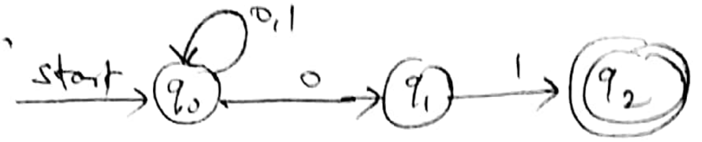
Convert the following NDFSM to the equivalent DFSM. (June 2018 exam) (Important)



Solution:- Construct Transistion table

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

Given NDFSM $N = (Q_N, \Sigma_N, \delta_N, q_{0N}, F_N)$

Where $Q_N = \{q_0, q_1, q_2\}$ - set of states of NDFSM

$\Sigma = \{0, 1\}$ - set of Symbols — " —

δ_N = Transistion functions of N

$q_{0N} = q_0$ start state of NDFSM

$F_N = \{q_2\}$ set of Final states of NDFSM

We need to Construct DFSM $D = (Q_D, \Sigma_D, \delta_D, q_{0D}, F_D)$

(i) $\Sigma_D = ?$, We know $\Sigma_D = \Sigma_N$ & $\Sigma_N = \{0, 1\}$

$$\boxed{\therefore \Sigma_D = \{0, 1\}}$$

(ii) $q_{0D} = ?$ We know $q_{0N} = q_0$ (start state of NDFSM)

We know Start state of DFSM is the set that contains start state of NDFSM

i.e. $q_{0D} = \{q_{0N}\}$

$$\boxed{q_{0D} = q_0}$$

(iii) $\delta_D = ?$ δ_D can be constructed using δ_N.

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

$\delta_D(\{q_0\}, 0) = \delta_N(q_0, 0) = \{q_0, q_1\}$

$\therefore \delta_D(\{q_0\}, 0) = \{q_0, q_1\}$

Note: determine δ_D using δ_N (transistion table of NDFSM)

$\delta_D(\{q_0\}, 1) = \delta_N(q_0, 1) = \{q_0\}$

$\delta_D(\{q_0, q_1\}, 0) = \delta_N(q_0, 0) \cup \delta_N(q_1, 0)$

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

$\delta_D(\{q_0, q_1\}, 0) = \{q_0, q_1\}$

$\delta_D(\{q_0, q_1\}, 1) = \delta_N(q_0, 1) \cup \delta_N(q_1, 1)$

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

$\delta_D(\{q_0, q_1\}, 1) = \{q_0, q_2\}$

$\delta_D(\{q_0, q_2\}, 0) = \delta_N(q_0, 0) \cup \delta_N(q_2, 0)$

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

$\delta_D(\{q_0, q_2\}, 0 = \{q_0, q_1\}$

$\delta_D(\{q_0, q_2\}, 1) = \delta_N(q_0, 1) \cup \delta_N(q_2, 1)$

$= \{q_0\} \cup \phi$

$\underline{\delta_D(\{q_0, q_2\}, 1) = \{q_0\}}$

(iv) $Q_D = \{\{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}\}$

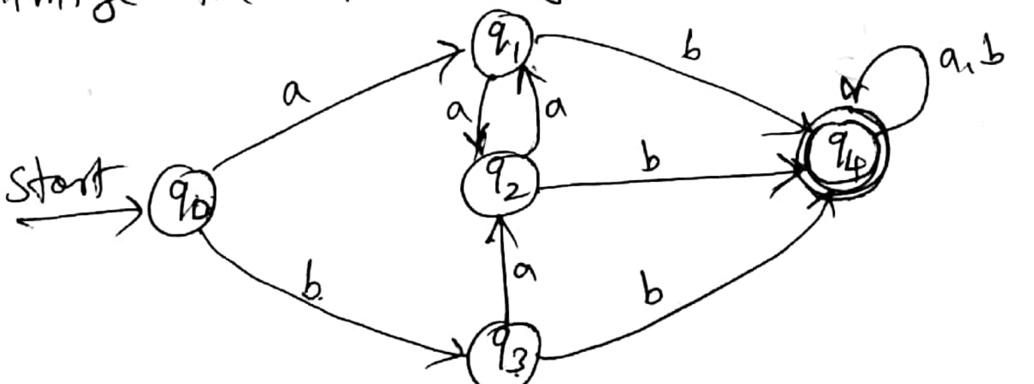
(v) $F_D = ?$ We know $F_N = \{q_2\}$ (final states of NDFSM)

F_D contains those states of Q_D that contains atleast one final state of F_N.

$\underline{F_D = \{\{q_0, q_2\}\}}$

June 2018 (exam)

Minimize the following Finite Automata (FSM)



Solution: Number of states of given FSM, $n = 5$

Draw a table of size $(n-1) \times (n-1)$

q_1	✗			
q_2	✗	✓		
q_3	✗	✓	✓	
* q_4	✗	✗	✗	✗
	q_0	q_1	q_2	q_3

equivalent pair of states are: $(q_1, q_3), (q_1, q_2)$

$(q_2, q_3) = \underline{(q_1, q_2, q_3)}$

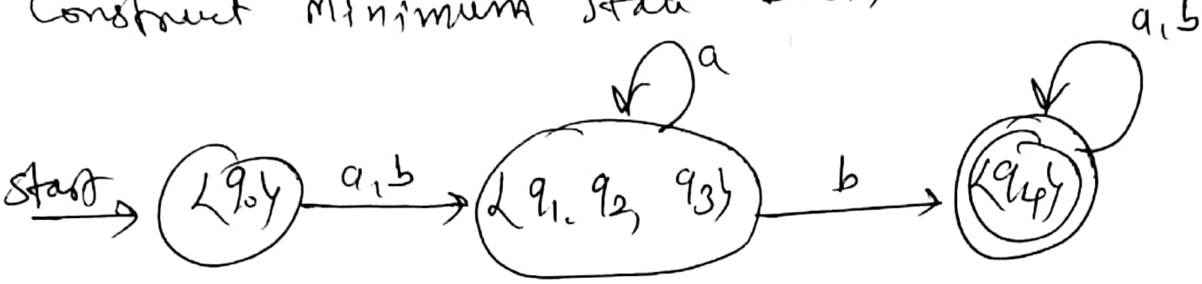
3 pairs of states ~~are~~ can be combined into (q_1, q_2, q_3)

Partition the set Q into blocks of mutually equivalent states

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

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

Now Construct Minimum state DFSM



$\delta (\{q_0\}, a) = q_1$ is in block $\{q_1, q_2, q_3\}$.

Draw an arc from $\{q_0\}$ to $\{q_1, q_2, q_3\}$ labelled with a

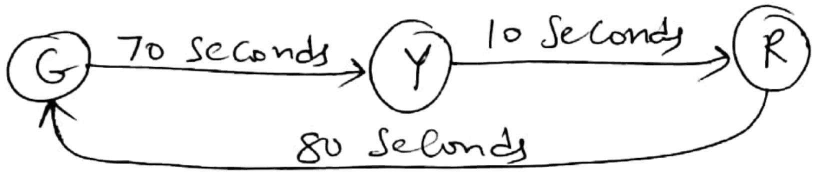
$\delta (\{q_0, b\}) = q_3$ is in block $\{q_1, q_2, q_3\}$

Draw an arc from $\{q_0\}$ to $\{q_1, q_2, q_3\}$ labelled with b

Important Finite state Transducers (FST) It is a finite state model that allow for output at each step of machine's operation.
One type of Finite State Transducer which associates an output with each state of a machine M. That output is generated when Machine (M) enters the associated state. Deterministic Finite State Transducers (DFST) of this sort are called
Moore Machines. A Moore machine M is a seven tuple (Q, Σ, δ, q_0, F, O, D)
Where Q is the Set of states
Σ is the set of input symbols
δ is the transistion function. It is the function from $(Q \times \Sigma)$ to Q
q_0 is the initial state
F is the Set Final (accepting) states.
O is output alphabet
D is the display or output functions, It is the function from (k) to (O^*)
A Moore machine M computes a function $f(w)$ iff it reads the input string w, its output sequence is $f(w)$,
Example: A typical United states traffic light, Consider the following Controller for a single direction of a very simple U.S. traffic light

(which ignores time of ~~delay~~ the day, traffic, need to let emergency vehicles through, etc)

The states in this simple Controller corresponds to light's colors: green yellow and Red. There are 3 inputs, all of which are elapsed time.



Mealy Machine (M) :- It is a deterministic Finite state Transducer which permits ~~each~~ to output any finite sequence of symbols as it makes each Transistion (as it reads each symbol of input). FSM that associates outputs with transistions are called Mealy Machines. A Mealy machine M is a six tuple $(Q, \Sigma, \delta, q_0, F, O)$ where

Q is the set of states

Σ is the set of input symbols (input alphabet)

δ is the transistion function

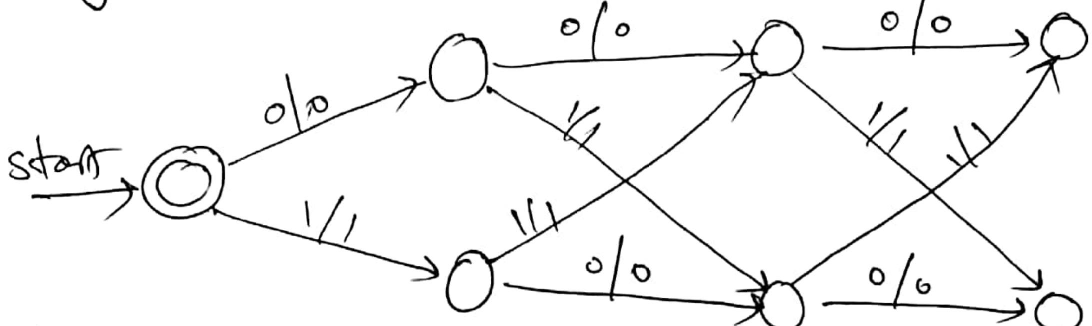
q_0 is the start state

F is the set of Final states

O is the output alphabet

A mealy Machine Computes a function $f(w)$ iff it reads the input string w, its output sequence is $f(w)$.

example, Generating Parity Bits.
The following Mealy machine adds an odd parity bit after every four binary digits that it reads. We will use the notation a|b on an arc to mean that the transition may be followed if the input character is a. if it is followed then string b will be generated.



Digital Circuits can be modelled as transducers using either Moore or Mealy Machines.

Bidirectional Transducers :- A Process that reads an input string and constructs a corresponding output string can be described in a veriety of different ways. The main reason we choose the Finite State Transducer (FST) model is that it provides a declarative, rather than a procedural way to describe the relationship between inputs and outputs.
Such a declarative model can ~~be~~ be run in two directions (Bi-directional Transducer).

(Important)

Equivalent (In-distinguishable) pair of states :-

Definition: Two states p and q of a FSM are equivalent if and only if $\delta(p, w)$ and $\delta(q, w)$ are both ~~but~~ Final states or Non-Final states for all strings w in Σ^*

That is if $\delta(P, w) \in F$ (belongs to) and $\delta(q, w) \in F$

Where F is a set of Final states

Distinguishable pair of states :-

Definition: Two states p and q of a FSM are distinguishable if and only if $\delta(p, w)$ is Final state and $\delta(q, w)$ is Non-final state and Vice Versa.

i.e $\delta(p, w) \in F$ and $\delta(q, w) \notin F$

(does not belong to)

Note: $\delta(P, w)$ is a state reached from p after reading string $w = a_1 a_2 \ldots a_n$.

$\delta(q, w)$ is a state reached from q after reading string $w = a_1 a_2 \ldots a_n$

While minimizing Finite State Machine (FSM), two or more equivalent states are combined into a single state.

Consider the following DFSM

(Jan. 2018 exam) (9 marks)

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

(i) Draw the table of distinguishable & indistinguishable (Equivalent)

(ii) Construct minimum state equivalent automata (FSM)

Number of states of DFSM, $n = 8$

Draw a table of size $n-1 \times n-1$ i.e. 7×7

	A	B	C	*D	E	F	G
B	✗						
C	✗	✗					
*D	✗	✗	✗				
E	✗	✗	✓	✗			
F	✗	✓	✗	✗	✗		
G	✓	✗	✗	✗	✗	✗	
H	✗	✗	✗	✗	✗	✗	✗

Distinguishable – ✗

Equivalent – ✓

∴ Equivalent pair of states are : (A, G), (B, F), (C, E)

(A, B) — $A \xrightarrow{0} B$ (NFS) $A \xrightarrow{1} A$ (NFS)
NFS NFS $B \xrightarrow{0} A$ (NFS) $B \xrightarrow{1} C$ (NFS) $(A\ B)$ Equivalent mark ✓

(A, C) — $A \xrightarrow{0} B$ (NFS)
NFS NFS $C \xrightarrow{0} D$ (FS) (A, C) — distinguishable mark ×

(A, E) $A \xrightarrow{0} B$ (NFS)
NFS NFS $E \xrightarrow{0} D$ (FS) (A, E) distinguishable mark ×.

(A, F) $A \xrightarrow{0} B$ (NFS) $A \xrightarrow{1} A$ (NFS)
NFS NFS $F \xrightarrow{0} G$ (NFS) $F \xrightarrow{1} E$ (NFS) ∴ $(A\ E)$ distinguishable mark ×

(A, G) $A \xrightarrow{0} B$ (NFS) $A \xrightarrow{1} A$ (NFS)
NFS NFS $G \xrightarrow{0} F$ (NFS) $G \xrightarrow{1} G$ (NFS) ∴ (A, G) is equivalent. mark ✓

(A, H) $A \xrightarrow{0} B$ (NFS) $A \xrightarrow{1} A$ (NFS)
NFS NFS $H \xrightarrow{0} G$ (NFS). $H \xrightarrow{1} D$ (FS) (A, D) distinguishable mark ×

(B, C) $B \xrightarrow{0} A$ (NFS).
NFS NFS $C \xrightarrow{0} D$ (FS) (B, C) distinguishable mark ×.

(B, D) — mark × — distinguishable
NFS FS

(B, E) — $B \xrightarrow{0} A$ (NFS) mark × since (B, E)
NFS $E \xrightarrow{0} D$ (FS) is distinguishable

(B, F) — $B \xrightarrow{0} A$ (NFS) $\left.\begin{matrix} A \xrightarrow{1} C \\ F \xrightarrow{1} E \end{matrix}\right\}$ (NFS) equivalent
NFS $F \xrightarrow{0} G$ (NFS) mark ✓

(B, G) — $\left.\begin{matrix} B \xrightarrow{0} A \\ G \xrightarrow{0} F \end{matrix}\right\}$ NFS $\left.\begin{matrix} B \xrightarrow{1} C \\ G \xrightarrow{1} G \end{matrix}\right\}$ NFS equivalent mark ✓
NFS

$\underbrace{(B, H)}_{NFS}$ – $\left.\begin{matrix} B \xrightarrow{0} A \\ H \xrightarrow{0} G \end{matrix}\right\}$ NFS $\quad$ $B \xrightarrow{1} C$ (NFS) $\quad$ $H \xrightarrow{1} D$ (FS)

(B, H) distinguishable mark X

$\underbrace{(\overset{NFS}{C} . \overset{(FS)}{D})}_{\cancel{FS}}$ – ~~$C \xrightarrow{0} D$ (FS)~~ ~~$C \xrightarrow{1} B$ (NFS)~~

~~$D \xrightarrow{0} D$ (FS)~~ ~~$D \xrightarrow{1} A$ (NFS)~~ ~~∴ (C.D) distinguishable mark X~~

(C.D) is distinguishable. mark X

$\underbrace{(C, E)}_{NFS}$ – $\left.\begin{matrix} C \xrightarrow{0} D \\ E \xrightarrow{0} D \end{matrix}\right\}$ FS $\quad$ $\left.\begin{matrix} C \xrightarrow{1} B \\ E \xrightarrow{1} F \end{matrix}\right\}$ NFS.

∴ (C.E) is equivalent mark ✓

$\underbrace{(C.F)}_{NFS}$ – $C \xrightarrow{0} D$ (FS) $\quad$ $F \xrightarrow{0} G$ (NFS)

(C.F) is distinguishable mark X

$\underbrace{(C.G)}_{NFS}$ – $C \xrightarrow{0} D$ (FS) $\quad$ $F \xrightarrow{0} G$ (NFS)

(C.G) is distinguishable mark X

$\underbrace{(C, H)}_{NFS}$ – $C \xrightarrow{0} D$ (FS) $\quad$ $H \xrightarrow{0} G$ (NFS)

(C, H) is distinguishable mark X

$\underbrace{(E, F)}_{NFS}$ – $E \xrightarrow{0} D$ (FS) $\quad$ $F \xrightarrow{0} G$ (NFS)

(E, F) is distinguishable mark X

$\underbrace{(E, G)}_{NFS}$ – $E \xrightarrow{0} D$ (FS) $\quad$ $G \xrightarrow{0} F$ (NFS)

(E, G) is distinguishable. mark X

$\underbrace{(E, H)}_{NFS}$ – $E \xrightarrow{0} D$ (FS) $\quad$ $H \xrightarrow{0} G$ (NFS)

(E, H) distinguishable mark X

$\underbrace{(F, G)}_{NFS}$ $\left.\begin{array}{l} F \xrightarrow{0} G \\ G \xrightarrow{0} H \end{array}\right\}$ NFS $\left.\begin{array}{l} F \xrightarrow{1} E \\ G \xrightarrow{1} G \end{array}\right\}$ NFS

(F, G) - ~~equi~~ equivalent. Mark ✓

$\underbrace{(F, H)}_{NFS}$ - $\left.\begin{array}{l} F \xrightarrow{0} G \\ H \xrightarrow{0} G \end{array}\right\}$ NFS $\begin{array}{l} F \xrightarrow{1} E \text{ (NFS)} \\ H \xrightarrow{1} D \text{ (FS)} \end{array}$

∴ (F, H) is distinguishable
Mark X

(G H) - $\left.\begin{array}{l} G \xrightarrow{0} F \\ H \xrightarrow{0} G \end{array}\right\}$ NFS $\begin{array}{l} G \xrightarrow{1} G \text{ (NFS)} \\ H \xrightarrow{1} D \text{ (FS)} \end{array}$

∴ (G, H) - Distinguishable.
Mark - X.

Now Consider only pair of states marked ✓.

(A, B), Since $\begin{array}{l} A \xrightarrow{1} A \\ B \xrightarrow{1} C \end{array}$ and (A, C) are marked distinguishable

∴ (A B) is also distinguishable

∴ re-mark (A B) with X

Consider (A G)

$\left.\begin{array}{l} A \xrightarrow{0} B \\ G \xrightarrow{0} F \end{array}\right\}$ marked ✓ $\left.\begin{array}{l} A \xrightarrow{1} A \\ G \xrightarrow{1} G \end{array}\right\}$ marked ✓

∴ (A G) is equivalent

Now Consider $\underbrace{(B, F)}_{NFS}$

$\left.\begin{array}{l} B \xrightarrow{0} A \\ F \xrightarrow{0} G \end{array}\right\}$ marked with ✓ $\left.\begin{array}{l} B \xrightarrow{1} A \\ F \xrightarrow{1} G \end{array}\right\}$ Marked with ✓

∴ (B, F) is equivalent – ✓

Now Consider (B, G)

$\left.\begin{array}{l} B \xrightarrow{0} A \\ G \xrightarrow{0} F \end{array}\right\}$ marked distinguishable.

∴ (B G) is distinguishable (✗)

Now Consider $\underbrace{(C \ E)}_{NFS}$

$C \xrightarrow{0} D$ (FS)
$E \xrightarrow{0} D$ (FS)

$\left.\begin{array}{l} C \xrightarrow{1} B \\ E \xrightarrow{1} F \end{array}\right\}$ marked with ✓

∴ (C, E) is equivalent

Consider $\underbrace{(F, G)}_{NFS}$

$\left.\begin{array}{l} F \xrightarrow{0} G \\ G \xrightarrow{0} F \end{array}\right.$ or $\left.\begin{array}{l} F \\ G \end{array}\right\}$ marked with ✓

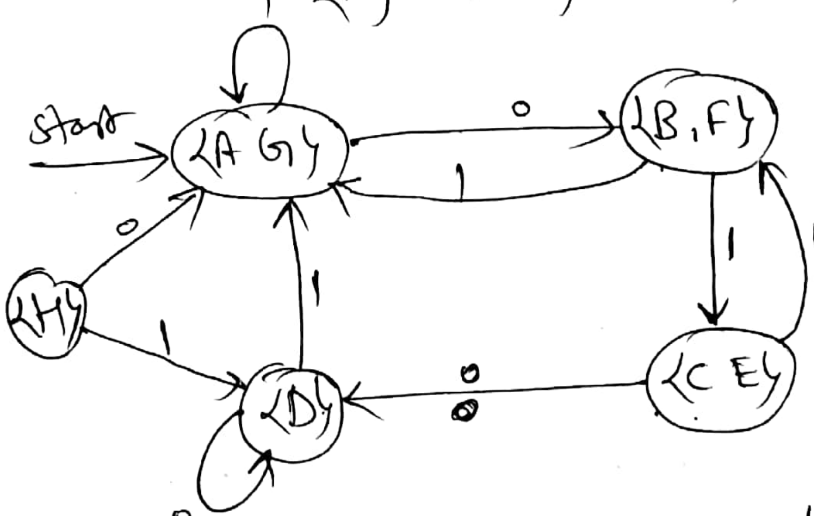
$\left.\begin{array}{l} F \xrightarrow{1} E \\ G \xrightarrow{1} G \end{array}\right\}$ marked with ✗ ∴ (F, G) is distinguishable
mark ✗

∴ Equivalent pair of states are: (A G), (B, F) (C, E)

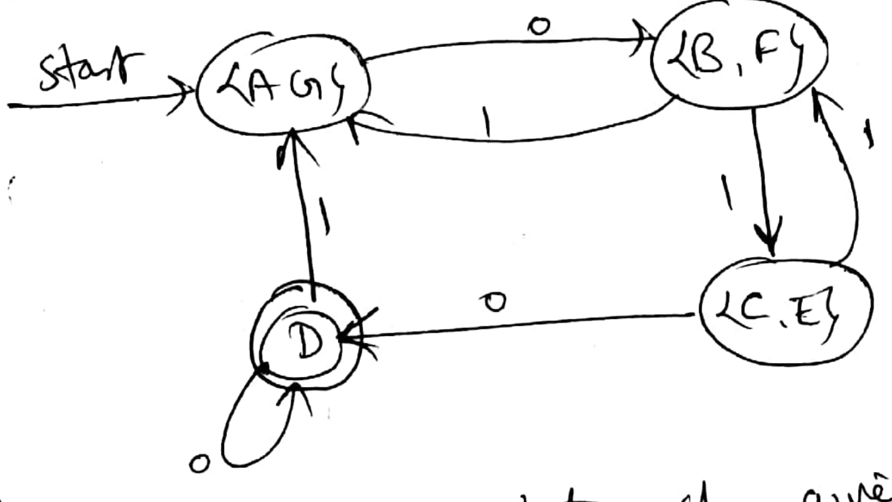
In a given FSM Set $Q = \underbrace{\{A, B, C, D, E, F, G, H\}}_{\text{States of FSM}}$

Partition set Q into blocks of states which are mutually equivalent

{D} {A G} {B, F} {C E} {H}



Since state H is not reachable from start state {A G}, so eliminate state H. To produce following <u>minimum state FSM</u>



Since D is Final state of given DFSM, D is retaining Final state for <u>Minimum State DFSM</u>