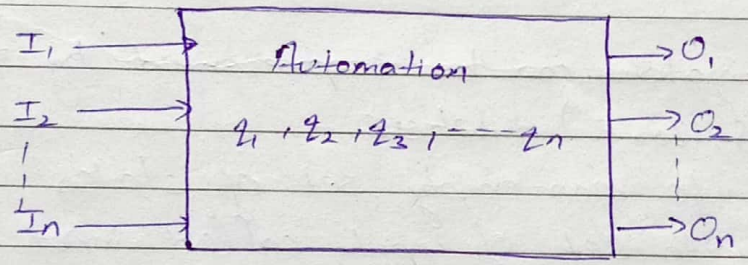
Theory of Automata and Formal languages (BCS T-405)

Automation

It is defined as a system where energy, materials & info are transformed, transmitted and used for performing some functions without direct participation of mass.

Example: Automatic machine tools, automatic packing machine, automatic photo printing machines etc. Aut

Automation is defined ~~as~~ in more abstract way as shown in fig:



- It's characteristics are now described as:

(i) INPUT: At each discrete instant of time $t_1, t_2, -- t_n$ input value $I_1, I_2 --- I_n$ each of which can take a finite no. of fixed value from the input alphabet 'Σ', are applied to the input side of model shown in figure

(ii) OUTPUT: $O_1, O_2, -- O_n$ are the outputs of the model, each of which can take a finite no. of fixed values from an output 'O'.

(iii) STATES: At any instant of time the automation can be in one of the states $q_1, q_2, --- q_n$.

(iv) State Relations: The next state of an automation at an instant of time is determined by the present state and the present input.

(v) Output Relations: Output is related to either state only or to both the input and the state. It should be noted that any instant of time, the automation is in some state. On reading an input symbol, the automation move to a next state which is given by the state relation.

Finite Automaton:-

A finite automaton can be represented by five-tuple $(Q, \Sigma, \delta, q_0, F)$ where,

(i) Q is a finite non-empty set of states

(ii) Σ is a finite set of input called input alphabet

(iii) δ is a transition function which maps $Q \times \Sigma$ into Q and is usually called direct transition function. This is the function which describes the changes of states during transition. This mapping is usually represented by a transition table or transition diagram.

(iv) q_0 belongs to Q is the initial state

(v) F is subset of Q ($F \subseteq Q$) is the set of final state. It is assumed that there may be more than one final state.

Imp Note: The transition function which maps $Q \times \Sigma^*$ into Q is called indirect transition function ie. maps a state & string ~~into~~ of input symbol including the empty string into a state. We shall

use the same symbol 'δ' to represent both types of transition function and difference can be easily identified by nature of mapping.

→ The Automation model can be represented graphically by given figure:-

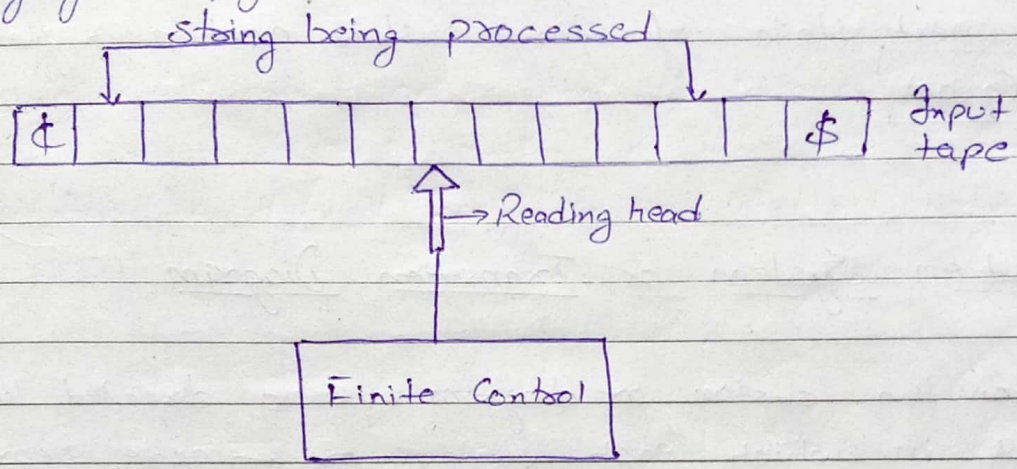


Fig: Block diagram of finite Automaton

The various components of a finite Automaton are explained as follows:-

(i) Input Tape:- It is divided into squares. Each square containing a single symbol from the input alphabet Σ. The end square of the tape contains end marker (¢) at left & ($) at the right end. Absence of end markers indicate that the tape is of infinite length. The left to right sequence of symbol b/w the end markers is the I/P string to be processed.

(ii) Reading Head:- The Head examine only one square at a time, and can move 1 square either to the left or to the right. For further analysis, we restrict the movement of reading head only to the right side.

(iii) Finite Control: The i/p to the finite control will be usually symbol under the reading head, say 'a' as present state of the machine, say 'q' to give the following output

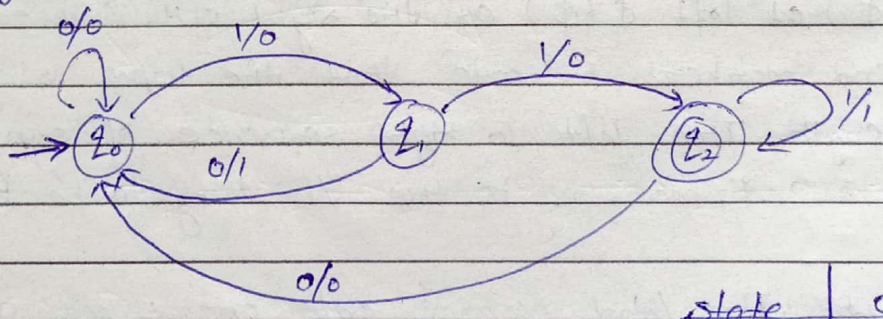
(a) A motion of reading head along the tape to the next square.

(b) The next state of finite state machine is given by $\delta(q, a)$

Transition System or Transition Diagram

A transition system or diagram is a directed labelled graph in which each vertex or node represent a state & the directed edge indicates the transition of state and the edge are labelled with I/P / o/p

A transition system is shown in the given fig., where the initial state is represented by a circle with an arrow pointing towards it, The final state by the 2 concentric circles, and the other state is represented by a circle.



State	Input 0	1
→ q_0	q_0	q_1
q_1	q_0	q_2
(q_2)	q_0	q_2

The edges are labelled as input/output, 1/0, 0/0, 0/1 etc.

For eg:- if the system is in state q_0 & the input '1' is applied, the system moves to q_1 as there is a directed edge from q_0 to q_1 with label 1/0 & the output is '0'

Definition of transition System

A transition system is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where,

(i) Q, Σ & F are the finite non-empty set of states, the I/P alphabet & set of final state, as in case of finite automata.

(ii) $q_0 \subseteq Q$ is the initial state and q_0 is non-empty.

(iii) δ is a transition funct^n which maps $Q \times \Sigma^*$ into Q.

Acceptability of string by a finite Automaton

A string is accepted by a finite automaton & is equal

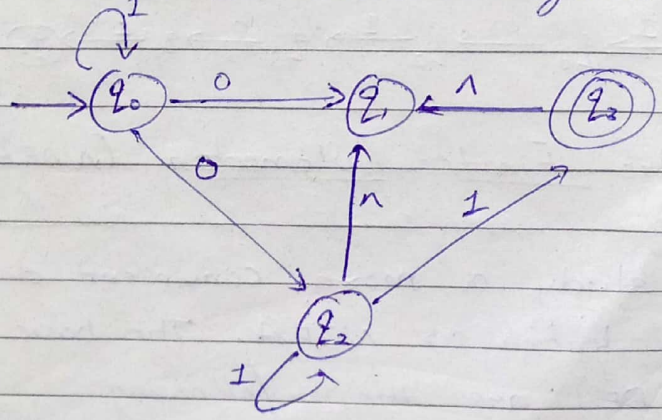
$M = (Q, \Sigma, \delta, q_0, F)$

if $\delta(q_0, x) = q$

for some, $q_0 \in F$

This is basically the acceptability of a string by a finite state.

Eg:- Consider the transition system given in fig.



Determine the initial & final states & the acceptability of

101, 1011, 1010, 10111

✓ ✓ ✗ ✓

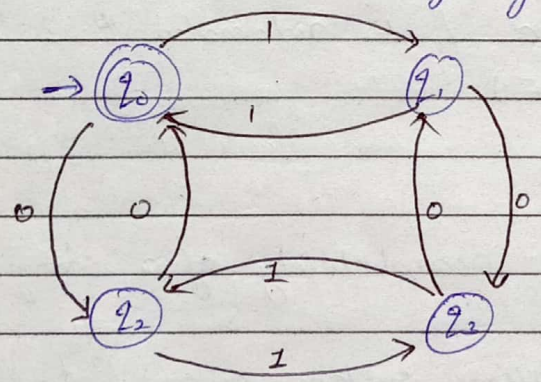
Acceptable.

(Q-) Consider the transition system given in below table in the form of transition table. Here $Q = \{q_0, q_1, q_2, q_3\}$ $\Sigma = \{0, 1\}$, $F = \{q_0\}$

state	Input	
	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

Determine the initial & final states & entry sequence of the states for I/P 110101

Sol:- The transition diag. of above table is



Seq:- $\rightarrow q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{0} q_1 \xrightarrow{1} q_0$

Non-Deterministic Finite Automaton (NDFA)

We shall now study a more convinient device to design a DFA known as NDFA. The basic diff b/w NDFA & DFA are the following

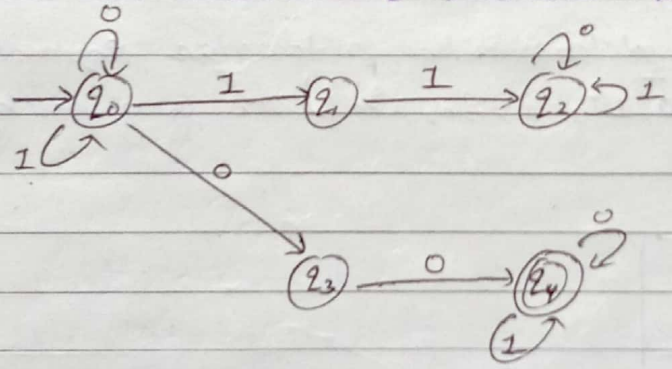
(1) In NDFA we can have several possible next states for a given pair of current state and I/P symbol.

(2) A state can change into next state without having any I/P symbol i.e. empty string move are possible

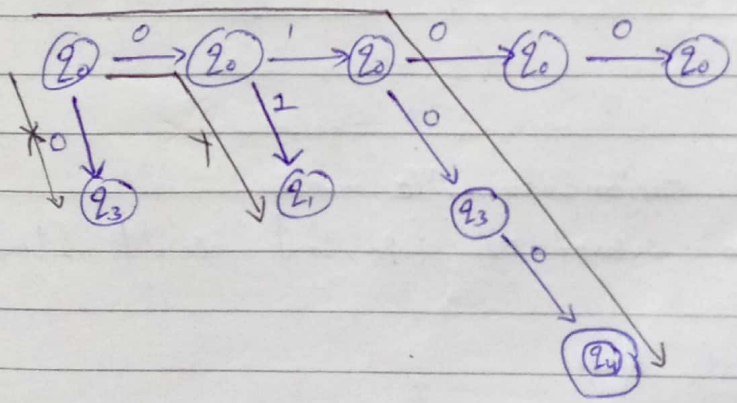
(3) The set $\delta(s,a)$ $(a \in \Sigma, s \in S)$ may be empty i.e. there may not be any transition defined for the specific pair of I/P symbol & current state

The idea of introducing the concept of NDFA is to simplify method of designing a DFA. There are several problems & situations when we donot know what be the next move or exactly next state. In case several moves are possible, we start with one & follow it until the path becomes the right path. Thus, the NDFA may be regarded as a parallel computer, where several process can run simultaneously.

Eg:- The non-deterministic automaton whose transition diag. is-



The seq. of states of I/P string 0100 is given in figure:-



NDFA

Definition: A non-deterministic finite automata (NDFA) is 5-tuple $(Q, \Sigma, \delta, q_0, F)$

where (i) Where Q is finite non-empty set

(ii) Σ is a finite non-empty set of I/P alphabet.

(iii) δ is a transition funct^n mapping from $Q \times \Sigma$ into 2^Q which is the powerset of Q, the set of all subset of Q

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

(v) $F \subseteq Q$ is the set of finite state.

Note: Diff b/w DFA & NDFA is only in δ. For DFA, the outcome is a state i.e an element of Q. For NDFA the outcome is a subset of Q

Equivalence of DFA & NDFA

Ex: Construct a deterministic automaton equivalent to $M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_0\})$ (Q, Σ, F) δ is given by it's state table

state	Input	
	0	1
$\rightarrow (q_0)$	q_0	q_1
q_1	q_1	q_0, q_1

Solu^n For deterministic automaton M_1

(i) The state are subset of $\{q_0, q_1\}$ i.e. ϕ, $[q_0]$, $[q_1]$, $[q_0, q_1]$

(ii) q_0 is the initial state

(iii) $[q_0]$ and $[q_0, q_1]$ is the final state as these are

the only state containing $[q_0]$

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

state	Input	
	0	1
ϕ	ϕ	ϕ
$[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0 q_1]$	$[q_0 q_1]$

Note: When M has n states the corresponding finite automaton has 2^n states However, we need not construct δ for all these 2^n states, but only for those state reachable from $[q_0]$, this is bcoz our interact is only in constructing M_1 accepting the machine so we start the constuction of δ for $[q_0]$. We continue by considering those state appearing earlier under input column and constructing δ for such states. We halt when no more new states appear under the i/p coloumn.

Ex: Find a deterministic acceptor equivalent to $M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$ δ is given by it's state table.

state	Input	
	a	b
$\rightarrow q_0$	q_0, q_1	q_2
q_1	q_0	q_1
q_2		$q_0 q_1$

Soln :-

state	Input a	Input b
ϕ	ϕ	ϕ
$[q_0]$	$[q_0 q_1]$	$[q_2]$
$[q_2]$	ϕ	$[q_0 q_1]$
$[q_0 q_1]$	$[q_0 q_1]$	$[q_1 q_2]$
$[q_1 q_2]$	$[q_0]$	$[q_0 q_1]$

Q-1 Construct a deterministic automata equivalent to $M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$ δ is given by it's state table.

state	Input 0	Input 1
$\rightarrow q_0$	$q_0 q_1$	q_0
q_1	q_2	q_1
q_2	q_3	q_3
(q_3)		q_2

Soln :-

state	Input 0	Input 1
ϕ	ϕ	ϕ
$[q_0]$	$[q_0 q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0 q_1 q_2]$	$[q_0 q_1]$
$[q_0 q_1 q_2]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_3]$
$[q_0 q_1 q_3]$	$[q_0 q_1 q_2]$	$[q_0 q_1 q_2]$
$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_2 q_3]$

Q) Construct a deterministic finite automaton equivalent to $M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_3\})$ δ is given in table

State	Input 0	Input 1
$\rightarrow q_1$	q_2, q_3	q_1
q_2	q_1, q_3	ϕ
(q_3)	q_2	q_1, q_2

Sol:

state	Input 0	Input 1
ϕ	ϕ	ϕ
$[q_1]$	$[q_2, q_3]$	$[q_1]$
$[q_2, q_3]$	$[q_1, q_2, q_3]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_1, q_2, q_3]$	q_1
$[q_1, q_2, q_3]$	$[q_1, q_2, q_3]$	$[q_1, q_2]$

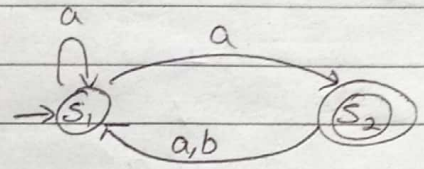
Q) The transition table of a NDFA M is given in table. Construct a DFA equivalent to M

State	Input 0	Input 1	Input 2
$\rightarrow q_0$	q_1, q_4	q_4	q_2, q_3
q_1		q_4	
q_2			$q_2 q_3$
(q_3)		q_4	
q_4			

Solⁿ

state	Input		
	0	1	2
q_0	q_1, q_4	q_4	q_2, q_3
ϕ	ϕ	ϕ	ϕ
q_4	ϕ	ϕ	ϕ
q_1, q_4	ϕ	q_4	ϕ
q_2, q_3	ϕ	ϕ	q_2, q_3

Q1 Obtain equivalent DFA for following NDFA

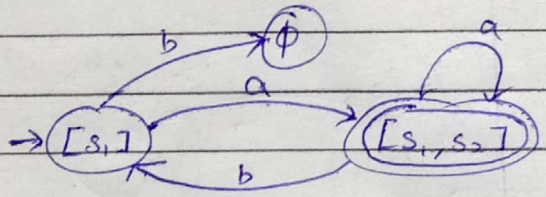


Solⁿ

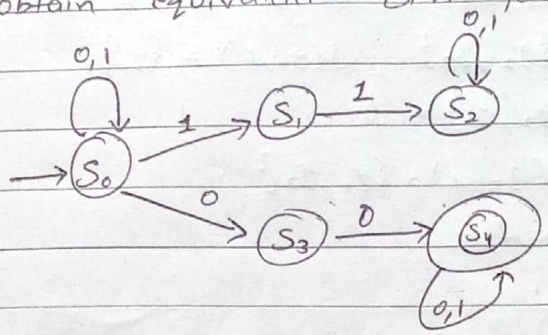
state	Input	
	a	b
$\rightarrow S_1$	S_1, S_2	ϕ
(S_2)	S_1	S_1

Equivalent

state	Input	
	a	b
$\rightarrow [S_1]$	$[S_1, S_2]$	ϕ
ϕ	ϕ	ϕ
$[S_1, S_2]$	$[S_1, S_2]$	$[S_1]$



Q.) obtain equivalent DFA for the following NDFA



Sol^n Transition ~~State~~ Table

State	Input	
	0	1
S_0	S_0, S_3	S_0, S_1
S_1	ϕ	S_2
S_2	S_2	S_2
S_3	S_4	ϕ
S_4	S_4	S_4

Equivalent table

State	Input	
	0	1
ϕ	ϕ	ϕ
$[S_0]$	$[S_0, S_3]$	$[S_0, S_1]$
$[S_0, S_1]$	$[S_0, S_2]$	$[S_0, S_1, S_2]$
$[S_0, S_3]$	$[S_0, S_3, S_4]$	$[S_0, S_1]$
$[S_0, S_1, S_2]$	$[S_0, S_3, S_2]$	$[S_0, S_1, S_2]$
$[S_0, S_3, S_4]$	$[S_0, S_3, S_4]$	$[S_0, S_1, S_4]$
$[S_0, S_1, S_4]$	$[S_0, S_3, S_4]$	$[S_0, S_1, S_2, S_4]$
$[S_0, S_1, S_2, S_4]$	$[S_0, S_2, S_3, S_4]$	$[S_0, S_1, S_2, S_4]$
$[S_0, S_2, S_3, S_4]$	$[S_0, S_2, S_3, S_4]$	$[S_0, S_1, S_2, S_4]$
$[S_0, S_2, S_3]$	$[S_0, S_2, S_3, S_4]$	$[S_0, S_1, S_2,]$

Now draw it's diagram

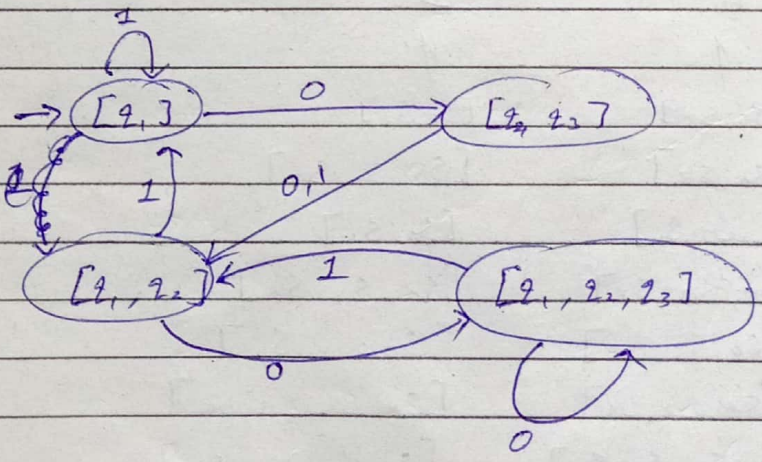
Q-1 $M = (\{q_1, q_2, q_3\}, \{0, 1\}, \delta, q_1, \{q_3\})$ is a NDFA when δ is given by $\delta(q_1, 0) = \{q_2, q_3\}$, $\delta(q_1, 1) = \{q_1\}$
$\delta(q_2, 0) = \{q_1, q_2\}$, $\delta(q_2, 1) = \phi$
$\delta(q_3, 0) = \{q_2\}$, $\delta(q_3, 1) = \{q_1, q_2\}$
Construct equivalent DFA

Solⁿ

state	Input 0	Input 1
$\rightarrow q_1$	q_2, q_3	q_1
q_2	q_1, q_2	ϕ
(q_3)	q_2	q_1, q_2

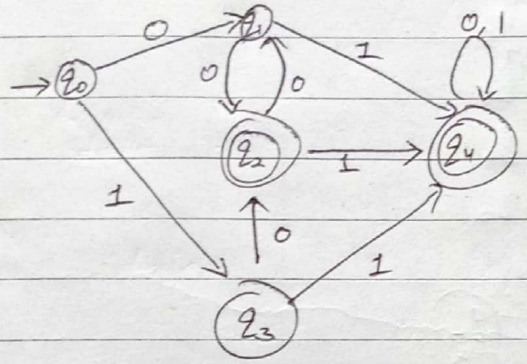
Equivalent table

state	Input 0	Input 1
ϕ	ϕ	ϕ
$[q_1]$	$[q_2, q_3]$	$[q_1]$
$[q_2, q_3]$	$[q_1, q_2]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_1, q_2, q_3]$	$[q_1]$
$[q_1, q_2, q_3]$	$[q_1, q_2, q_3]$	$[q_1, q_2]$



Contruction of minimum automaton

Example: Construct a min. state automaton equivalent to the finite automaton given in Fig



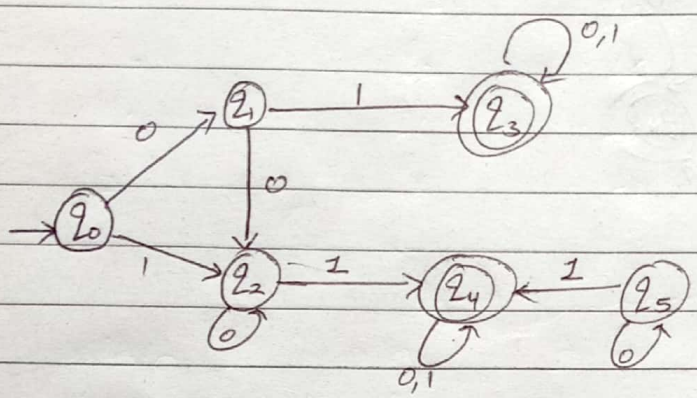
Soln:

state	Input 0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_4
(q_2)	q_1	q_4
q_3	q_2	q_4
(q_4)	q_4	q_4

state	Input 0	1	
$\rightarrow q_0$	q_{1_1}	q_{3_1}	Group 1
q_1	q_{2_2}	q_{4_2}	
q_3	q_{2_2}	q_{4_2}	
(q_2)	q_{1_1}	q_{4_2}	Group 2 (final state)
(q_4)	q_{4_2}	q_{4_2}	

state	Input 0	1
→[q_0]	[q_1, q_3]	[q_1, q_3]
[q_1, q_3]	[q_2]	[q_4]
[q_2]	[q_1, q_3]	[q_4]
[q_4]	[q_4]	[q_4]

Q)



Soln

state	Input 0	1
→ q_0	q_1	q_2
q_1	q_2	q_3
q_2	q_2	q_4
(q_3)	q_3	q_3
(q_4)	q_4	q_4
q_5	q_5	q_4

state	Input 0	1	
→ q_0	q_{1_1}	q_{2_1}	Group 1
q_1	q_{2_1}	q_{3_2}	
q_2	q_{2_1}	q_{4_2}	
q_5	q_{5_1}	q_{4_2}	
q_3	q_{3_2}	q_{3_2}	Group 2
q_4	q_{4_2}	q_{4_2}	

Note:- The state q_5 play absolutely no role in the automaton since it can never be reached from the initial state q_0. Such a state is inaccesible, and it can be removed along with all transition relating to it without affecting the language accepted by the automaton

Now

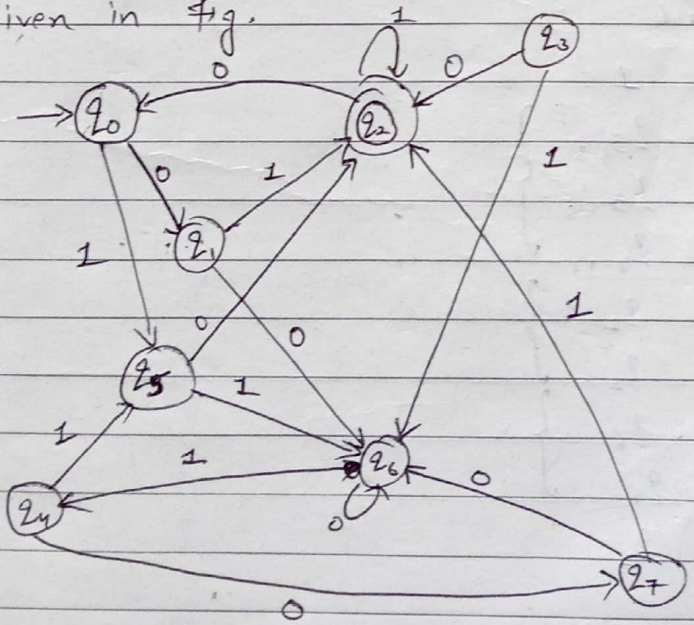
State	Input 0	Input 1
$\rightarrow q_0$	q_1	q_2
q_1	q_2	q_3
q_2	q_2	q_4
q_3	q_3	q_3
q_4	q_4	q_4

Agr 2 state same group me hai to vo merge ho jaengi

State	Input 0	Input 1
$[q_0]$	$[q_1, q_2]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_2, q_4]$	$[q_3, q_4]$
$[q_3\ q_4]$	$[q_3, q_4]$	$[q_3, q_4]$

Imp!

Q-1 Construct min. state automaton equivalent to the finite automaton given in fig.



Soln

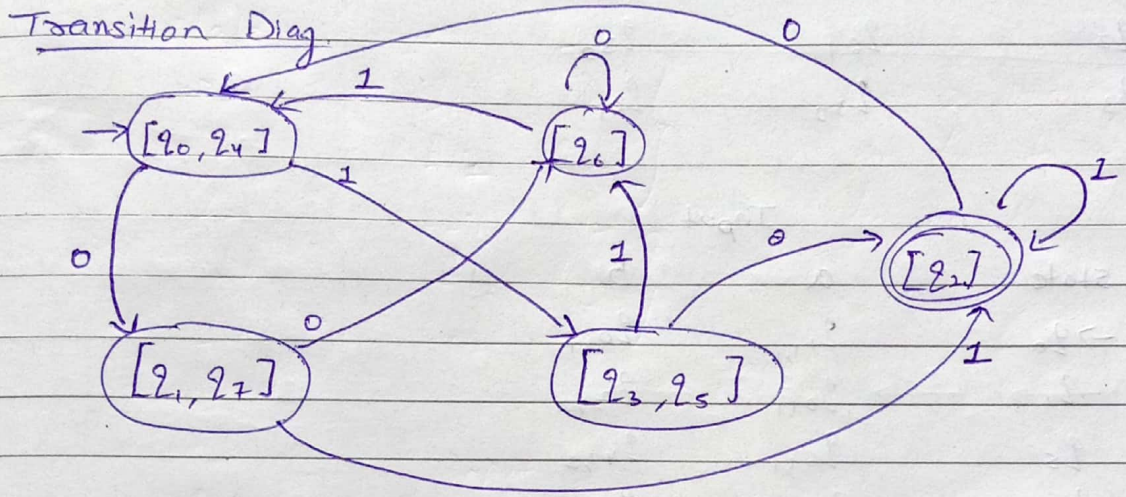
State	Input 0	Input 1
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
(q_2)	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

State	Input 0	Input 1	
$\rightarrow q_0$	$q_{1,1}$	$q_{5,1}$	Group 1
q_1	$q_{6,1}$	$q_{2,2}$	
q_3	$q_{2,2}$	$q_{6,1}$	
q_4	$q_{7,1}$	$q_{5,1}$	
q_5	$q_{2,2}$	$q_{6,1}$	
q_6	$q_{6,1}$	$q_{4,1}$	
q_7	$q_{6,1}$	$q_{2,2}$	
q_2	$q_{0,1}$	$q_{2,2}$	Group 2

State	Input 0	Input 1	
$\rightarrow q_0$	$q_{1,2}$	$q_{5,3}$	
q_4	$q_{7,2}$	$q_{5,3}$	
q_6	$q_{6,1}$	$q_{4,1}$	
q_1	q_6	q_2	
q_7	q_6	q_2	
q_3	q_2	q_6	
q_5	q_2	q_6	
q_2	q_0	q_2	

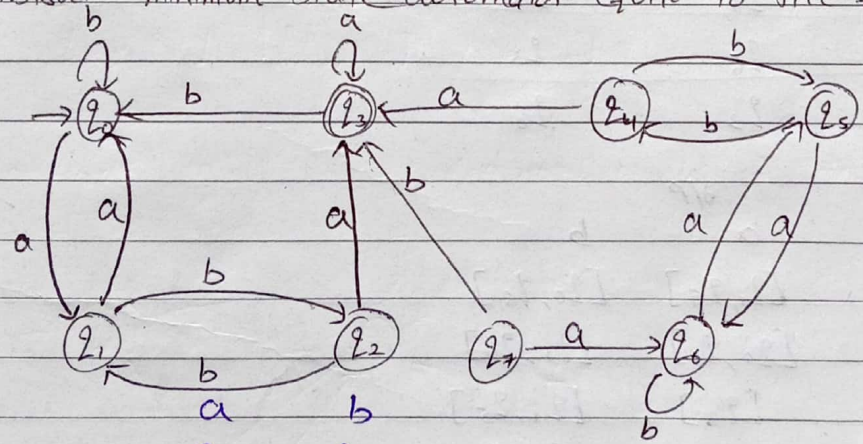
state	Input 0	1
$[q_0, q_4]$	$[q_1, q_2]$	$[q_3, q_5]$
$[q_6]$	$[q_6]$	$[q_0, q_4]$
$[q_1, q_7]$	$[q_6]$	$[q_2]$
$[q_3, q_5]$	$[q_2]$	$[q_6]$
$[q_2]$	$[q_0, q_4]$	$[q_2]$

Transition Diag



Input

Q1 Construct minimum state automaton equil. to the finite automaton



	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
q_7	q_6	q_3

Input

state	a	b
$\to q_0$	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
q_7	q_6	q_3
q_3	q_3	q_0

Input

state	a	b
$\to q_0$	$q_{1,1}$	$q_{0,1}$
q_1	$q_{0,1}$	$q_{2,2}$
q_5	$q_{6,1}$	$q_{4,2}$
q_6	$q_{5,1}$	$q_{6,1}$
q_2	q_3	q_1
q_4	q_3	q_5
q_7	q_6	q_3
q_3	q_3	q_0

o/p

state	a	b
$[q_0, q_6]$	$[q_1, q_5]$	$[q_0, q_6]$
$[q_1, q_5]$	$[q_0, q_6]$	$[q_2, q_4]$
$[q_2, q_4]$	$[q_3]$	$[q_1, q_5]$
$[q_7]$	$[q_0, q_6]$	$[q_3]$
$[q_3]$	$[q_3]$	$[q_0, q_6]$

Finite automata with outputs

The finite automata which we considered in earlier have binary output that is they accept the string or not accept the string. This acceptability was decided on the basis of the string reach from initial state to final state. Now we remove this restriction & consider the model where the o/p can be select from some other alphabet. The value of o/p function $z(t)$ in the most general case is a funct^n of the present state $q(t)$ & the present i/p $x(t)$ i.e.

$$z(t) = \lambda(q(t), x(t))$$

where λ is called o/p function. This generalised model is called "Mealy machine". If the o/p funct^n $z(t)$ depends only on the present state & is independent of the current i/p, the o/p funct^n may be written as

$$z(t) = \lambda(q(t))$$

This restricted model is called 'Moore Machine'.

Now we give the most general definition of these machines

Moore Machine

It is a 6 Tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where.

(i) Q is a finite set of state
(ii) Σ is a input alphabet
(iii) Δ is a output alphabet
(iv) δ is a transition funct^n which maps $Q \times \Sigma$ into Q
(v) λ is a output funct^n which map Q into Δ
(vi) q_0 is a initial state

Eg:- The given table give a moore machine, the initial state q_0 is marked with an arrow. The given table define δ and λ

Present state	Next state $a=0$	Next state $a=1$	output
$\rightarrow q_0$	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

For the i/p string 0111 the transition state is given by

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

the output string is 0001

Mealy Machine

A mealy machine is a 6 tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where

(i) Q is a finite set of state
(ii) Σ is a input alphabet
(iii) Δ is a output alphabet
(iv) δ is a transition funct^n which maps $Q \times \Sigma$ into Q
(v) λ is a output funct^n which maps $Q \times \Sigma$ into Δ
(vi) q_0 is the initial state.

The above definition all the symbol except λ have meaning as in the moore machine The only diff in o/p funct^n which map $Q \times \Sigma$ into Δ

Eg:- The given table give a Mealy Machine, the initial state is marked as arrow

present state	Next state			
	a = 0		a = 1	
	state	output	state	output
→ 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

for the input string 0011, the transition of state is

$\rightarrow q_1 \xrightarrow{0} q_3 \xrightarrow{0} q_2 \xrightarrow{1} q_4 \xrightarrow{1} q_3$

the o/p string is 0100

Note ⇒ For a moore machine if the input string is of length 'n', the o/p string is of length n+1. The first o/p is $\lambda(q_0)$ for all o/p string. In the case of melay machine if the input string is of length 'n', the o/p string is also of the same length.

Transforming Mealy Machine into Moore Machine

Example:- Consider the mealy machine described by below table. Contruct Moore Machine which is equivalent to mealy machine

Present	Next state a=0		Next state a=1	
State	state	o/p	state	o/p
→ q_1	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_4	1
q_4	q_4	1	q_3	0

Soln 1

Present state	Next state a=0 state	a=0 output	a=1 state	a=1 output
→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

Next state

Present state	a=0	a=1	output
→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

Next state

Present state	a=0	a=1	output
→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

Eg: Consider the mealy machine described by below table. Construct a moore machine which is equivalent to Mealy Machine.

Present state	Next state a=0	o/p	Next state a=1	o/p
→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

Soln:

Present State	Next state a=0	o/p	Next state a=1	o/p
q_{10}	q_{11}	1	q_2	0
q_{11}	q_{11}	1	q_2	0
q_2	q_{41}	1	q_{41}	1
q_3	q_2	1	q_3	1
q_{40}	q_3	0	q_{11}	1
q_{41}	q_3	0	q_{11}	1

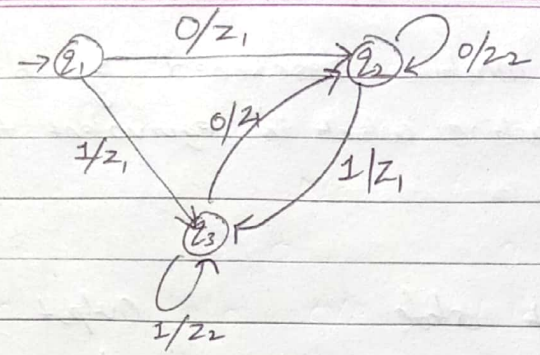
Present state	Next state a=0	Next state a=1	output
→q_{11}	q_{11}	q_2	1
q_2	q_{41}	q_{41}	1
q_3	q_2	q_3	0
q_{41}	q_3	q_{11}	1
q_0	q_{11}	q_2	0
q_{10}	q_{11}	q_2	0 or 1
q_{11}	q_{11}	q_2	0
q_{41}	q_{41}	q_{41}	1
q_3	q_2	q_3	0 or 1
q_{40}	q_3	q_4	
q_{41}	q_3	q_4	

Eg:-

Present state	a = 0 state	a = 0 o/p	a = 1 state	a = 1 o/p
$\rightarrow q_1$	q_3	1	q_2	1
q_2	q_1	0	q_4	1
q_3	q_2	0	q_1	1
q_4	q_4	0	q_3	1

Soln^n P.S. a = 0 a = 1

Eg:-



Solu^

Present state	Next state a=0		Next state a=1	
	state	o/p	state	o/p
→q_1	q_2	z_1	q_3	z_1
q_2	q_3	z_2	q_3	z_1
q_3	q_2	z_1	q_3	z_2

Present state	Next state a=0		Next state a=1	
	state	o/p	state	o/p
→q_1	q_2z_1	z_1	q_3z_1	z_1
q_2z_1	q_2z_2	z_2	q_3z_1	z_1
q_2z_2	q_2z_2	z_2	q_3z_1	z_1
q_3z_1	q_2z_1	z_1	q_3z_2	z_2
q_3z_2	q_2z_1	z_1	q_3z_2	z_2

Present state	Next state a=0	Next state a=1	Output
→q_0	q_2z_1	q_3z_1	z_1
q_1	q_2z_1	q_3z_1	z_1
q_2z_1	q_2z_2	q_3z_1	z_1
q_2z_2	q_2z_2	q_3z_1	z_2
q_3z_1	q_2z_1	q_3z_2	z_1
q_3z_2	q_2z_1	q_3z_2	z_2

Q1 Consider the Moore machine described by below table Construct a Mealy machine which is equivalent to moore machine.

Present state	Next state a=0	Next state a=1	Output
→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

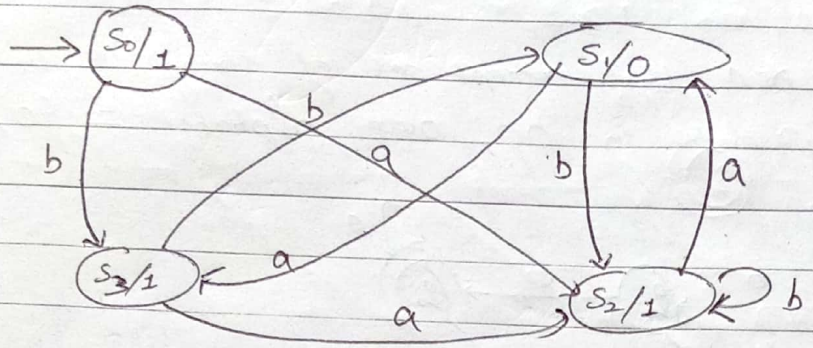
Soln

Present state	a=0 state	a=0 o/p	a=1 state	a=1 output
→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. If two states have identical transition i.e. the row corresponding to these two states are identical, then we can delete one of them

Present state	a=0 state	a=0 o/p	a=1 state	a=1 o/p
→q_0	q_2	0	q_1	1
q_1	q_1	1	q_2	0
q_2	q_2	0	q_2	0

Q-1 Convert Moore to Melay.



Solu^n

Present state	Next state a=a	Next state a=b	Output
→S_0	S_2	S_3	1
S_1	S_3	S_2	0
S_2	S_1	S_2	1
S_3	S_2	S_1	1

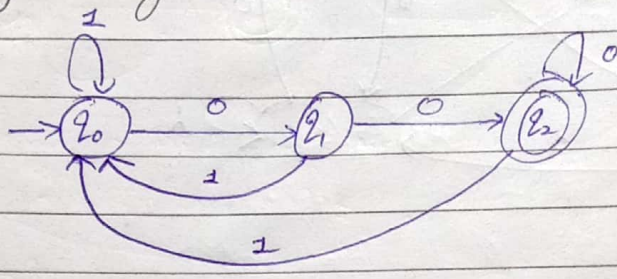
Present state	Next state I=a state	I=a o/p	Next state I=b state	I=b o/p
→S_0	S_2	1	S_3	1
S_1	S_3	1	S_2	1
S_2	S_1	0	S_2	1
S_3	S_2	1	S_1	0

Present State	Next state I=a state	I=a o/p	Next state I=b state	I=b o/p
→S_0	S_2	1	S_3	1
S_2	S_0	0	S_2	1
S_3	S_2	1	S_0	0

Design of DFA

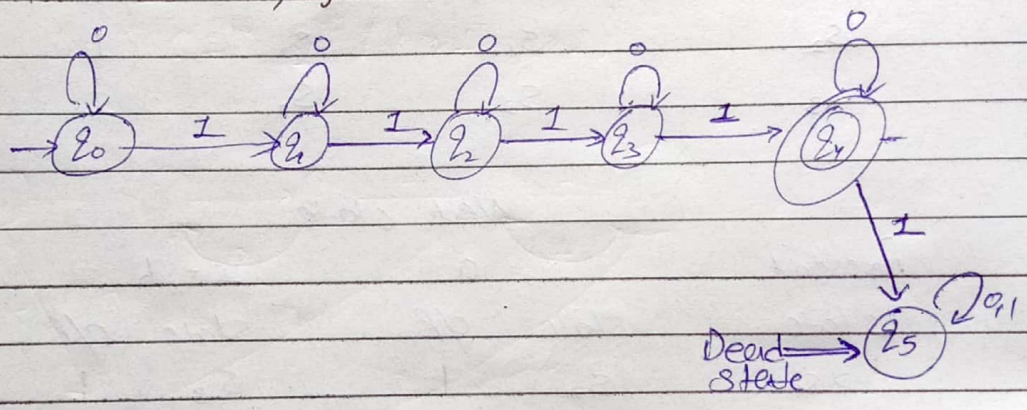
Example: Design a DFA that accept set of string such that every string ends in 00, over alphabet $\Sigma = \{0, 1\}$.

Solu^n



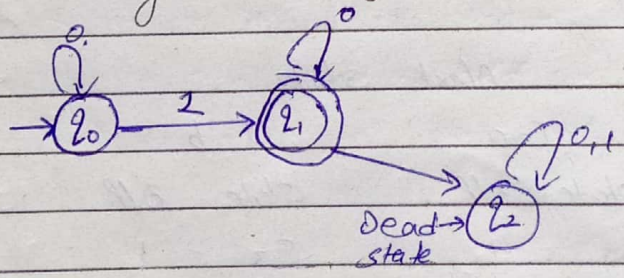
Eg: Design a DFA that accept set of string such that every string contains exactly four 1's over alphabet $\Sigma = \{0, 1\}$

Solu'



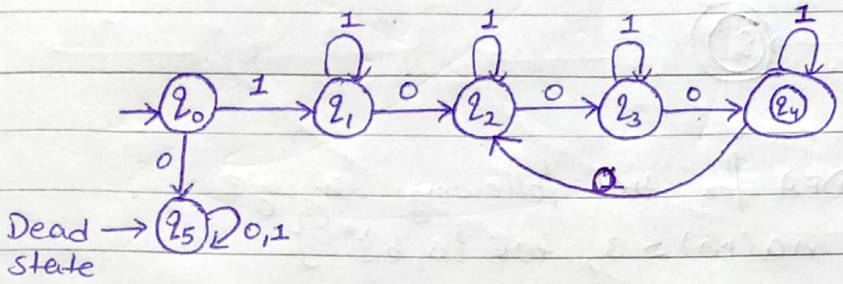
Eg: Design an DFA that accept set of string such that every string containing exactly 1 exactly over alphabet $\Sigma = \{0, 1\}$

Solu'



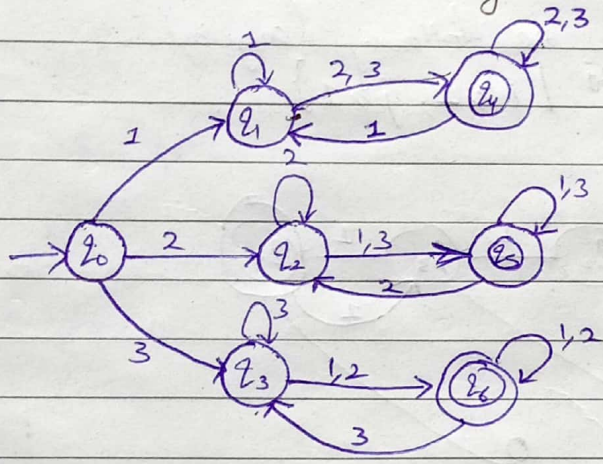
Example: Design a DFA that accept set of string over alphabet $\Sigma = \{0,1\}$ set of string starting with 1 and such that the no. of 0's is divisible by 3

Solu^n



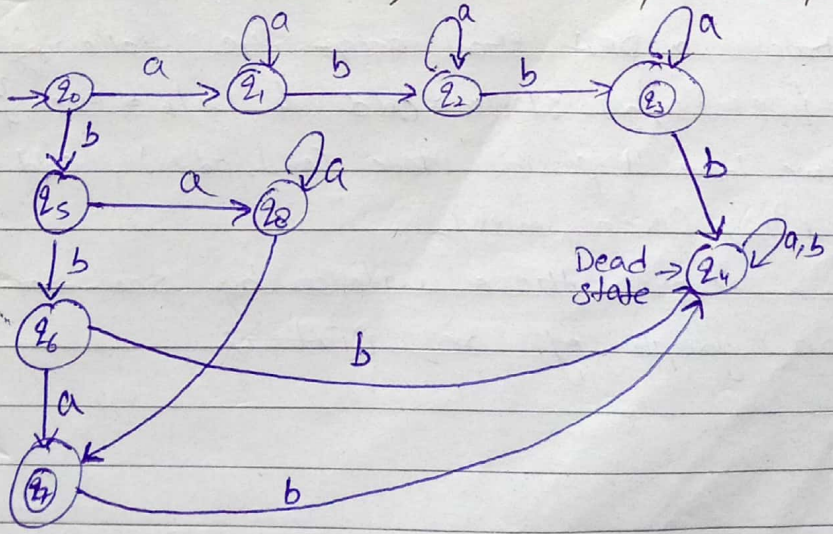
Example: Design a DFA that accept the following language.
$L = \{ x \in \{1,2,3\}^* \; x \text{ begin and end with diff. symbol.}$

Solu^n

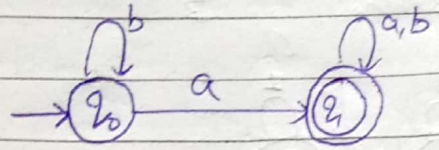


Example: Design a DFA that accept the following language.
$L = \{ w : n_a(w) \geq 1, \; n_b(w) = 2, \; w = \{a,b\}^* \}$

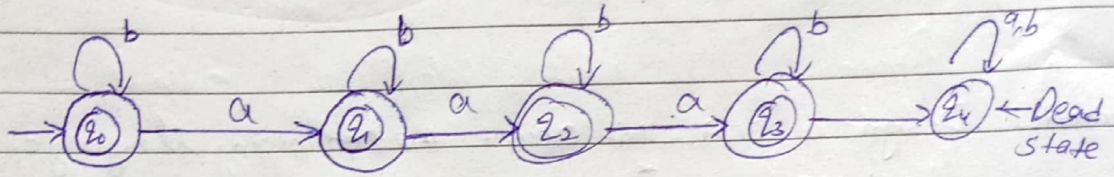
Solu^n



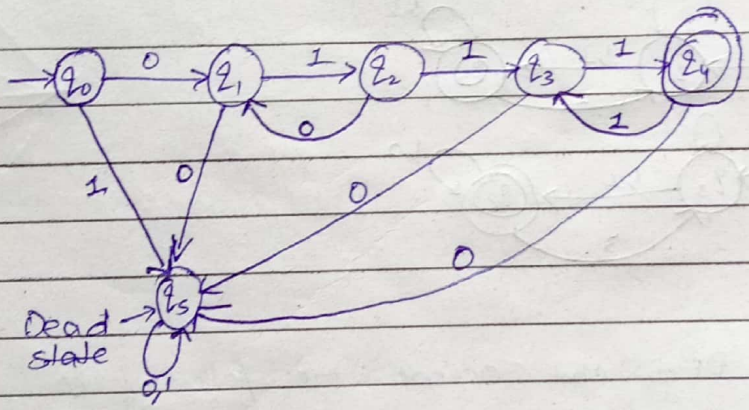
Example: Design a DFA for the following language
$L = \{w : n_a(w) \geq 1, w \in \{a,b\}^*\}$



Example: Design a DFA for the following language
$L = \{w : n_a(w) \leq 3, w \in \{a,b\}^*\}$

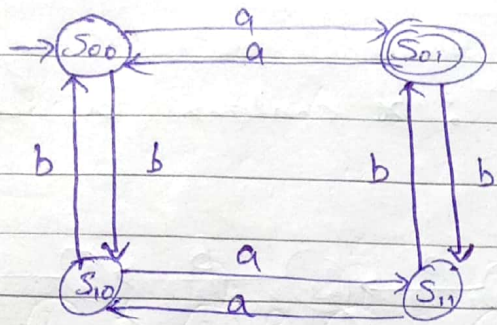


Example: Design a DFA for the following language
$L = \{(01)^i 1^{2j} \mid i \geq 1, j \geq 1\}$



Example: Construct a DFA that accept the following language
$L = \{x \in \{a,b\} : |x|_a = \text{odd} \ \& \ |x|_b = \text{even}\}$

Soln: clearly we have 4 possible state (odd, odd), (odd, even), (even, odd), (even, even)
let's we denote these states by $s_{oo}, s_{oe}, s_{eo}, s_{ee}$
Transition graph of req DFA is



Example: Construct a DFA that accept the following language

$$L = (ab)^* (ba)^* \cup aa^*$$

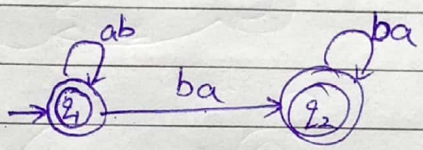
Solu^n

$$L = L_1 \cup L_2$$

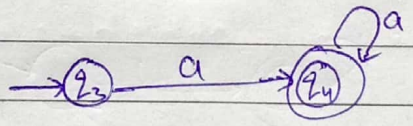
$$L_1 = (ab)^* (ba)^*$$

$$L_2 = aa^*$$

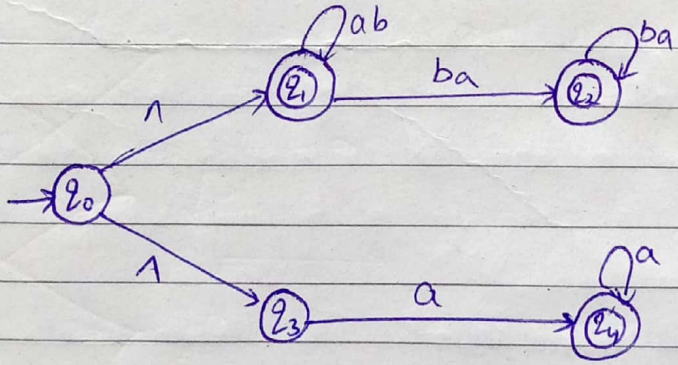
DFA for L_1



DFA for L_2



Now we can combine L_1 and L_2 introduce another stat q_0 & connect both with null transition as follows:



Example: Construct a DFA that accept the following language.
$L = \{x \in (a,b,c)^* : x$ contains a substring $acaba\}$.

Solu?