August 3, 2022
Wednesday
UNIT: 1

More than one character
read it from left to right.

1

Finite Automaton

(Countable)
- device or machine in which we build logic
- input in form of string

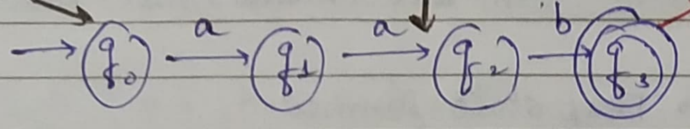
two states accept or reject.

Finite automata are used to recognized patterns

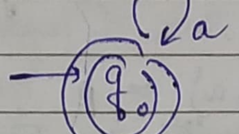
L to R

$L = \{a\ a\ b\}$ (single arrow)

States — Transitions — final / accepting state (2 circle)



Initial state usually denoted with q_0 with an arrow with no source.



→ String that accepts all a's

* String get accepted when it's start from initial state and reached to final state.

Terminologies:

1) Symbols: entity or individual characters or objects which can be any letter, alphabet or any picture. Ex: 1, a, b, #

2) Alphabets: finite set of symbols. It is denoted by Σ
   $\Sigma \rightarrow$ I/P symbol.
   $\Sigma = \{a, b\}$ (single and comma separated)

3) String: finite collection of symbols from the alphabet. Denoted by w.

- String with zero occurrences of symbols is known as empty string. It is represent by ε.

Ex: If $\Sigma = \{a, b\}$, various string that can be generated from Σ are $\{ab\}, \{aa\}, \{aaa\}, \{bb\}, \{bbb\}, \{ba\}, \{aba\}, \ldots\}$.

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

Length

- W = 101 , ~~No.~~ of string |w| = 3
  No. of symbols.

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

Types of automata:

→ finite automata
- DFA (Deterministic finite Automata)
- NFA (Non-deterministic finite Automata)

Note: In case of reject ⟶ Trap state formed

Transition state Diagram:

$\delta(q, a) = p$
↳ transition function

δ(Present state, I/P symbol) = Next state

$\delta(q, a) = p$ $\quad q \xrightarrow{a} p$

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

$\delta(q_0, 1) = q_1$ and $\delta(q_1, 0) = q_2$

L_1 = {set of string of length 2} = {aa, bb, ba, bb} finite language
L_2 = {set of all strings starts with 'a'} = {a, aa, aaa, abb, abbb} Infinite language

Date ____
Page ____

3

Que Write a language having string that accepts all 01. $\Sigma = \{0, 1\}$

Que Write a language having string that starts with a and ends with bb. $\Sigma = \{a, b\}$

Que ______ accepts a combination of a & b $\Sigma = \{a, b\}$

1) L = {0, 1, 001, 1~~0~~0, 1010, 1001, 101, 010, ...}

2) L = {abb, ababb, aabb, aaabb, ababb, abbabb, ...}

3) L = {ab, ba, aab, aba, bab, bba, abb, ...}

Finite automata :

It takes the string of symbol as input and changes its state accordingly. When the desired symbol is found, then the transition occur.

- At the time of transition, the automaton can either move to the next state or stay in the same state.
- Finite automata have two states, Accept and reject state. When the input string is processed successfully and the automata reached its final state, then it will accept.

Theory of automata :

→ Theory of automata is the study of abstract machines is called the automata and the computation problems that can be solved using these machines.

→ An automata is defined as a system where energy, materials and information are transformed, transmitted and used for performing some func. without direct participation of man

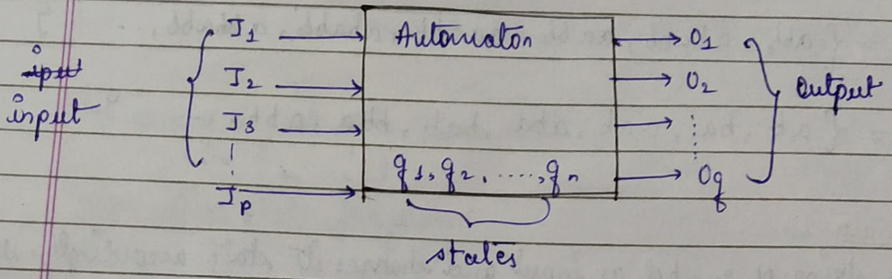
→ In computer science the term 'automaton' means 'discrete automaton'.

Date
Page

The automaton consists of states and transitions. The state is represented by circles and the Transitions is represented by arrows.

Automata is the kind of machine which takes some string as inputs and this input goes through a finite number of states and may enter in the final state.

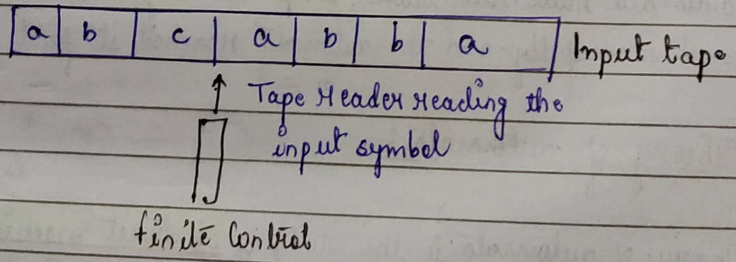
Model of a discrete automaton



gust 4, 2022 hursday

Finite Automata Model:

Finite automata can be represented by input tape and finite control



Input Tape: It is a ~~similar~~ linear tape having some number of cells. Each input symbol is placed in each cell

Finite ~~Automata~~: Control: The finite control decides the next state on receiving particular input from input tape. The tape header reads the cells one by one from left ~~to~~ to right, and at a time only one input symbol is read

Trap state: other conditions / possibilities

Note: $q_n \rightarrow$ is not always a final state.

Formal Definition of finite Automaton

A DFA can be represented by 5 tuples:

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

Q = finite non-empty set of states

Σ = finite non-empty set of input alphabets

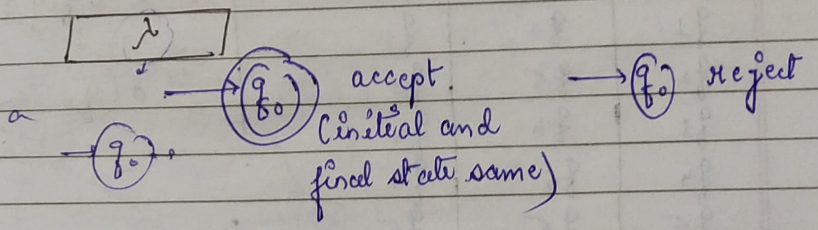
δ = (direct) transition function that maps $\boxed{Q \times \Sigma \rightarrow Q}$

q_0 = initial state $(q_0 \in Q)$

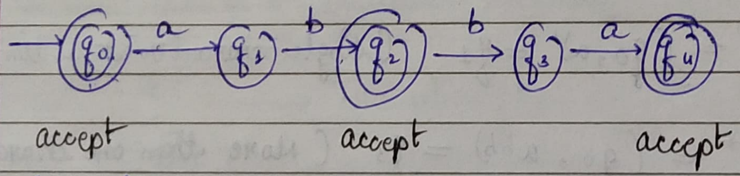
F = set of final states. Q $(F \subseteq Q)$

Note: Always take no. of possibilities in acceptence and rejection.

string :



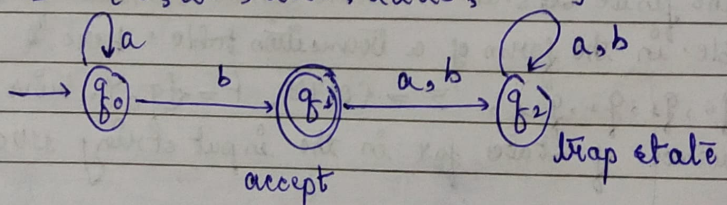
Ex 1: $L(M) = \{\lambda, ab, abba\}$



$F = \{q_0, q_2, q_4\}$

Ex 2: $L(M) = \{a^n b : n \geq 0\}$

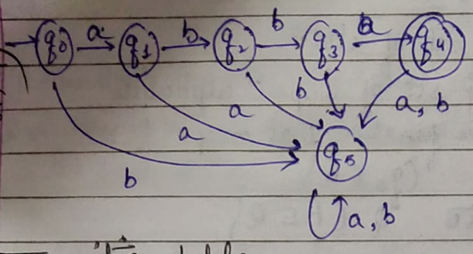
$L = \{b, ab, aab, aaab, \ldots\}$



Transition function δ

$\delta : Q \times \Sigma \longrightarrow Q$

$\delta(q_1, a) = q_5$
$\delta(q_5, a) = q_5$
$\delta(q_0, a) = q_1$
$\delta(q_4, b) = q_5$



Transition table

δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5

$\delta^*(q, ya)$

$\delta(\delta^*(q, y), a)$

$\delta(q_0, a)$

Imp **Extended transition function (δ^*)**

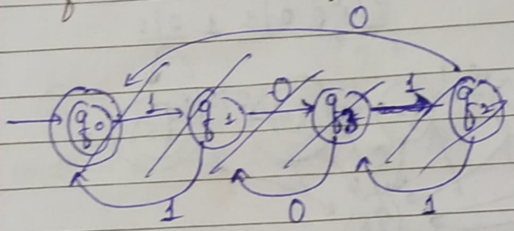
$\delta = (q_0, a) = q_1$ (single character at a time)

$\delta^* = (q_0, \underset{\Sigma^*}{abb}) = q_3$ (More than one character at a time

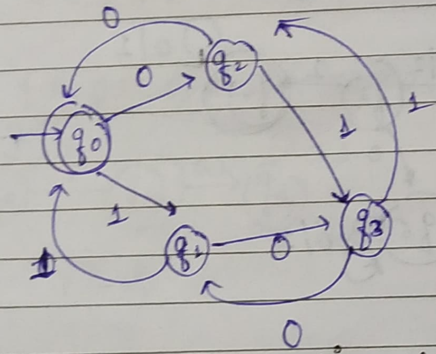
$\delta^* = \boxed{Q \times \Sigma^* \longrightarrow Q}$

Que Consider the finite ~~set~~ state machine whose transition func. δ is given by table. in the form of a transition table. Here, $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $F = \{q_0\}$. Give the entire sequence of state for in the input string 110001.

state	Input 0	Input 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



110001 → rejected



Notations used in the transition diagram.

q_1 state

$\longrightarrow$ Transition from one state to another

start $\rightarrow q_0$ or q_0] start state

q_n or q_n] final state

Recursive Definition:

$$\delta^*(q, \lambda) = q$$

$$\delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$$

$$q \xrightarrow{w} q_1 \xrightarrow{\sigma} q'$$

$\left.\begin{array}{l} \delta^*(q, w\sigma) = q' \\ \delta(q_1, \sigma) = q' \end{array}\right\} \Rightarrow \left.\begin{array}{l} \delta^*(q, w\sigma) = \delta(q_1, \sigma) \\ \delta^*(q, w) = q_1 \end{array}\right\} \Rightarrow \delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$

August 9, 2022
Tuesday

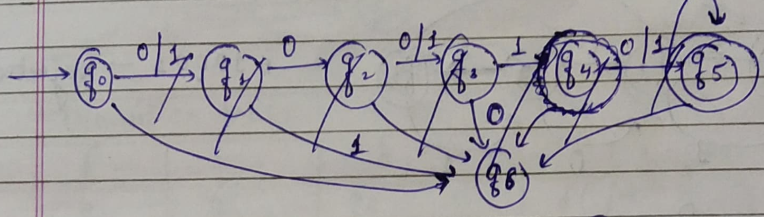
Date
Page

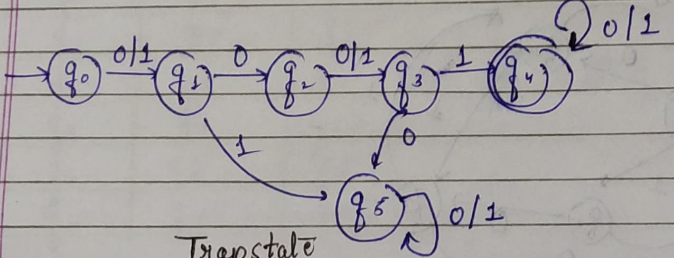
Que Construct a DFA that accepts all strings, over (0, 1) in which 2nd symbol is 0 and fourth symbol is 1

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

$L = \{0001, 1011, 0011, 1001, \ldots$

0/1 0 0/1 1 0/1



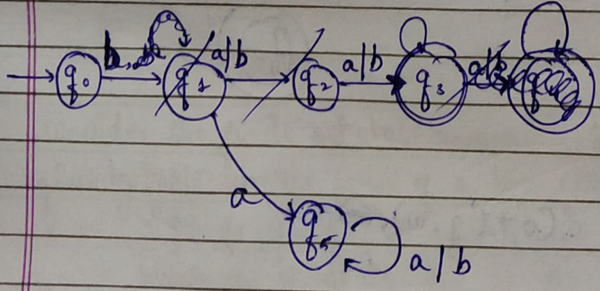


Trapstate

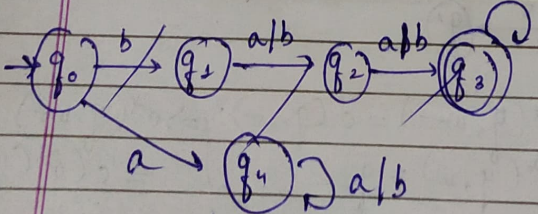
Que (2) Construct a DFA that accepts all strings having substring bba or bab over {a, b}

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

$L = \{bba, bab, abba, abab, \ldots\}$



ba b

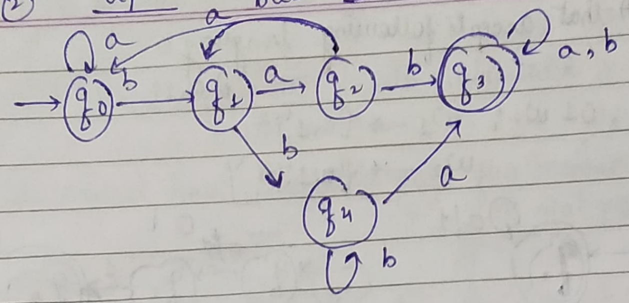


No. of states = Minimum string length + 1

9

(1) $\underline{a|b}$ bba $\underline{a|b}$ OR

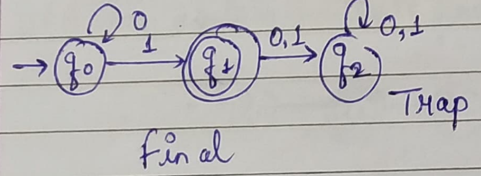
(2) $\underline{a|b}$ bab $\underline{a|b}$



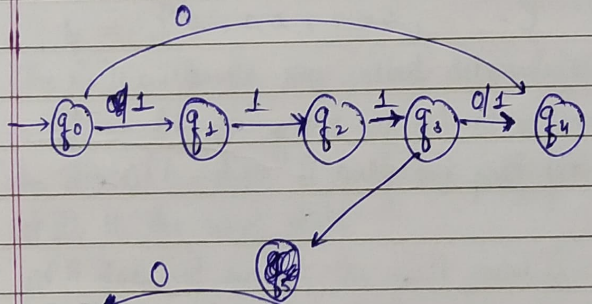
Que (3) Construct a DFA that accepts the foll. language.?

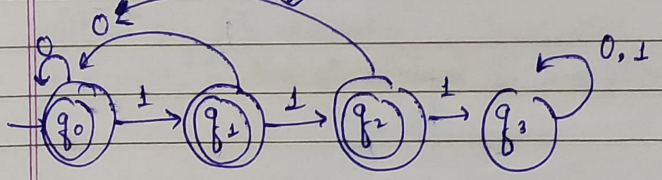
$$L = \{0^n 1 : n \geq 0\}$$

$$L = \{1, 01, 001, 0001, 00001, \ldots\}$$



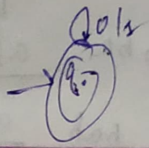
Que (4) Construct a DFA that accepts all strings that does not contain 3 consecutive 1's over {0,1} $\Sigma = \{0,1\}$





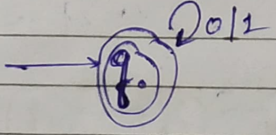
10

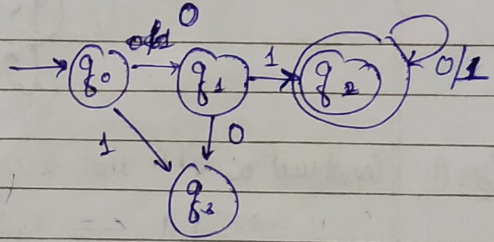
Date
Page

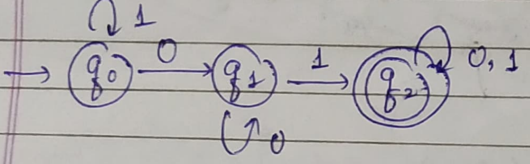


Ques (5) Construct a DFA that accepts following language $\Sigma\ 0,1$

$$L = \{ w_1 , 01\, w_2 : \; w_1 \rightarrow \{0,1\}^* \; w_2 \rightarrow \{0,1\}^* \}$$



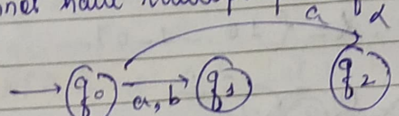




Types of Automata:

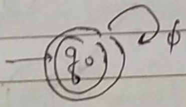
1) DFA (Deterministic finite Automata)
↳ Uniqueness

- We cannot have multiple path for same input symbol.



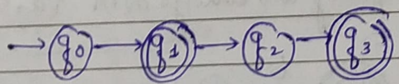
← Not possible

- DFA does not ~~have~~ accept the null move.



(Not possible in case of DFA

- DFA can contain multiple final states



- Acceptance of languages: Reaching to final state

DFA to accept zero or more "a"

$a^n, n \geq 0$

$L = \{a, aa, aaa, \ldots\}$

- The finite automata are called deterministic finite automata if the m/c is read an i/p string one symbol at a time.
- In the DFA, there is only one path for specific i/p from the current state to the next state.
- DFA does not accept the null move i.e DFA cannot change state without any i/p character.
- It is used in lexical Analysis in computer.

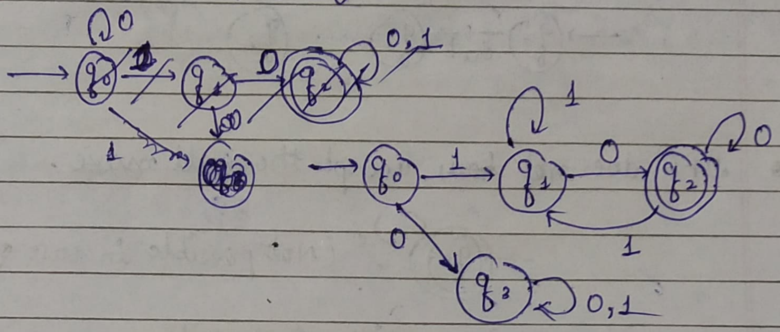
If in quest. it is written that starting with —, don't put self loop on intial state

Note: ending with —, don't put self loop on final state

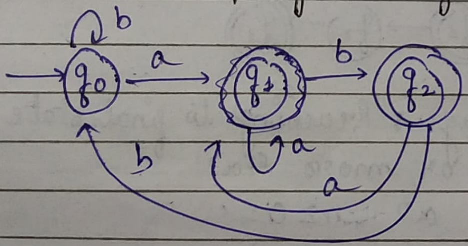
No. of states = |Minimum string| + 1
(Trap state will not be included in No. of states)

1) Construct a DFA accepting all strings over {0, 1} that starts with 1 ends with 0

$\Sigma = \{0,1\}$, $L = \{10, -1-0\}$...



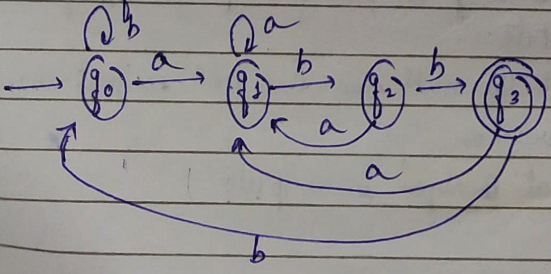
2) Construct DFA accepting all strings over {a, b} ending with a, b



Construct a dfa accepting all string over {a, b} ending with abb

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

No. of states = 4



August 11, 2022
Thursday

13

Date
Page

NDFA or NFA

Non Deterministic Finite Automata

1) There can be multiple paths for same symbol is allowed.
2) It is easy to construct NFA than DFA for a given regular language.
3) Every NFA is not DFA, but each NFA can be translated into DFA.

DFA is a subset of NFA ($DFA \subseteq NFA$)

NFA is defined in the same way as DFA but with two exceptions:

i. It contain multiple next states.
ii. It contains ϵ (Null moves) transitions.

Formal Definition of NDFA:
5 - tuple $(Q, \Sigma, \delta, q_0, F)$

1. Q: finite set of states
2. Σ: finite set of symbols called the alphabets.
3. δ is the transition func. where $\boxed{\delta : Q \times \Sigma \longrightarrow 2^Q}$
4. (Here the power set of Q (2^Q) has been taken because in case of NDFA, from a state, transition can occur to any combination of Q states)
5. q_0 is the initial state from where any input is processed ($q_0 \in Q$).
6. F is a set of final state/states of Q ($F \subseteq Q$)

Graphical Representations

An NDFA is represented by digraphs called state diagram.

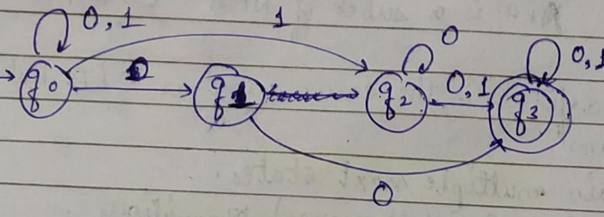
- The vertices represents the state
- The arcs labeled with an input alphabets show the transitions
- The initial state is denoted by an empty single incoming arc.
- The final state is indicated by double circles.

NFA: saving memory

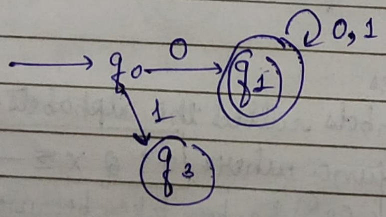
Ques ① Design NFA for given transition table.

state	0	1
$\rightarrow q_0$	q_0, q_1	q_0, q_2
q_1	q_3	ϵ
q_2	q_2, q_3	q_3
$* q_3$	q_3	q_3

starts with → loop at end
ends with → loop at first
substring → loop at first or last.

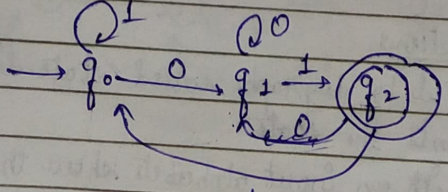


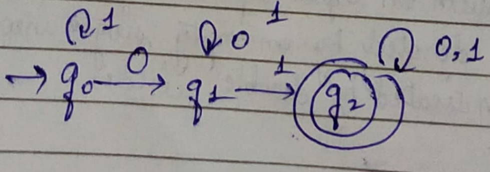
② Construct NFA, L = {set of all strings that starts with 0}, with $\Sigma = \{0, 1\}$



③ Construct NFA with $\Sigma = \{0, 1\}$ that accepts all strings with 01.

111000



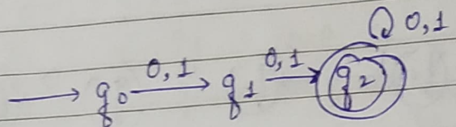


10 1001 100101

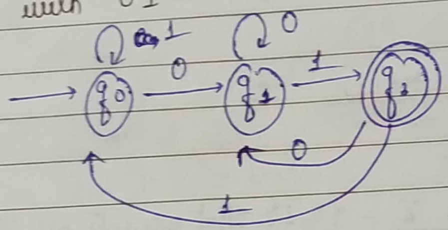
15

Date
Page

(4) Construct NFA with $\Sigma = \{0, 1\}$ that accepts all string of length atleast 2



(5) Design an NFA with $\Sigma = \{0, 1\}$ that accepts all strings ending with 01



***

(5) DFA vs NDFA:

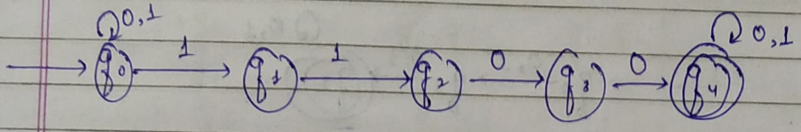
DFA	NDFA
i. The transition from a state is to a single particular next state for each input symbol. Hence it is called deterministic.	The transition from a state can be of multiple next states for each input symbol. Hence it is called non-deterministic.
ii. Empty string transitions are not seen in DFA.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

Date ______
Page ______

Practice questions:

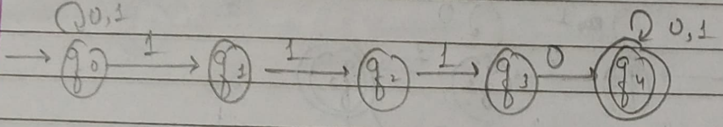
(6) Construct NFA with $\Sigma = \{0, 1\}$ in which double '1' is followed by double '0'.

$L = \{ 1100 , 011000 , \dots \}$



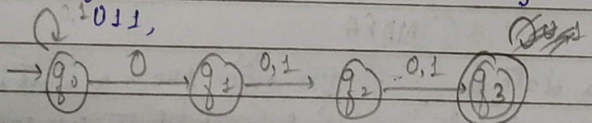
(7) Construct NFA in which all the string contain a substring 1110.

$L = \{ 1110 , 01110 , 111100 , \dots \}$

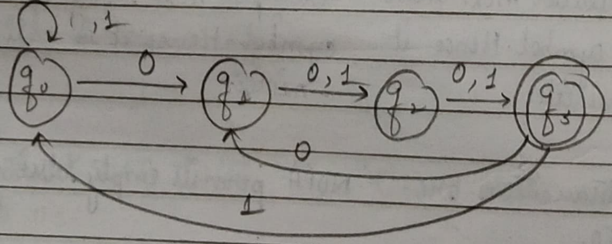


(8) Design an NFA with $\Sigma = \{0, 1\}$ that accept all string in which the third symbol from the right end is always 0.

0 $L = \{1000 , 0011 , 1011 , \dots \}$
1011,



or



0 1111

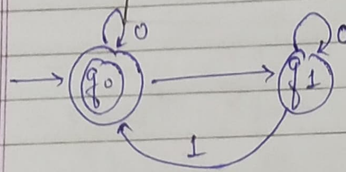
Date
Page

Equivalence of NFA and DFA:

if language is same in both the case.

Ques Convert this NFA to DFA

δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_1	q_0, q_1

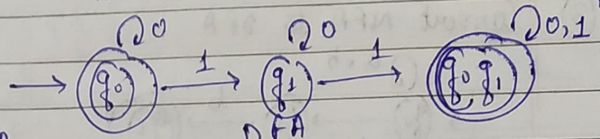


DFA $\Rightarrow$

δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_0	$\{q_0 q_1\}$
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_0, q_1\}$

$\{q_0, q_1\}$

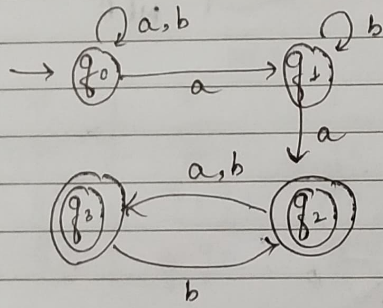
DFA $\subseteq$ NFA



Note: if new state generated also contain q_0, then it will also be final state

$\{q_0, q_1\} \Longrightarrow \{q_0, q_1\}, \{q_0, q_1\}$

Ques (2) Convert NFA to DFA

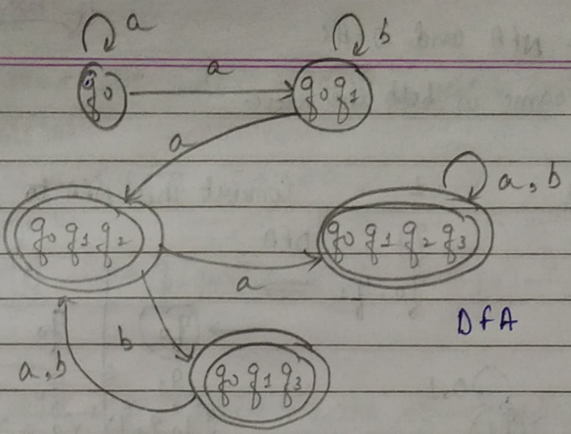


δ	a	b
$\rightarrow q_0$	q_1, q_0	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	ϵ	q_2

NFA

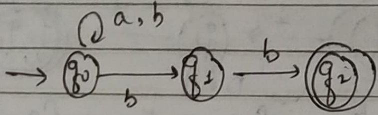
DFA

δ	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
$\{q_0 q_1\}$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1\}$
$\{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_3\}$
$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_2 q_3\}$
$\{q_0 q_1 q_3\}$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2\}$



DFA

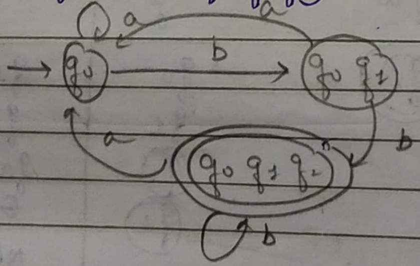
(3) Convert NFA to DFA



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

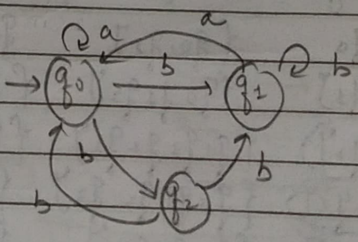
NFA

δ	a	b
$\rightarrow q_0$	q_0	$\{q_0 q_1\}$
$\{q_0 q_1\}$	q_0	$\{q_0 q_1 q_2\}$
$\{q_0 q_1 q_2\}$	q_0	$\{q_0 q_1 q_2\}$

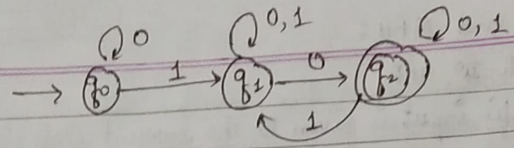


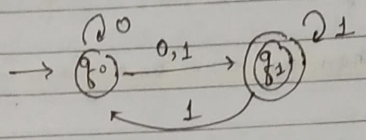
(4)

δ	a	b
$\rightarrow q_0$	$q_0 q_1$	q_2
q_1	q_0	q_1
q_2	–	$q_0 q_1$



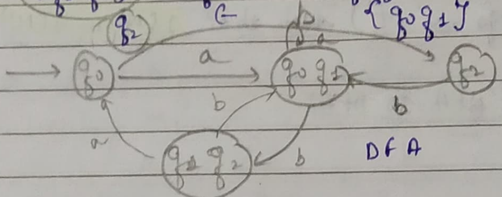
Date
Page

(5) 

(6) 

(4) →

δ	a	b
→ q_0	$\{q_0 q_1\}$	q_2
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_1 q_2\}$
$\{q_1 q_2\}$	q_0	$\{q_0 q_1\}$
q_2	ε	$\{q_0 q_1\}$



DFA

(5)

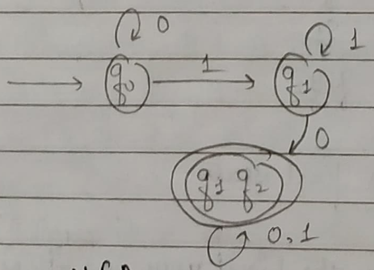
NFA

δ	0	1
→ q_0	q_0	q_1
q_1	$q_1 q_2$	q_1
q_2	q_2	$q_1 q_2$

$\xrightarrow{DFA}$

DFA

δ	0	1
→ q_0	q_0	q_1
q_1	$\{q_1 q_2\}$	q_1
$\{q_1 q_2\}$	$\{q_1 q_2\}$	$\{q_1 q_2\}$



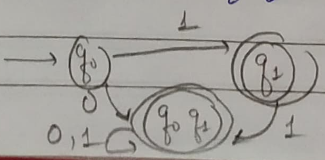
(6)

NFA

δ	0	1
→ q_0	$q_0 q_1$	q_1
q_1	ε	$q_0 q_1$

$\xrightarrow{DFA}$

δ	0	1
→ q_0	$\{q_0 q_1\}$	q_1
q_1	ε	$q_0 q_1$
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_0 q_1\}$



Date
Page

Equivalence of DFA and NFA:

Conversion from NFA to DFA

- In NFA, when a specific input is given to the current state, the machine goes to multiple states.
- It can have zero, one or more than one move on a given input symbol
- On the other hand, in DFA, when a specific input is given to the current state, the machine goes to only one state
- DFA has only one move on a given input symbol.
- Let, $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA which accepts the lang $L(M)$
- There should be equivalent DFA denoted by
  $M' = (Q', \Sigma', q_0', \delta', F')$ such that $L(M) = L(M')$

Steps for converting NFA to DFA:

Step 1: Initially $Q' = \epsilon$

Two methods given to derive it:

1) Transition table
2) Design of automata

Ex:

state	a	b
$\rightarrow q_0$	$q_0 q_1$	q_2
q_1	q_0	q_1
q_2	ϵ (Null)	$q_0 q_1$

q_0 is initial state & q_2 is final state.

i) Design a table by copying the first row as it is

δ	a	b
$\rightarrow q_0$	$q_0 q_1$	q_2
q_2		
$q_0 q_1$		

ii. Copy the states from a & b and writes them in state column.
iii. check for both states that we obtained just now for a & b
column. (checking transitions)

δ	a	b
$\rightarrow q_0$	$q_0 q_1$	q_2
q_2	$\epsilon(\phi)$	$q_0 q_1$
$q_0 q_1$	$q_0 q_1$	$q_2 q_1$

write the a & b from q_2 as it is. Then we check ~~with~~ which state we not processed & copying them down in state column and writing their transition with a & b from previous results

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

This completes the DFA table.

Theorem: For every NFA, there exists a DFA which simulates the behaviour of NFA. Alternatively, if L is the set accepted by NFA, then there exists a DFA which also accepts L

Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA accepting L. We construct a DFA M' as
$M' = (Q', \Sigma, \delta, q_0', F')$

i. $Q' = 2^Q$ (any state in Q' is denoted by $[q_1, q_2, \dots q_j]$ where $q_1, q_2, \dots q_j \in Q$);

ii. $q_0' = [q_0]$;

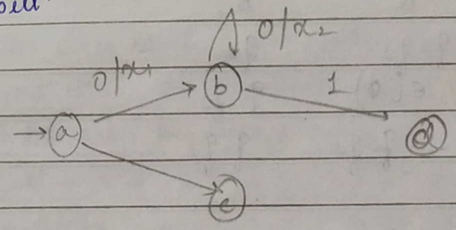
iii. F' is the set of all subsets of Q containing an element of F.

Date
Page

Finite automata: may have outputs corresponding to each transition

Mealy and Moore Machine:

- No concept of final state
- Depends on output alphabet
- Output depends on the present state as well as the present input



The value of the output function z(t) in the most general case is a func. of the present state q(t) and the present input x(t), i.e

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

where λ is called the output func. This generalized model is usually called Mealy machine. If the output func. z(t) depends only on the present state and is independent of the current input, the output function may be written as

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

This restricted model is called the Moore Machine (Convenient to use in automata).

Mealy Machine:

is an FSM whose output depends on the present state as well as the present input.

It can be described by a 6 tuples $(Q, \Sigma, O, \delta, X, q_0)$ where -

i. Q is a finite set of states.

ii. Σ is a finite set of symbols called the input alphabet.

iii. O (Δ) is a set of symbols called the output alphabet.

iv. δ is the input transition func. where $\delta: Q \times \Sigma \longrightarrow Q$

v. $X(\lambda)$ is the output transition func where $X: Q \times \Sigma \longrightarrow O$

vi. q_0 is the initial state from where any input is processed $(q_0 \in Q)$

25 0 → b   01 → I/P / O/P symbol

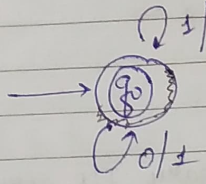
build in such away always take accepting strings

① Construct Mealy machine that produce 1's complement of any binary i/p string.

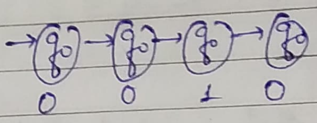
$\Sigma = \{0, 1\}$

1 1 0 1

0 0 1 0

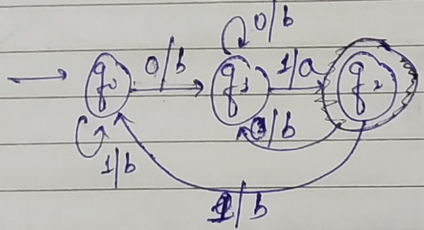


(Or)



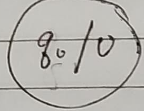
② Construct a mealy m/c that prints 'a' whenever the sequence '01' is encountered in any i/p binary string

$\underbrace{01}_{a}$ $\underbrace{00}_{b}$ $\underbrace{11}_{b}$ $\underbrace{0}_{b}$ $\underbrace{1}_{b}$



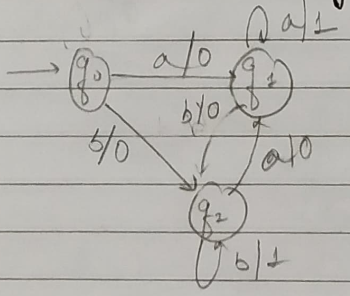
1 1 1 0 0 1

b b b b a



(OR)

③ Construct a mealy m/c accepting language consisting of strings where {a, b} and the string should end with either aa or bb.



aaab

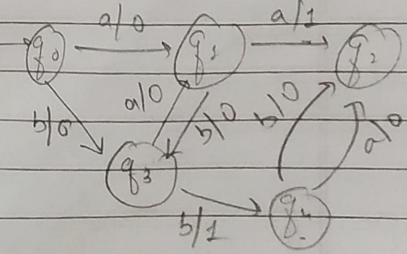
aa or bb

True : O/P Symbol as 1

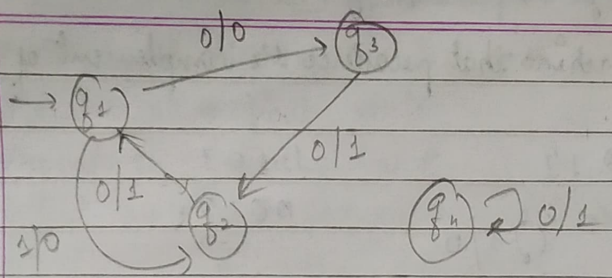
false : O/P Symbol as 0

abaa abab

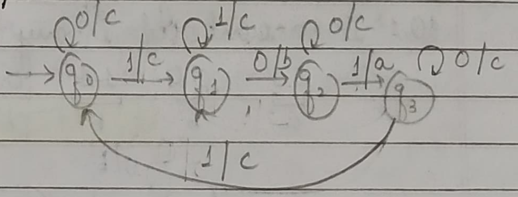
001 0000



a/o 24



Design a Mealy m/c for a binary I/P sequence such that if it has a substring 101, the m/c O/P A, if I/P has substring 110, it O/P's B, otherwise C.



if sub: 101
{ O/P: A
{
if sub: 110
{ O/P: B
}
}
else:
O/P: C

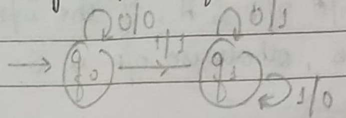
010
ccb

0110
bacb

Date ____
Page ____

Note: if input length is n then output length n (Mealy m/c)
(n+1) (Moore m/c)

Que Mealy m/c to find the 2's compliment of a number



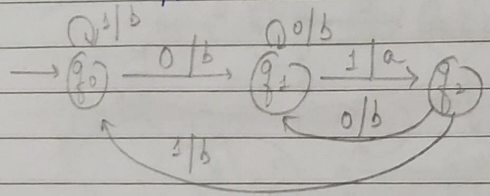
1 1 0 0
0 0 1 1
+ 1
0 1 0 0

1 1 0 0
0 1 0 0

(2) Construct a mealy m/c that prints 'a' whenever the sequence '01' is encountered in any i/p binary string.

i/p $\Sigma = \{0,1\}$
o/p $O = \{a, b\}$

1 0 0 0 0 1 1 0
b b b b b a b 0



0 1 1 0 0 1
ba

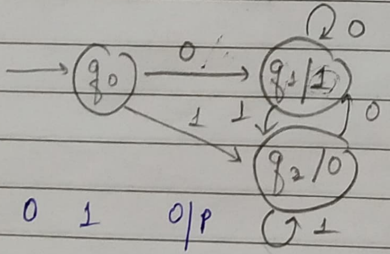
August 23, 2022
Tuesday.

Moore Machine:

Output depends on only the present state

$\boxed{P.S \mid O.P}$

1. 1's complement



0 1 1 0 1
1 0 0 1 0
q_0 q_1 q_2 q_1 q_2

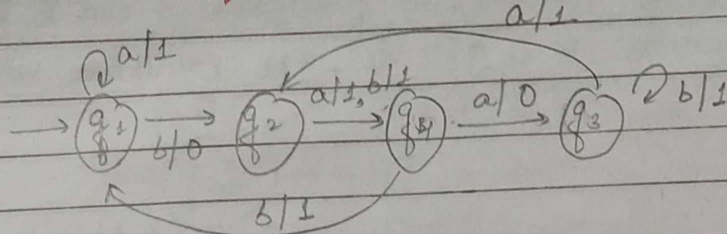
δ	0	1	O/P
q_0			—
q_1			1
q_2			0

st 24, 2022 Order doesn't matter (28)

nesday

Date
Page

Conversion from Mealy Machine to Moore Machine:



Convert it into Moore m/c

m selection
e of
ly m/c

Present state	a	o/p	b	o/p
q_1	q_1	~~0~~ 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

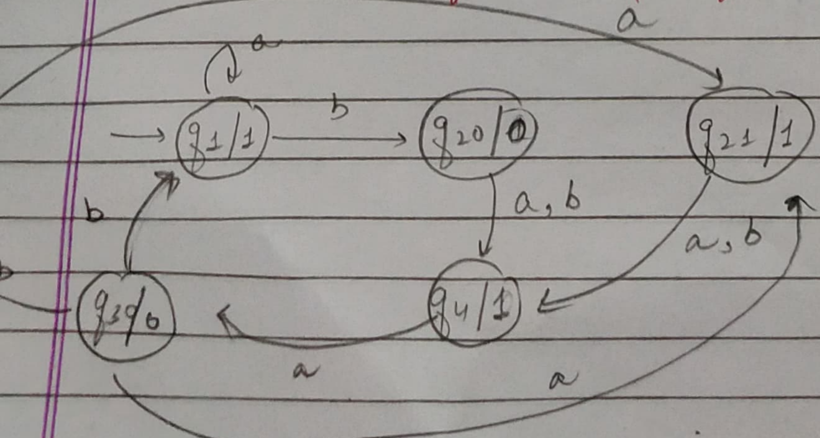
⇓ Moore

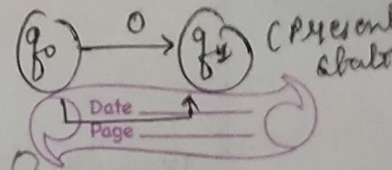
$q_2 \to 0 \; q_{20}$
$\quad \to 1 \; q_{21}$

$q_3 \to 0 \to q_{30}$
$\quad \to 1 \to q_{31}$

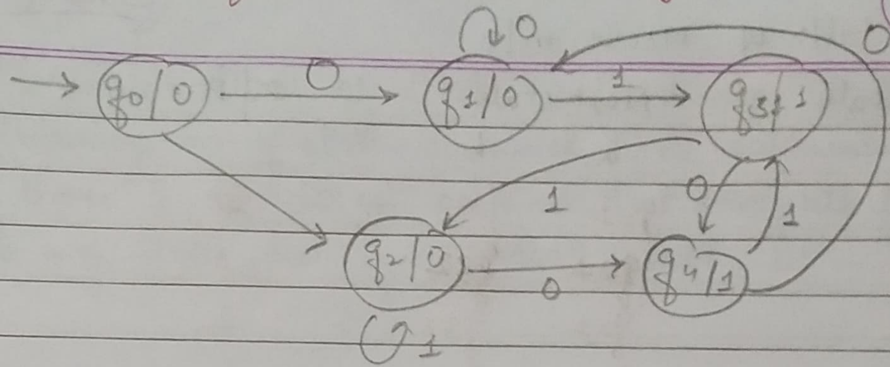
2
ection
for
re m/c

Present state	Next state a	Next state b	o/p
→ q_1	q_1	q_{20}	1
q_{20}	q_4	q_4	0
q_{21}	q_4	q_4	1
q_4	q_{30}	q_1	1
q_{30}	q_{21}	q_{31}	0
q_{31}	q_{21}	q_{31}	1





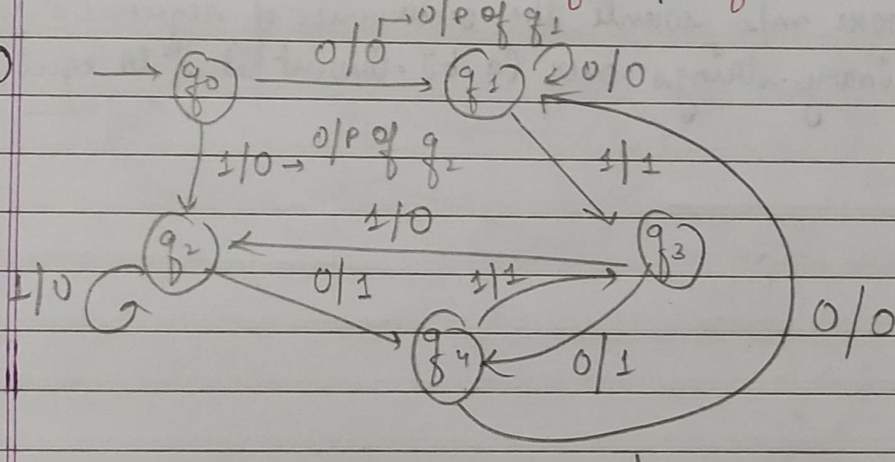
* Conversion from moore to mealy m/c



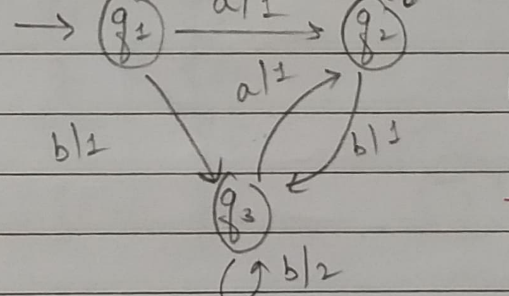
step ①

Present state δ	Next state 0	1	O/P
q0	q1	q2	0
q1	q1	q3	0
q2	q4	q2	0
q3	q4	q2	1
q4	q1	q3	1

step ②

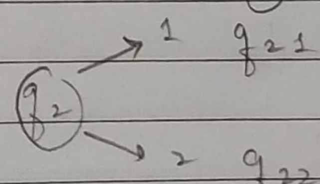


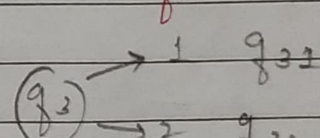
Ques



Convert into moore

Present s δ	N.S a	O/P	b	O/P
→ q1	q2	1	q3	1
q2	q2	2	q3	1
q3	q2	1	q3	2



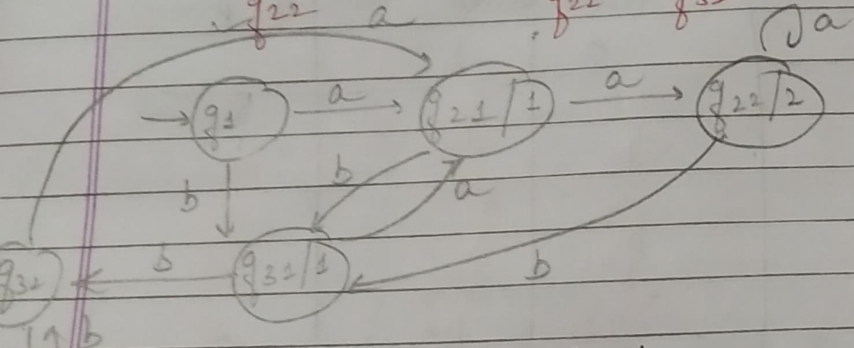


Date

Page

Transition table of Moore m/c

Present state δ	Next state a	Next state b	O/P
q_1	q_{21}	q_{31}	[illegible]
q_{31}	q_{21}	q_{32}	1
q_{32}	q_{21}	q_{32}	2
q_{21}	q_{22}	q_{31}	1
q_{22}	q_{22}	q_{31}	2

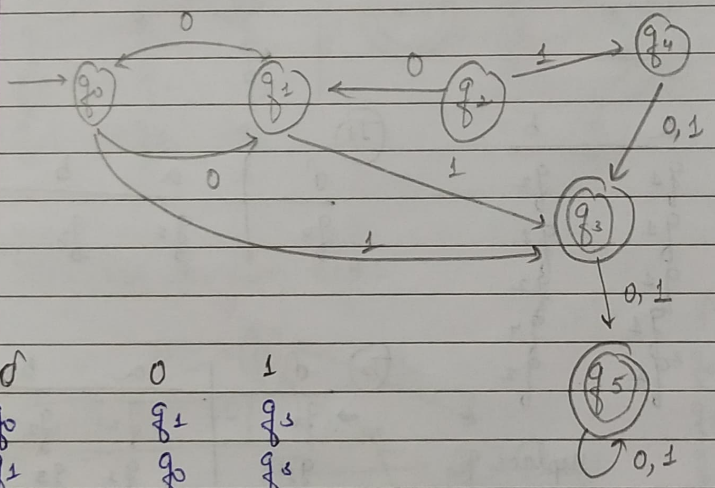


The given moore m/c counts the occurrence of sequence 'abb' in any o/p binary strings over {a, b}. Convert it into equivalent mealy m/c.

Imp **Minimization of DFA Equivalence Theorem**

Reducing no. of states in given finite automata.
If there is no direct or indirect ~~fro~~ path from initial state to any state, then ignore that state.

Now split the transaction table into two table.



δ	0	1
→ q_0	q_1	q_3
q_1	q_0	q_3
q_3	q_5	q_5
q_5	q_5	q_5

T_1

	q_3	q_3
q_3	~~q_5~~	~~q_5~~
q_5	q_5	q_5

- x Remove identical rows (last one)

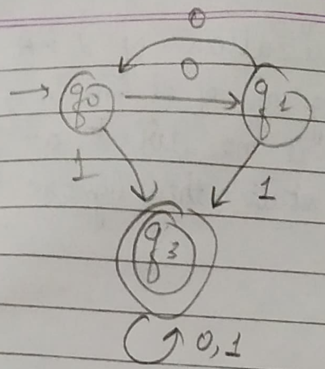
T_2

q_0	q_1	q_3
q_1	q_0	q_3

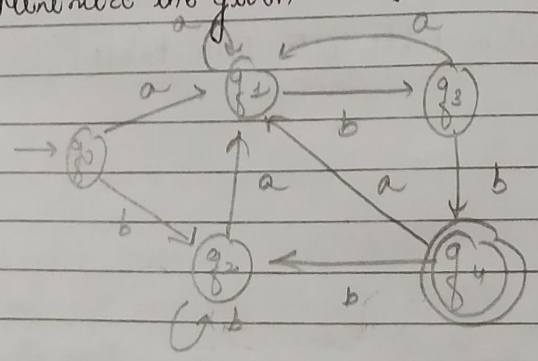
q_5 is replaced by q_3

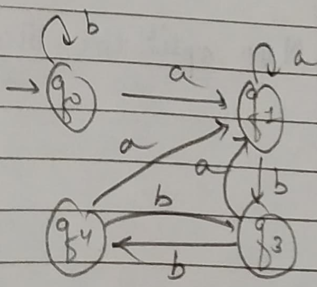
we have two identical rows remove last one

δ	0	1
q_0	q_1	q_3
q_1	q_0	q_3
q_3	q_3	q_3



② Minimize the given DFA:





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

T1

δ	a	b
q_4	q_1	~~q_2~~ q_0

Replace q_2 with q_0

T2

δ	a	b
→ q_0	q_1	~~q_2~~ q_0
q_1	q_1	q_3
q_2	q_1	q_2 ×
q_3	q_1	q_4

T2

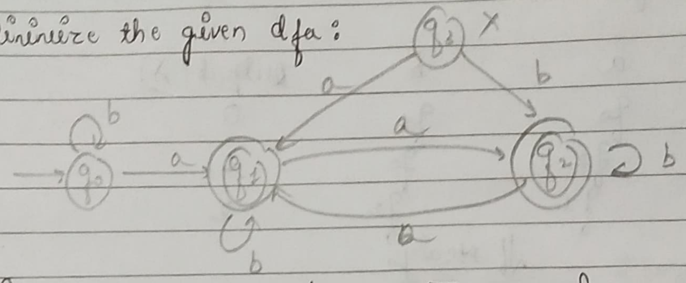
δ	a	b
→ q_0	q_1	q_0
q_1	q_1	q_3
q_3	q_1	q_4
q_4	q_1	q_0

Note: in final combined table we will not see identical rows.

Note: There is a chance that there will be no state that is unreachable from start state

→ There is a chance that there will be no minimization in states.

Ques ⑧ Minimize the given dfa:



δ	a	b
→ q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_1	q_2

T_1

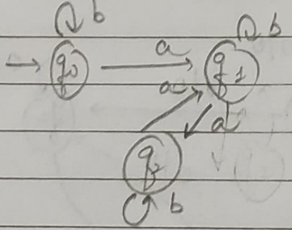
δ	a	b
q_1	q_2	q_1
q_2	q_1	q_2

T_2

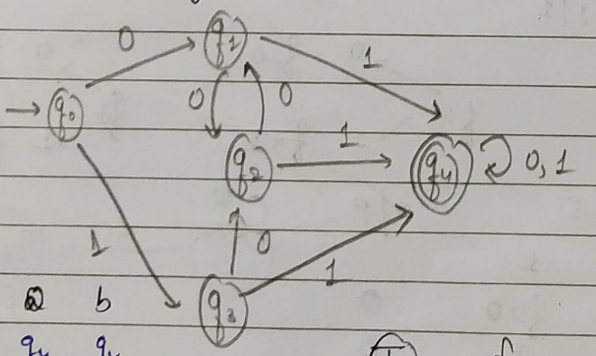
δ	a	b
q_0	q_1	q_0

Combined table:

δ	a	b
→ q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_1	q_2



Ques ⑨ Minimize the given DFA:

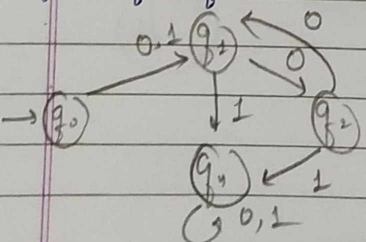


δ	0	1
→ q_0	q_1	q_3
q_1	q_2	q_4
q_2	q_1	q_4
q_3	q_2	q_4
q_4	q_4	q_4

δ	a	b
q_4	q_4	q_4

T_2

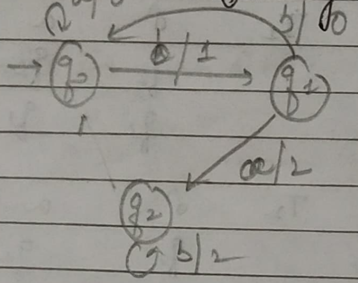
δ	0	1
→ q_0	q_1	~~q_3~~ q_1
q_1	q_2	q_4
q_2	q_1	q_4
q_3	q_2	q_4 ×



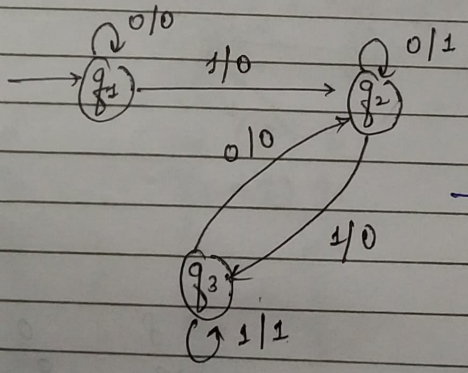
(5) Convert the given Moore m/c into its equivalent Mealy m/c

δ	a	b	Output (λ)
q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2

⇓ Mealy



(6)



P.S δ	N.S 0	O/P	1	O/P
→ q_1	q_1	0	q_2	0
q_2	q_2	1	q_3	0
q_3	q_2	0	q_3	1

$q_2 \to 0 \; q_{20}$, $q_2 \to 1 \; q_{21}$

$q_3 \to 0 \; q_{30}$, $q_3 \to 1 \; q_{31}$

P.S δ	N.S 0	1	O/P
→ 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