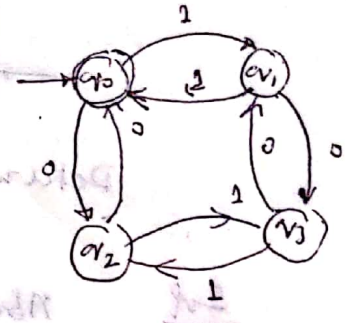
Acceptability of a String by a Finite Automata

A string x is accepted by a FA $M = (Q, \Sigma, \delta, q_0, F)$ if $\delta(q_0, x) = q$ for some $q \in F$.

goto page-09

Example: Consider the finite state machine whose transition function δ is given in table.

States \ Inputs	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2



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

Give the entire sequence of states for the input string 11010.

Sol

$\delta(q_0, 11010) = \delta(q_1, 1010)$ Property (2) $\because \delta(q_0, 1) = q_1$

$= \delta(q_0, 010)$ Property (2) $\because \delta(q_1, 1) = q_0$

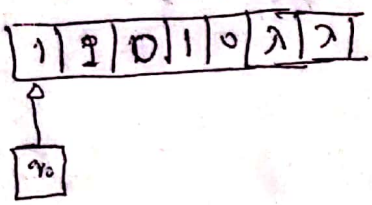
$= \delta(q_2, 10)$ " " $\because \delta(q_0, 0) = q_2$

$= \delta(q_3, 0)$ " " $\because \delta(q_2, 1) = q_3$

$= \delta(q_1, \lambda)$ Property (2) $\because \delta(q_3, 0) = q_1$

$= \delta(q_0, \lambda)$ Property (1)

$= q_0$



Hence.

$$q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{0} q_1 \xrightarrow{1} q_0$$

11010 is a accepted string by d.F.A.

$\because \delta(q_0, 11010) = q_0 \in F$

Hence proved (5)

Nondeterministic Finite automata

Nondeterminism means a choice of moves for an automaton. Rather than prescribing a unique move in each situation, we allow a set of possible moves.

A nfa is defined by the quintuple

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

where

(i) Q is a finite nonempty set of states

(ii) Σ is a " " " of input

(iii) δ is a transition function

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \to 2^Q \quad \text{(power set)}$$

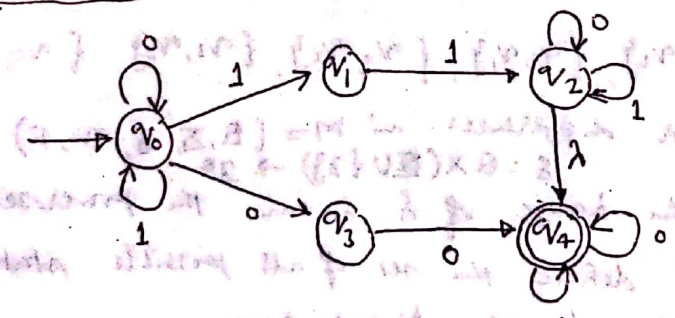
2^Q = All the subset of Q

(iv) $q_0 \in Q$ is the initial state

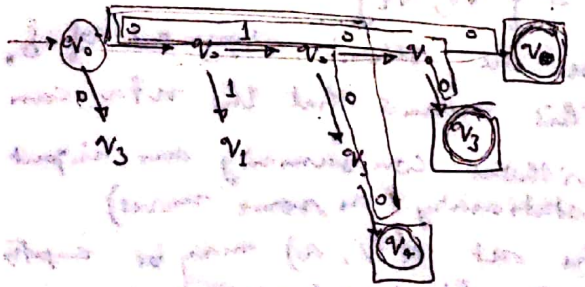
(v) $F \subseteq Q$ is the set of final states

---

e.g.



The sequence of states for input string 0100 is



$q_0 \to q_0 \to q_0 \to q_0 \to [q_3]$

$q_0 \to q_0 \to q_0 \to q_1 \to [q_1]$

$q_0 \to q_0 \to q_0 \to q_1 \to [q_4]$

Hence $\delta(q_0, 0100) = \{q_0, q_1, q_4\}$

Since q_4 is an accepting final state therefore the input string 0100 will be accepted by the nondeterministic auto

Acceptability of a String By a NFA

A string $w \in \Sigma^*$ is accepted by a NDFA $M = (Q, \Sigma, \delta, q_0, F)$ if $\delta^*(q_0, w)$ contains some final state.

i.e $\delta^*(q_0, w) \cap F \neq \phi$

Acceptability of a Language by a NFA

The language L is accepted by a nfa $M = (Q, \Sigma, \delta, q_0, F)$ is defined as the set of all strings accepted in the above sence. Formally.

$$L(M) = \{ w \in \Sigma^* : \delta^*(q_0, w) \cap F \neq \phi \}$$

Difference between DFA & NDFA

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

$2^Q = \{ \{\phi\}, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\} \}$

There are three major differences in $M = (Q, \Sigma, \delta, q_0, F)$

$\delta : Q \times (\Sigma \cup \{\lambda\}) \to 2^Q$

(i) In a nfa, the range of δ is in the powerset 2^Q, subset of 2^Q define the set of all possible states that can be reached by the transition.

e.g. $\delta(q_0, a) = \{q_0, q_2\}$

(ii) λ (empty string) may be the second argument of δ. $\delta(q, \lambda) = q'$. This means that the nfa can make a transition without consuming an input symbol. (Head may be stationary on some moves)

(iii) In an a nfa, the set $\delta(q, a)$ may be empty

$\delta(q, a) = \phi$ [Dead configuration]

i.e. there is no transition defined for this specific situation.

Equivalence of DFA and NDFA

Two finite automata M_1 and M_2 are said to be equivalent if

$$L(M_1) = L(M_2)$$

i.e. if they both accept the same language.

Theorem: For every NFA there exists a DFA which simulate the behavior of NDFA.
OR
If L is the set accepted by NDFA, then there exists a DFA which also accepts L.

Procedure NFA to DFA: (Transition Table)

Let $M_N = (Q_N, \Sigma, \delta_N, q_{0N}, F_N)$ be a NDFA accepting L.
We construct a DFA M_D' as follows:

$$M_D' = (Q_D', \Sigma, \delta_D', q_{0D}', F_D')$$

(i) $Q_D' \equiv 2^{|Q_N|}$ power set of Q_N

(ii) $q_{0D}' = [q_{0N}]$

(iii) F_D' is the set of all subsets of 2^{Q_N} containing an element of F_N.

(iv) $\delta_D'([q_1 q_2 \ldots q_i], a) = \delta_N(q_1, a) \cup \delta_N(q_2, a) \cup \ldots \cup \delta_N(q_i, a)$

Process:→
Start the construction by initial state $\{q_0\}$ first.
Whenever new state apearing under input columns, we add it as a new state in the state column.

Example 1 Construct a DFA equivalent to N-DFA $M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_0\})$ δ is given as

States \ Σ	0	1
→ q_0	q_0	q_1
q_1	q_1	q_0, q_1

Solution Let the equivalent DFA is $M' = (Q', \Sigma, \delta', q_0', F')$

(i) The states Q' are subset of 2^Q

$$Q' = 2^Q = \{\phi, [q_0], [q_1], [q_0, q_1]\}$$

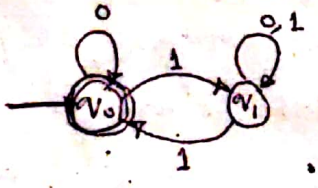
(ii) Initial state $q_0' = [q_0]$

(iii) $F' \subseteq 2^Q$ such that containing an element of $F = \{q_0\}$

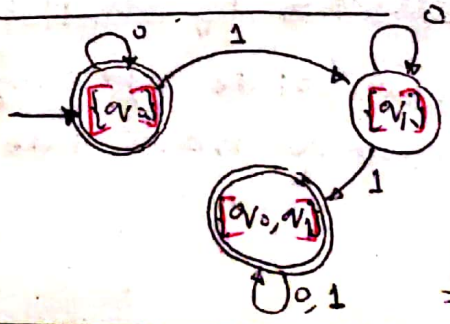
i.e. $F' = \{[q_0], [q_0, q_1]\}$

(iv) δ' is defined by the state table given below

States \ Σ	0	1
$[\phi]$	ϕ	ϕ
→ $[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_1]$



NDFA



DFA

18

Example: 2 Find a deterministic accepted equivalent to Nondeterministic FA

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

δ is given by

State \ Σ	a	b
→ q_0	q_0, q_1	q_2
q_1	q_0	q_1
(*) q_2	—	q_0, q_1

Solution The DFA M' equivalent to M is defined as

$$M' = (2^Q, \{a, b\}, \delta', [q_0], F')$$

where

(i) $2^Q = \{\phi, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$

(ii) Initial state is $[q_0]$

(iii) $F' \subseteq 2^Q$ such that each $\{F_i \subseteq F' : F_i \cap F \neq \phi\}$

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

(iv) δ' is defined as:

States \ Σ	a	b
→ $[q_0]$	$[q_0, q_1]$	$[q_2]$
(*) $[q_2]$	ϕ	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_0]$	$[q_0, q_1]$

$\delta'([q_0, q_1], a) = \delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \{q_0\} = \{q_0, q_1\}$

$\delta'([q_0, q_1], b) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_1, q_2\}$

$\delta'([q_1, q_2], a) = \delta(q_1, a) \cup \delta(q_2, a) = \{q_0, \phi\} = \{q_0\}$

$\delta'([q_1, q_2], b) = \delta(q_1, b) \cup \delta(q_2, b) = \{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\}$

Imp
- Start the construction by initial state $\{q_0\}$ first.
- Whenever New state appearing under Input columns.
- We add it as a New state in the state column.

Example 3 Construct a dfa equivalent to

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

and δ is

State \ Σ	a	b
$\rightarrow q_0$	$q_0\ q_1$	q_0
q_1	q_2	q_1
q_2	q_3	q_3
(q_3)	—	q_2

Sol: Let $Q = \{q_0, q_1, q_2, q_3\}$. Then dfa M' equivalent to M is given by

$$M' = (2^Q, \{a, b\}, \delta', [q_0], F')$$

where $F' \subseteq 2^Q$ such that contains an element of $F = \{q_3\}$

i.e. $F' = \{\{q_3\}, \{q_0 q_3\}, \{q_1 q_3\}, \{q_2 q_3\}, \{q_0 q_1 q_3\}, \{q_0 q_2 q_3\}, \{q_1 q_2 q_3\}, \{q_0 q_1 q_2 q_3\}\}$

δ' given by

State \ Σ	a	b
$[q_0]$	$[q_0\ q_1]$	$[q_0]$
$[q_0\ q_1]$	$[q_0\ q_1\ q_2]$	$[q_0\ q_1]$
$[q_0\ q_1\ q_2]$	$[q_0\ q_1\ q_2\ q_3]$	$[q_0\ q_1\ q_2]$
$[q_0\ q_1\ q_3]$	$[q_0\ q_1\ q_2]$	$[q_0\ q_1\ q_2]$
$[q_0\ q_1\ q_2\ q_3]$	$[q_0\ q_1\ q_2\ q_3]$	$[q_0\ q_1\ q_2\ q_3]$

MEALY AND MOORE MACHINE

MOORE MACHINE

An Automaton in which output depends only on the states of the machine called Moore machine

value of output function at instant $t = t$ sec.

$$\boxed{Z(t) = \lambda(q(t))}$$

where $q(t)$ = present state at $t = t$ sec.

λ is output function

Formal definition

The Moore machine is a six-tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where

1) - Q is a finite set of states.

2) - Σ is the input alphabet.

3) - Δ is the output alphabet.

4) - δ is the transition function over $\Sigma \times Q \to Q$

5) - λ is the output function mapping $Q \to \Delta$

6) - q_0 is the initial state

MEALY MACHINE

Automaton in which the output depends on the state and the input at any instant of time is called a Mealy machine.

i.e The value of the output function $Z(t)$ at $t = t_{sec}$. is the function of present state $q(t)$ at $t = t_{sec}$ and present input $x(t)$ at $t = t_{sec}$.

i.e $$\boxed{Z(t) = \lambda\,(q(t), x(t))}$$

where λ is the output function

Formal definition:

The Mealy machine is a 6-tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where

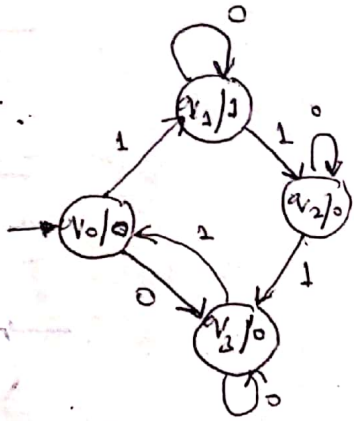
1) – Q is a finite set of states
2) – Σ is the input alphabet
3) – Δ " output "
4) – δ is the transition function mapping $\Sigma \times Q \to Q$
5) – λ is the output function mapping $\Sigma \times Q \to \Delta$
6) – q_0 is the initial state

e.g.

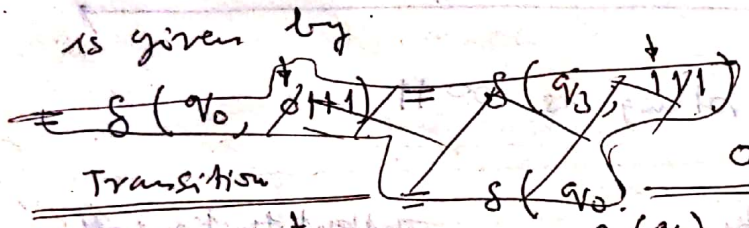
Following table gives a Moore Machine

Table defines δ and λ

Present state	Next state δ		Output
	$a=0$	$a=1$	λ
→ 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



For the input string 0111 the transition states is given by



$= \delta(q_0, 0111)$

$= \delta(q_3, 111)$

$= \delta(q_0, 11)$

$= \delta(q_1, 1)$

$= \delta(q_2, \lambda) = q_2$ (empty string)

$\lambda(q_0) = 0$

$\lambda(q_3) = 0$

$\lambda(q_0) = 0$

$\lambda(q_1) = 1$

$\lambda(q_2) = 0$

Transition States: $q_0 \rightarrow q_3 \rightarrow q_0 \rightarrow q_1 \rightarrow q_2$

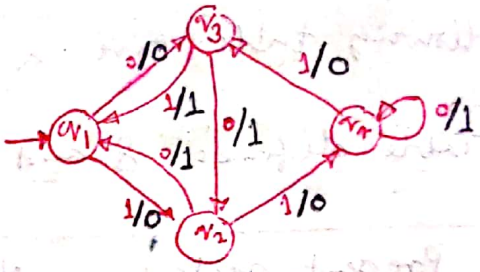
Output string: 0 0 0 1 0

NOTE: For Moore machine if the input string is of length n, then output string will be of length $n+1$.

$n = 4$

$n+1 = 5$

Example of Mealy Machine



δ & λ Table

Present State	NEXT STATE δ			
	$a=0$		$a=1$	
	State	output λ	State	output λ
→ 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

Let the Input string is 0011

Transition will be

$= \delta(q_1, \overset{\downarrow}{0}011) = q_3$

$= \delta(q_3, \overset{\downarrow}{0}11) = q_2$

$= \delta(q_2, \overset{\downarrow}{1}1) = q_4$

$= \delta(q_4, \overset{\downarrow}{1}) = q_3$

$= \delta(q_3, \lambda) = q_3$

$q_1 \rightarrow q_3 \rightarrow q_2 \rightarrow q_4 \rightarrow q_3$

Output function will be

$\lambda(q_1, 0) = 0$

$\lambda(q_3, 0) = 1$

$\lambda(q_2, 1) = 0$

$\lambda(q_4, 1) = 0$

$\lambda(q_3, \Lambda) = \Lambda$ (NULL)

↑ empty string

NOTE

Output = 0100

NOTE: In Mealy machine the input string of length n, the O/P will also be length n

$n \geq 4$

NOTE An FA can be converted into a Moore machine by introducing $\Delta = \{0, 1\}$ and defining $\lambda(q) = 1$ if $q \in F$
$= 0$ if $q \notin F$

TRANSFORMING A MEALY M/C TO MOORE M/C

In this output string will be same (except for the first symbol) for a given input in both the machine.

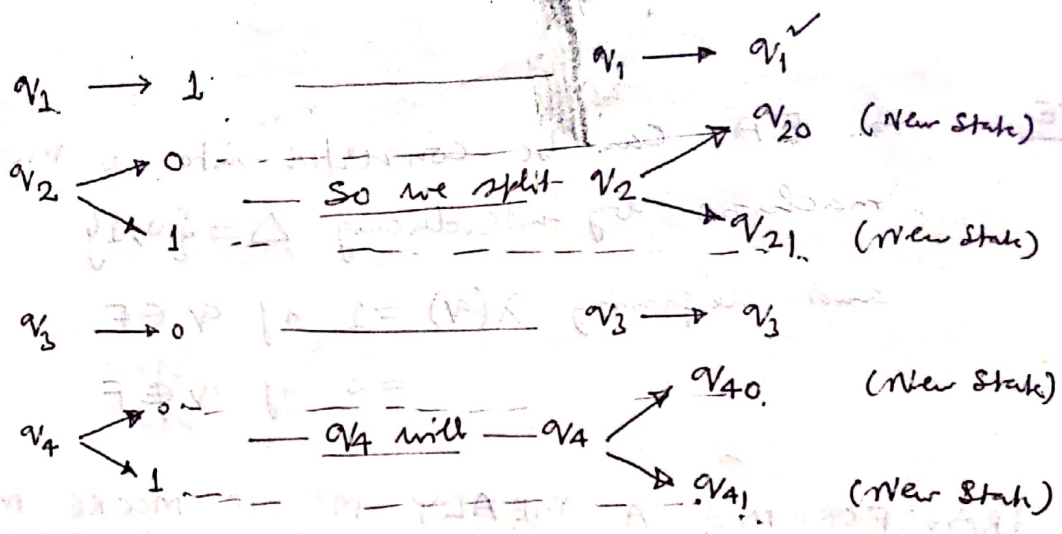
Example 1: Consider the Mealy machine described by the ~~described by the~~ transition table, construct a Moore machine which is equivalent to the Mealy machine.

Mealy machine

Present state	Next state δ			
	Input $a = 0$		Input $a = 1$	
	State	output λ	State	output λ
$\rightarrow q_1$	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

Look at Next state column for any state q_i and determined the no of different ~~associated~~ outputs associated with q_i in that column.

Now we split q_i into several different states the no of such states being equal to the no of different o/p associated with q_i



Given table will be

Present State	NEXT STATE			
	$a=0$		$a=1$	
	State	O/P	State	O/P
$\rightarrow q_1$	q_3	0	q_{20}	0
q_{20}	q_1	1	q_{40}	0
q_{21}	q_1	1	q_{40}	0
q_3	q_{21}	1	q_1	1
q_{40}	q_{41}	1	q_3	0
q_{41}	q_{41}	1	q_3	0

Now the pair of states and O/P in the next state col can be rearranged as called (MOORE Machine)

Present State	NEXT STATE		OUTPUT
	$a=0$	$a=1$	
$\rightarrow q_1$	q_3	q_{20}	1
q_{20}	q_1	q_{40}	0
q_{21}	q_1	q_{40}	1
q_3	q_{21}	q_1	0
q_{40}	q_{41}	q_3	0
q_{41}	q_{41}	q_3	1

In this table initial state q_1 is associated with o/p → 1 i.e. Moore machine accepts a zero length sequence [i.e I/P : 'Λ' $\lambda(q_1) = 1$] which is not accepted by the Mealy machine [$\lambda(q_1, \lambda) = \Lambda$]

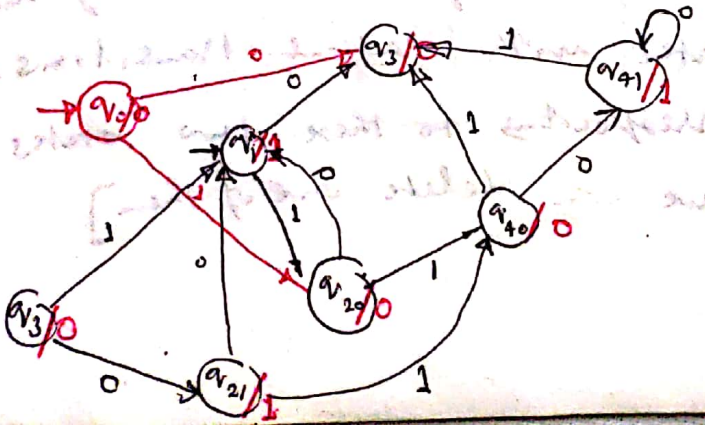
To overcome this problem

either we must neglect the response of Moore machine to input Λ

OR we must add a new starting state q_0 whose transitions are same with q_1 but o/p $\lambda(q_0) = 0$ therefore

MOORE MACHINE

Present State	NEXT State a=0	NEXT State a=1	OUTPUT λ
→ q_0	q_3	q_{20}	0
q_1	q_3	q_{20}	1
q_{20}	q_1	q_{40}	0
q_{21}	q_1	q_{40}	1
q_3	q_{21}	q_1	0
q_{40}	q_{41}	q_3	0
q_{41}	q_{41}	q_3	1



Procedure for MOORE to MEALY M/C

Example 1 — MOORE machine

Present State	NEXT		output
	$a=0$	$a=1$	
$\rightarrow q_0$	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

We must follow the reverse procedure Mealy → Moore

In this case, for every I/P symbol we form the pair consisting of the next state and the corresponding output and reconstruct the table for Mealy M/C

Present State	NEXT State			
	$a=0$		$a=1$	
	State	O/P	State	O/P
$\rightarrow q_0$	q_3	0	q_1	1
q_1	q_1	1	q_2	0
q_2	q_2	0	q_3	0
q_3	q_3	0	q_0	0

NOTE: We can reduce the no of states in any model by considering states with identical transitions. [i.e the rows corresponding to these two states are identical, we can delete one of them]

Example 2

Consider the Moore m/c described by the transition table construct the corresponding Mealy m/c

Moore m/c

Present State	NEXT State		o/p
	$a=0$	$a=1$	
→ q_1	q_1	q_2	0
q_2	q_1	q_3	0
q_3	q_1	q_3	1

Sol:

Present State	NEXT State			
	$a=0$		$a=1$	
	State	o/p	State	o/p
→ q_1	q_1	0	q_2	0
q_2	q_1	0	q_3	1
q_3	q_1	0	q_3	1

In this table corresponding to q_2 & q_3 rows are identical. So we can delete one of the two states i.e q_2 or q_3

Present State	NEXT State			
	$a=0$		$a=1$	
	State	o/p	State	o/p
→ q_1	q_1	0	q_2	0
q_2	q_1	0	q_3 (= q_2)	1