ALL PDF NOTES
AVAILABLE IN APP
(LINK IN DESCRPTION)

AKTU

B. Tech II-Year

Gateway Classes

BCS402 Theory of Automata and Formal Languages

Unit-1

Introduction to Basic Concepts and Automata Theory

Syallbus

Basic Concepts and Automata Theory: Introduction to Theory of Computation- Automata, Computability and Complexity, Alphabet, Symbol, String, Formal Languages, Deterministic Finite Automaton (DFA)- Definition, Representation, Acceptability of a String and Language, Non Deterministic Finite Automaton (NFA), Equivalence of DFA and NFA, NFA with ε-Transition, Equivalence of NFA's with and without ε-Transition, Finite Automata with output- Moore Machine, Mealy Machine, Equivalence of Moore and Mealy Machine, Minimization of Finite Automata.

What is Automata?

- Automata are like little machines that follow a set of rules. Imagine a vending machine: you put in money, press a button, and it gives you a snack. The vending machine can be in different states, such as idle, accepting coins, dispensing a product) and rules (if you put in enough money and press the right button, it gives you a snack).
- In computer science, automata are used to understand how computers process information. They have states (like "start" and "end") and rules (if certain conditions are met, move from one state to another). Automata help us understand how computers can solve problems, recognize patterns, and process data.
- Automata, in computer science, refer to abstract machines that can perform specific tasks or computations. They are often used to model and analyze the behavior of computational systems.

BASIC TERMINOLOGY

1.SYMBOL

- A symbol (often also called a character) is the smallest building block, which can be any alphabet, letter etc.

a,b,c,0,1,2,A,B,C...

2.Alphabets (Σ):

- Alphabets are a set of symbols, which are always *finite*.
- Collection of symbols

Σ={0,1} is an alphabet of binary digit

Σ={0,1,2,3.....9} an alphabet of decimal digit

Σ={a, b, c}

Σ={A,B,C,D ......,Z}

Σ(SIGMA)

SOME BASIC TERMINOLOGY

3.String:

- A string is a finite sequence of symbols from some alphabet. A string is generally denoted as w and the length of a string is denoted as $|w|$.
- Collection of symbol over alphabet

$\Sigma=\{a, b\}$ possible string over the alphabet

a, b, ab, ba, aa, bb ....

$W = a$

$|W| = 1$

Empty string is the string with zero occurrence of symbols, represented as ε.(epsilon)

ϵ

ε

ϵ

Number of Strings (of length 2) that can be generated over the alphabet {a, b}: - -

aa
bb
ab
ba

- Length of String $|w| = 2$
- Number of Strings = 4
- For alphabet {a, b} with length n, number of strings can be generated = 2^n

- **NOTE**

aaa, bbb, ccc

The set of strings, including the empty string, over an alphabet Σ is denoted by Σ^*.

For $\Sigma = \{0, 1\}$ we have set of strings as $\Sigma^* = \{\epsilon, 0, 1, 01, 10, 00, 11, 10101, ...\}$.

And $\Sigma 0=\{\epsilon\}$ $\Sigma 1 = \{0,1\}$, $\Sigma 2 = \{00, 01, 10, 11\}$, ...

$\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

,bba ,bbb}

- Σ^* contains an empty string ϵ. The set of non- empty string is denoted by Σ^+. From this we get:

$\Sigma^* = \Sigma^+ \cup \{\epsilon\}$ → Positive

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

$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11\}$

$\Sigma^+ = \{0, 1, 00 ...$

4. Language

- language is a *set of string all of which are chosen from some* Σ^**, where* Σ *is a particular alphabet*. This means that language L is subset of Σ^*. An example is English language, where the collection of legal English words is a set of strings over the alphabet that consists of all the letters.

- For $\Sigma = \{0, 1\}$ we have set of strings as $\Sigma^* = \{\epsilon, 0, 1, 01, 10, 00, 11, 10101, ...\}$.

L={ set of all string end with 1 and start with 1} $L = \{1\}$

L={1,101,1111,100001}

If Σ is an alphabet, and $L \subseteq \Sigma^*$

- A language that can be formed over ' Σ ' can be Finite or Infinite.

SOME BASIC TERMINOLOGY

5.Kleene star Σ^* (universal set) INFINITE SET/ Kleene closure/ Sigma star

empty string

$\epsilon / \Lambda / \lambda$

Representation: $\Sigma^* = \Sigma 0 \cup \Sigma 1 \cup \Sigma 2 \cup \Sigma 3 \ldots\ldots\ldots$ $\Sigma = \{0, 1\}$(let)

Set of all possible string of length p

$\Sigma^0 = \{\varepsilon\}$ (LENGTH OF STRING MUST BE ZERO/NULL STRING/EMPY STRING}

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

$\Sigma^2 = \{00,01,10,11\}$

Example – If $\Sigma = \{a, b\}$, $\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, \ldots\ldots\ldots\}$

Definition – The Kleene star, Σ^*, is a unary operator on a set of symbols or strings, Σ, that gives the infinite set of all possible strings of all possible lengths over Σ including λ OR ε

BASIC TERMINOLOGY

6.Kleene plus/kleene positive/positive clouser Σ^+

- **Representation**

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

Kleene closure & Kleene positive

$\Sigma^+ = \Sigma^* - \Sigma 0$

$\Sigma^+ = \Sigma 1 \cup \Sigma 2 \cup \Sigma 3......$

- **Example** – If $\Sigma = \{ a, b \}$, $\Sigma+ = \{ a, b, aa, ab, ba, bb,...........\}$
- **Definition** – The set $\Sigma+$ is the infinite set of all possible strings of all possible lengths over Σ excluding λ.

FINITE AUTOMATA

- It is a mathematical model Used to describe computation, this model have discrete inputs, outputs, states and a set of transitions from state to state that occurs on input symbols from the alphabet Σ.

FORMAL DEFINITION OF FINITE AUTOMATA

Finite automata is defined as a 5-tuples

M=(Q, Σ, δ, q0, F)

Symbol Meaning: Σ (sigma), δ (delta)

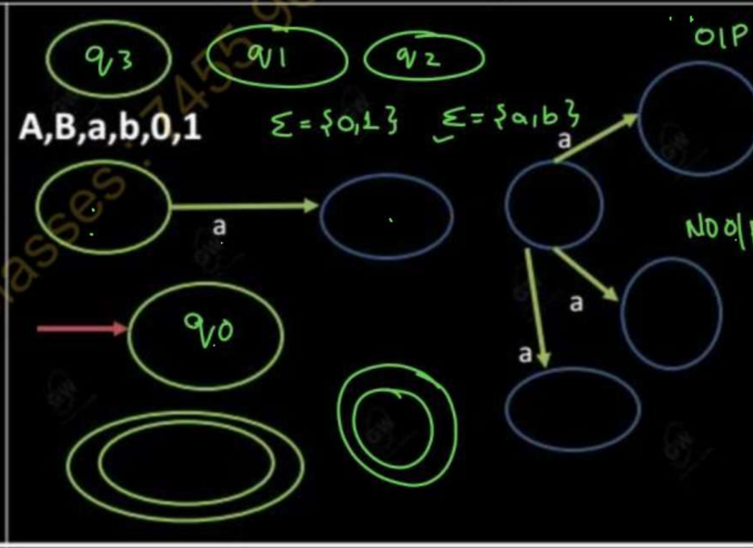
Q: Finite set called states.$\{q_0, q_1..q_n\}$---------->

Σ: Finite set called alphabets.------------------>

δ: is the transition function.---------------------->

q0 ∈ Q is the start or initial state.----------->

F: Final or accept state.$\{q_f\}$--------------->



DIFFERENT TYPES OF FINITE AUTOMATA

Finite Automata without output	Finite Automata with Output
➢ Deterministic Finite Automata (DFA). ➢ Non-Deterministic Finite Automata (NFA or NDFA). ➢ Non-Deterministic Finite Automata with epsilon moves (e-NFA or e-NDFA).	➢ Moore machine. ➢ Mealy machine

Finite Automata Representation

Graphical (Transition diagram)

Tabular (Transition table)

Mathematical (Transition function)

DETERMINISITIC FINITE AUTOMATA

- The behavior of a DFA is entirely determined by its current state and the input symbol being processed.
- For any given state and input symbol, there is only one possible next state

- It has a finite number of states.

- An automaton is a mathematical model of a system that processes inputs according to a set of rules.
- In the case of DFAs, the system transitions between states based on the inputs it receives

FORMAL DEFINITION OF DETERMINISTIC FINITE AUTOMATA

A DFA can be represented by a 5-tuple (Q, ∑, δ, q0, F)

- Q is a finite set of states. $\{q_0, q_1, q_3, q_4\}$
- ∑ is a finite set of symbols called the alphabet.
- δ is the transition function where $\delta: Q \times \Sigma \rightarrow Q$
- q_0 is the initial state from where any input is processed ($q_0 \in Q$).
- F is a set of final state/states of Q ($F \subseteq Q$)

NOTE:

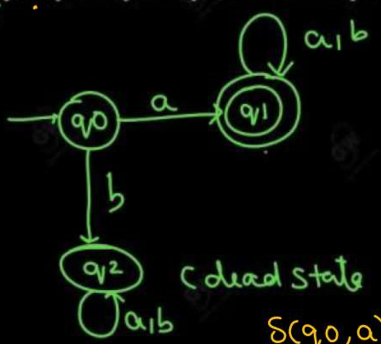
- *There can be many possible DFAs for a pattern.*

Minimum

Design a DFA/MDFA/FA $\Sigma=\{a , b\}$ such that string accepted must start with substring a

TRANSITION DAIGRAM

$L=\{a, ab, aab, aba, abbbabab, ...\}$



q1 - final state

$*q_1$ | q_1

Wrong String

× ba

× baba

bbabb

baaa

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_2$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_1$

$\delta(q_2, a) = q_2$

$\delta(q_2, b) = q_2$

TRANSITION TABLE

Input

	a	b
$\rightarrow q_0$	q_1	q_2
$*q_1$ | q_1	q_1	q_1
q_2	q_2	q_2

$\{Q, \Sigma, \delta, q_0, F\}$

$\{q_0, q_1, q_2\}, \{a , b\}, \delta, q_0, \{q_1\}$

δ transition function is defined by the transition table

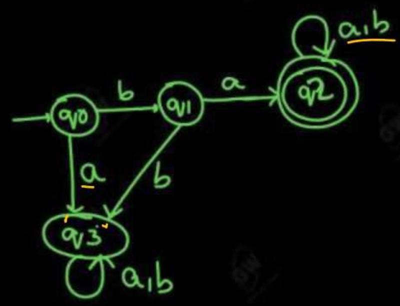
Note:

- Dead state is that state where there is no way to go back to final state

Design a DFA/MDFA ∑={a , b} such that string accepted must start with substring ba

TRANSITION DAIGRAM

L={ba, baaaaa, babb,babababab .... }



Wrong string
bba
bbb
abab

TRANSITION TABLE

	a	b
→q_0	q_3	q_1
q_1	q_2	q_3
*q_2	q_2	q_2
q_3	q_3	q_3

$\{Q, \sum, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_o, q_2\}$

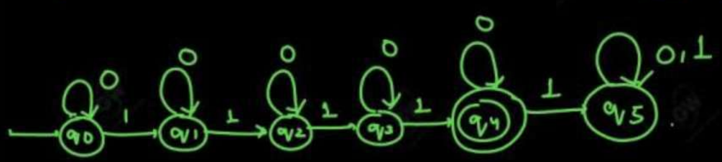
Note

Minimum state= minimum string length+2

Design the finite automata which accept set of string containing exactly four 1's in every string
∑=[0,1}

TRANSITION DAIGRAM

L={1111,001111,01010101,110011,11110000,111010}



$\delta(q0,0) = q0$
$\delta(q0,1) = q1$
$\delta(q1,0) = q1$
$\delta(q1,1) = q2$

Wrong string
0011101011
1111010
101010111110

TRANSITION TABLE

	0	1
→q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_3
q_3	q_3	q_4
*q_4	q_4	q_5
q_5	q_5	q_5

$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0, 1\}, \delta, q_o, q_4\}$

APPLICATION OF FINITE AUTOMATA

- **DNA Sequence Analysis:** Finite automata can be used to search for patterns in DNA sequences.
- **Robotics:** Finite automata can be used to model and control robot behavior in certain scenarios.
- **Pattern Recognition:** Finite automata can be used in image and signal processing for pattern recognition tasks.

LIMITATION OF FINITE AUTOMATA

- **Limited Memory:** Finite automata have a finite number of states,
- **Limited Expressiveness:** Finite automata can only recognize regular languages,
- **Difficulty with Complex Patterns:** While finite automata are good at recognizing simple pattern

What do you mean by Kleene closure of set A ?

- The Kleene closure of a set A, denoted A^*
- It includes the empty string ε all single elements of A, all possible pairs of elements from A, and so on.

A is the set $\{0,1\}$ then $A^* =$
$\{\varepsilon,0,1,00,01,10,11,000,\ldots\}$,

AKTU QUESTIONS

Q1	Define alphabet and string in automata theory?	AKTU 2021-22
Q2	Give the definition of DFA.	AKTU 2021-22 AKTIU 2011-12
Q3	Write down the application of finite automata.	AKTU 2018-19
Q4	Define alphabet string and language.	AKTU 2017-18
Q5	Write down the application and limitation of finite automata.	AKTU 2017-18
Q6	Define and give the difference between positive closure and Kleene closure.	AKTU 2016-17
Q7	What do you mean by the Kleene closure of set A?	AKTU 2008-09
Q8	Design the finite automata which accept set of string containing exactly four 1's in every string $\Sigma=\{0,1\}$.	AKTU 2014-15
Q9	Differentiate between L^* and L^+	AKTU 2013-14

DPP

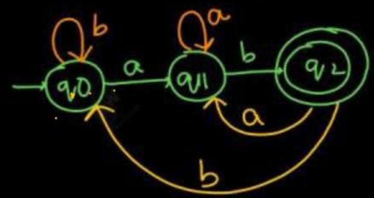
Q1 Draw the DFA over $\Sigma = \{0,1\}$ Start Withe 10.

Q2 Draw the DFA | MDFA | FA over $\Sigma = \{0,1\}$ Start with 00

Design a DFA/MDFA/FA $\Sigma=\{a, b\}$ such that every string accepted must end with a string w=ab

TRANSITION DAIGRAM

L={ab, bab, aaabbbab ,aaaaab, bbbabaaab ……}



DFA

Wrong string

× ababaaa

× bababb

× Abababbbb

W = ab

|W| = 2

Minimum number of states = 2+1=3

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$*q_2$ / q_2	q_1	q_0

$\{Q, \Sigma, \delta, q_0, F\}$

$\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, q_2\}$

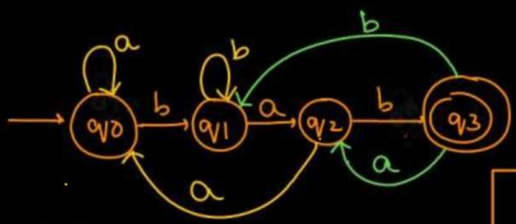
Note no dead state

Number of state = length of minimum string+1

Design a DFA/MDFA $\Sigma=\{a, b\}$ such that string accepted must end with substring bab

TRANSITION DAIGRAM

L={bab, aaaabab, aaababbbab, bbabaabab ....}



Min length string = bab(W)

|W| = 3

Minimum States = 3+1 = 4

Wrong string

→ aaabaaa

→ aaababa

→ aaabbbaba

→ bbbbaba

TRANSITION TABLE

	a	b
→q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_3
*q_3	q_2	q_1

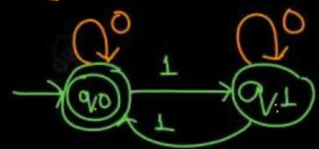
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_o, q_3\}$

Design a DFA/MDFA $\Sigma=\{0,1\}$ such that accept the string which contain even number of 1s

TRANSITION DAIGRAM

L={0,00000,11,0101,110011,0101011, 1100110011 -------}



NOTE 010100

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

Wrongstring

- 111000
- 0001010 1
- 1111111000

TRANSITION TABLE

	0	1
$\rightarrow q_0$ / $*q_0$	q_0	q_1
q_1	q_1	q_0

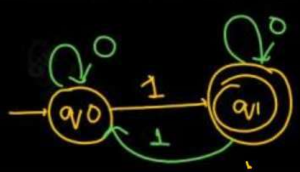
$\{Q, \Sigma, \delta, q_0, F\}$

$\{\{q_0, q_1\}, \{0, 1\}, \delta, q_0, q_0\}$

Design a DFA/MDFA $\Sigma=\{0,1\}$ such that accept the string which contain 0dd number of 1s

TRANSITION DAIGRAM

L={ 1, 10, 01, 1110, 10000,1110000,00001111
000110100, 001110, ...}



NOTE→ 1110101

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

final state

this means that string is accepted

TRANSITION TABLE

	0	1
$\rightarrow q_0$	q_0	q_1
q_1 | $*q_1$	q_1	q_0

$\{Q, \Sigma, \delta, q_0, F\}$

$\{\{q_0, q_1\},\{0, 1\}, \delta, q_0, q_1\}$

Non accepting string

111100

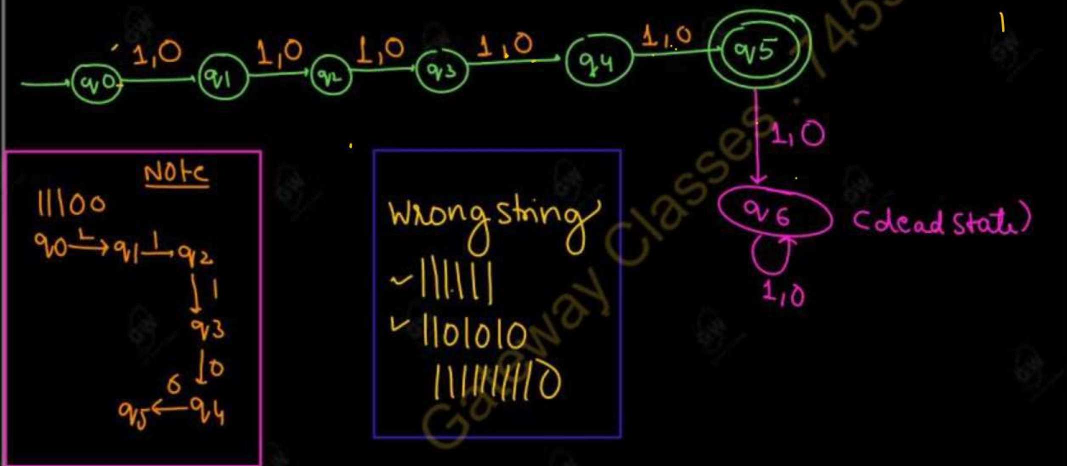
0001111

1010101

Construct a DFA that accept all the string of 1s' and 0 s' where length of string is exactly 5

TRANSITION DAIGRAM

L={00000,11111, 10101,01011,00011,11000,10101 ...}



TRANSITION TABLE

	0	1
→q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_4	q_4
q_4	q_5	q_5
*q_5	q_6	q_6
q_6	q_6	q_6

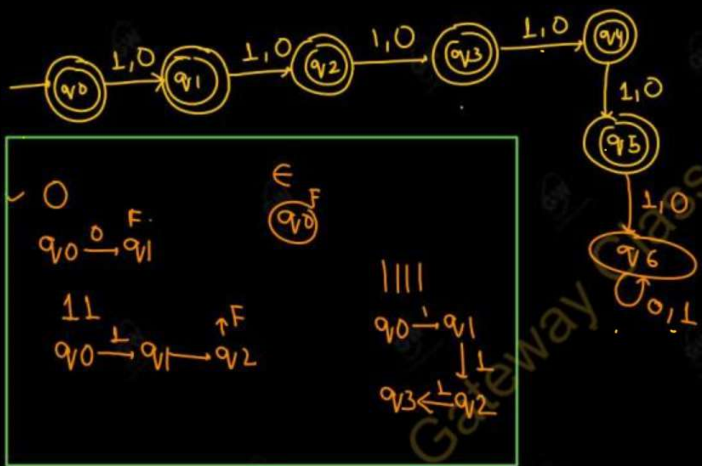
$\{Q, \Sigma, \delta, q_0, F\}$

$\{\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, \delta, q_0, q_5\}$

Construct a DFA that accept all the string of 1s' and 0 s' where length of string is at most 5

TRANSITION DAIGRAM

L={∈ ,1,0,11,01,10 ,111,0111,01010,10101...... }



wrong string

1111111

0000000

1010101

TRANSITION TABLE

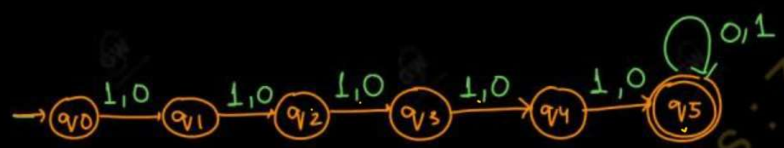
	0	1
→* q_0	q_1	q_1
* q_1	q_2	q_2
* q_2	q_3	q_3
* q_3	q_4	q_4
* q_4	q_5	q_5
* q_5	q_6	q_6
q_6	q_6	q_6

$\{Q, \Sigma, \delta, q_o, F\} = \{\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, \delta, q_o, \{q_0, q_1, q_2, q_3, q_4, q_5\}\}$

Construct a DFA that accept all the string of 1s' and 0 s' where length of string is at least 5

TRANSITION DAIGRAM

L={00000, 11111, 11111111, 1010100011,010111,0001000,1100000, 101010 ...}



NOTE

101011

$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{0} q_4 \xrightarrow{1} q_5 \xrightarrow{1} q_5$

Non-acceptable string

1111

010

111

1110

TRANSITION TABLE

	0	1
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_4	q_4
q_4	q_5	q_5
$*q_5$	q_5	q_5

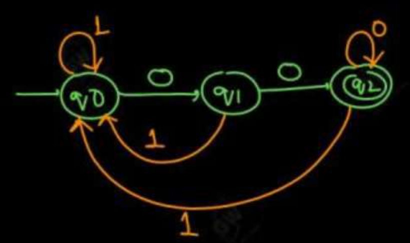
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0, 1\}, \delta, q_o, q_5\}$

Design a DFA/MDFA/FA $\Sigma=\{0,1\}$ such that every string accepted must end with a string w=00

TRANSITION DAIGRAM

L={00, 100,1100, 110100,0000 }



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

q_0 → Accept (001100)

Non accepting string

1111

1110

0010

TRANSITION TABLE

	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
$*q_2$	q_2	q_0

$\{Q, \Sigma, \delta, q_o, F\}$

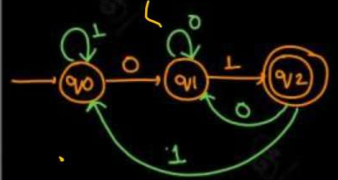
$\{q_0, q_1, q_2\}, \{0 , 1\}, \delta, q_0, q_2\}$

Design a DFA/MDFA/FA $\Sigma=\{0, 1\}$ such that every string accepted does not end with a string w=01

TRANSITION DAIGRAM

L={ ε,0,1,1111,000000,0101111,......}

① end with 01



→ convert final state to non-final

→ convert all non-final state to final

DFA that will accept (01)

wrongstring
00001
11101
11100001
01

ε (Right String)
0
1
00
111
11
10
011
010

TRANSITION TABLE

	0	1
→*q_0	q_1	q_0
*q_1	q_1	q_2
q_2 (Non-final State)	q_1	q_0

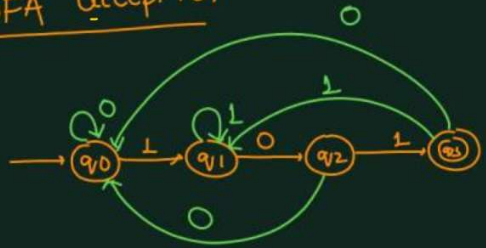
$\{Q, \Sigma, \delta, q_0, F\}$

$\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_0, q_1\}\}$

DFA Accept every string except 101 $\Sigma = \{0,1\}$

end

DFA accept 101



does not end

$Q, \Sigma, \delta, q_0, F$

$\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta,$

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

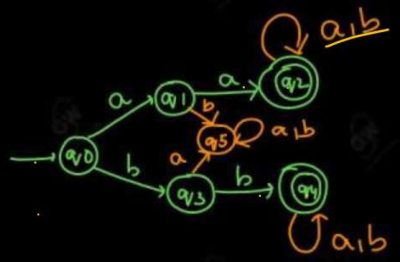
transition table

	0	1
$\rightarrow * q_0$	q_0	q_1
$* q_1$	q_2	q_1
$* q_2$	q_0	q_3
q_3	q_0	q_1

Design a DFA/MDFA/FA $\Sigma=\{a, b\}$ such that every string accepted that start with either aa or bb

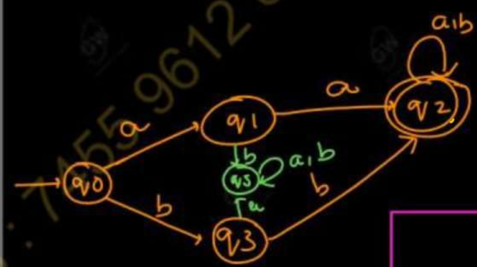
TRANSITION DAIGRAM

L={aa, bb, aabab, bbbaba, aabbbaba……}



Minimize →

q5 (dead state)



Wrong string
- abab
- baba
- b

Right String
- aabb
- aababa
- bbaa

Design a DFA/MDFA/FA $\Sigma=\{a, b\}$ such that every string accepted that start with either aa or bb

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_5
$*q_2$	q_2	q_2
q_3	q_5	q_2
q_5	q_5	q_5

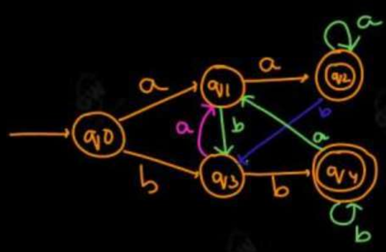
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3, q_5\}, \{a, b\}, \delta, q_o, q_2\}$

Design a DFA/MDFA/FA $\Sigma=\{a, b\}$ such that every string accepted that end with either aa or bb

TRANSITION DAIGRAM

L={aa, bb, baaababbabaa, ababbaaa, babababb, ababbabb, abababbbbaa.....}



baaaba bbabaa

$q_0 \xrightarrow{b} q_3 \xrightarrow{a} q_1 \xrightarrow{a} q_2$

$q_3 \xleftarrow{b} q_2 \xleftarrow{a} q_2$

$q_3 \xrightarrow{a} q_1 \xrightarrow{b} q_3 \xrightarrow{b} q_4 \xrightarrow{a} q_1 \xrightarrow{b} q_3$

$q_2 \xleftarrow{a} q_1 \xleftarrow{a} q_3$

Wrong string

→ abab

→ baba

Design a DFA/MDFA/FA $\Sigma=\{a , b\}$ such that every string accepted that end with either aa or bb

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_3
$*q_2$	q_2	q_3
q_3	q_1	q_4
$*q_4$	q_1	$q_{\cancel{3}4}$

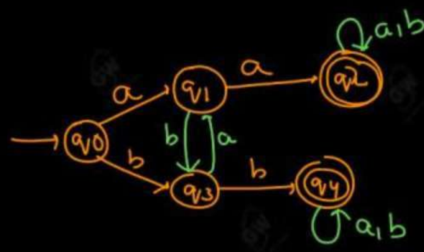
$\{Q, \Sigma, \delta, q_o, F\}$

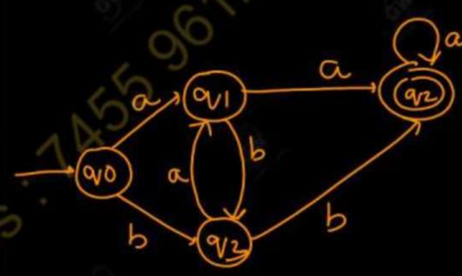
$\{\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_o, \{q_2, q_4\}\}$

Design a DFA/MDFA/FA $\Sigma=\{a, b\}$ such that every string must contain substring either aa or bb

TRANSITION DAIGRAM

L={aa, bb, aababab, bbbabab, bababaabbab, bbabaaabb, abababbbabab}





Design a DFA/MDFA/FA $\Sigma=\{a, b\}$ such that every string must contain substring either aa or bb

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_3
$*q_2$	q_2	q_2
q_3	q_1	q_2

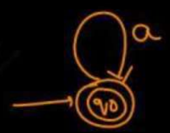
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_o, q_2\}$

Design a DFA/MDFA $\Sigma=\{a\}$ such that accept the string a^n |n>=0

TRANSITION DAIGRAM

L= {ε, a, aa, aaaa, aaaa , aaaaa.................}



TRANSITION TABLE

	a
→ $*q_0$	q_0

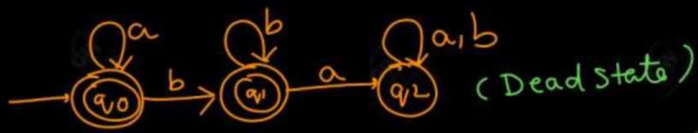
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0\},\{a\}, \delta, q_o, q_0\}$

Design a DFA/MDFA $\Sigma=\{0,1\}$ such that accept the string $a^n b^m \mid n, m>=0$

TRANSITION DAIGRAM

$L=\{\varepsilon, a, aaaa, b, bbbb, aabbb \ldots\ldots\}$



aba	∈ (Acceptable String)
abaaaa	a
babbaa	aaaaa
(Non-acceptable String)	b
	bbbbbb
	aab
	aaabb

TRANSITION TABLE

	a	b
$\rightarrow {}^*q_0$	q_0	q_1
${}^*q_1$	q_2	q_1
q_2	q_2	q_2

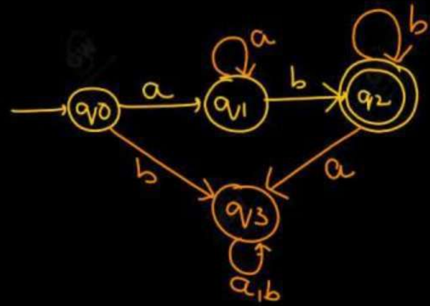
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_0, q_1\}\}$

Design a DFA/MDFA $\Sigma=\{0,1\}$ such that accept the string $a^n b^m \mid n, m>=1$ // $L=\{w \in \{a, b\}^* \mid w= a^n b^m$ for $m,n>0$

TRANSITION DAIGRAM

L={ ab, aaabbbbb,abbbb,aaaaab}



Wrong string
- aaba
- baba
- bba
- a
- b

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_1	q_2
$^*q_2$	q_3	q_2
q_3	q_3	q_3

$\{Q, \Sigma, \delta, q_o, F\}$

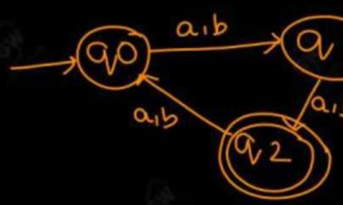
$\{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_o, \{q_2\}\}$

Construct a DFA over{ a, b} such that every string accepted must |w|=2(mod3) / {w:| w | mod 3>1}

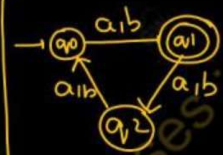
TRANSITION DAIGRAM

L={ aa,bb,aaabb,aabbb,aabbaabb,bbbbaaaa…….}

|W| = 2 (mod3)



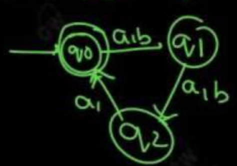
|W|=1(mod 3)



	a	b
→q0	q1	q1
*q1	q2	q2
q2	q0	q0

(Multiple of 3), divisible by 3

|W| = 0(mod 3)



	a	b
*→q0	q1	q2
q1	q2	q2
q2	q0	q0

TRANSITION TABLE |W| = 2(mod 3)

	a	b
→q_0	q_1	q_1
q_1	q_2	q_2
*q_2	q_0	q_0

{Q, ∑, δ, q_o, F}

{{q_0, q_1, q_2},{a,b } , δ, q_o ,q_2}

NOTE

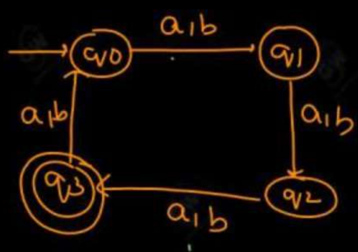
W=aa |W|=2 (length)

Construct a DFA over{ a, b} such that every string accepted must |w|=3(mod 4)

TRANSITION DAIGRAM

L={ aaa, bbb, abababa, bababab ,abababababbbb.....}

|W|: 3 3 7 7 11



(wrong string)

W = ∈ |W| = 0
W = aa |W| = 2
W = abab |W| = 4
W = abbbb |W| = 5
W = abab |W| = 4
W = baba |W| = 4

NOTE

	Intial State	Finalstate
→ 0(mod 4) / Multiple of 4	q0	q0
→ 1(mod 4)	q0	q1
→ 2(mod 4)	q0	q2

TRANSITION TABLE

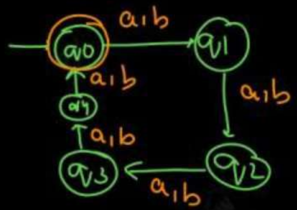
	a	b
→ q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
* q_3	q_0	q_0

$\{Q, \sum, \delta, q_o, F\} = \{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_o, \{q_3\}\}$

Construct a DFA over{ a, b} such that every string accepted must |w|=0(mod 5)/ {w: |w|mod5 =0}

TRANSITION DAIGRAM

L={ε,aaaaa,bbbbb, ababababab, bababababa..............................}



TRANSITION TABLE

	a	b
→*q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_4	q_4
q_4	q_0	q_0

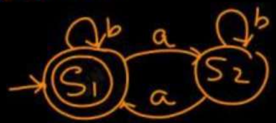
$\{Q, \sum, \delta, q_o, F\} = \{\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_o, \{q_o\}\}$

Construct a DFA that accept all the string in which a is even(a is divisible by 2) and b is divisible by 3

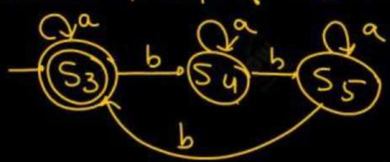
a is multiple of 2

a is divisible 2 | $|w|_a = 0 (\text{mod } 2)$

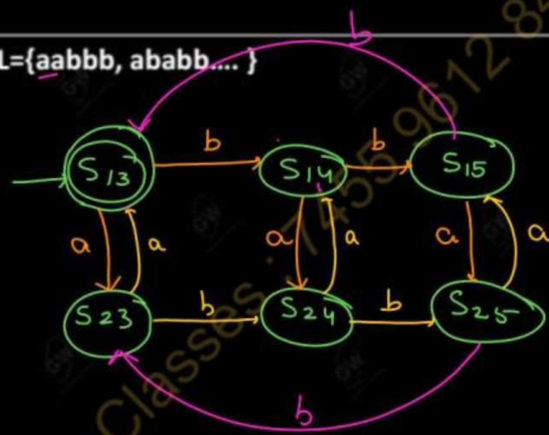
a is even



b is divisible | Multiple of 3 $|w|_b = 0 \text{ mod } 3$



L={aabbb, ababb.... }



$|w|_a \text{ mod } 2 = 0$ and

$|w|_b \text{ mod } 3 = 0$

Wrong string

aaabb

bbaaaaa

Construct a DFA that accept all the string over={a,b} in which a is even(a is divisible by 2) and b is divisible by 3

TRANSITION TABLE

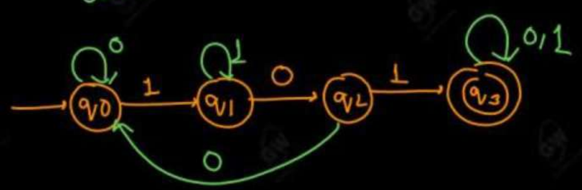
	a	b
→*S_{13}	S_{23}	S_{14}
S_{14}	S_{24}	S_{15}
S_{15}	S_{25}	S_{13}
S_{23}	S_{13}	S_{24}
S_{24}	S_{14}	S_{25}
S_{25}	S_{15}	S_{23}

$\{Q, \Sigma, \delta, q_o, F\} = \{\{S_{13}, S_{14}, S_{15}, S_{23}, S_{24}, S_{25}\}, \{a, b\}, \delta, S_{13}, S_{13}\}$

Construct a DFA that accept all the string over {1,0} except 101 string

DFA accept 101 as substring)

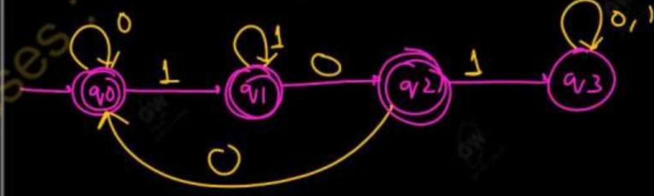
L = {101, 10111, 1010010l, 111101000 . . .
000101



complement of L (L̄)

L̄ = {ε, 0, 1, 11, 1000, 001, 1111, 1001 - - }

(this will not accept 101 as substring}



Wrong string

~~101~~ 000101

11011

Construct a DFA that accept all the string over {1,0} except 101 string

TRANSITION TABLE

	0	1
$\rightarrow *q_0$	q_0	q_1
$*q_1$	q_2	q_1
$*q_2$	q_0	q_3
q_3	q_3	q_3

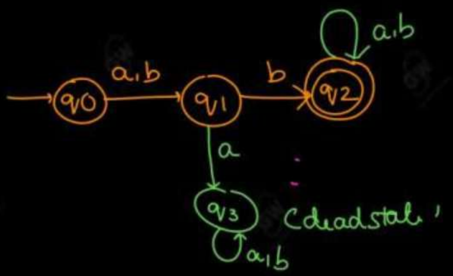
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_o, \{q_0, q_1, q_2\}\}$

Design a DFA that accept all string over the $\Sigma=\{a , b\}$ such that every accepted string 2^{nd} symbol from left end is always b

TRANSITION DAIGRAM

L={ab, bb, abbbabab, bbababab.....}



Wrong string
aaba
baba

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_3	q_2
$^*q_2$	q_2	q_2
q_3	q_3	q_3

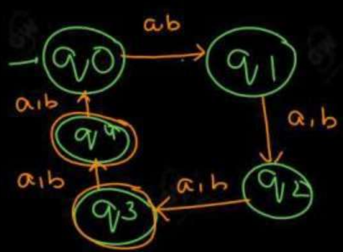
$\{Q, \Sigma, \delta, q_0, F\}$

$\{\{q_0, q_1, q_2, q_3\},\{a, b\}, \delta, q_0, \{q_2\}\}$

Design a DFA that accept all string over the $\Sigma=\{a, b\}$ $|w| \bmod 5>2$// $|w| \bmod 5=3$ and $|w| \bmod 5=4$

TRANSITION DAIGRAM

L={aaa,baba, bababab a....}



W: ababbbb
|W| = 7

W = ababa
|W| = 5

W: aba
|W| = 3

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
$*q_3$	q_4	q_4
$*q_4$	q_0	q_0

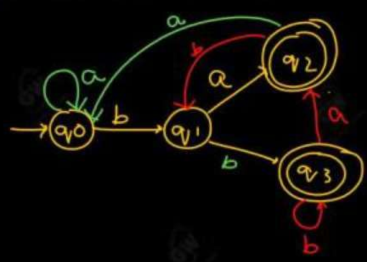
$\{Q, \Sigma, \delta, q_0, F\}$

$\{\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_0, \{q_3, q_4\}\}$

Design a DFA that accept all string over the $\Sigma=\{a,b\}$ such that every accepted string 2^{nd} symbol from right end is always b

TRANSITION DAIGRAM

L={ba,bb,aba,aaaaba....}



$2^n \quad 2^2 = 4$

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_0	q_1
q_1	q_2	q_3
$*q_2$	q_0	q_1
$*q_3$	q_2	q_3

Number of states 2^n n (postion of symbol from right}

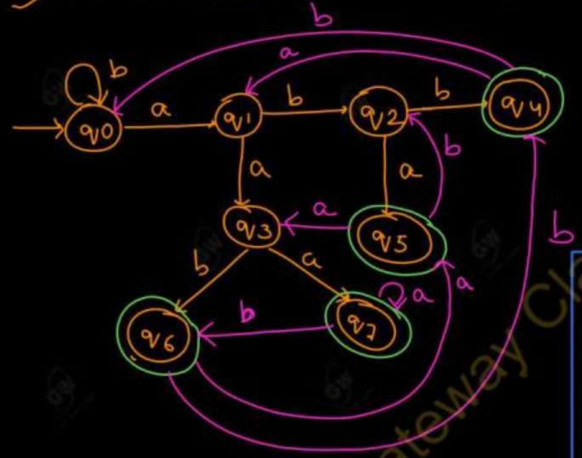
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_o, \{q_2, q_3\}\}$

Design a DFA that accept all string over the $\Sigma=\{a,b\}$ such that every accepted string 3 rd from right is always a

TRANSITION DAIGRAM

L={aaa, baaa,bbbaba ....}



Wrong string

ababbb

$Q, \Sigma, \delta, q_0, F$

$\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}, \{a,b\}, \delta, q_0$

$\{q_4, q_5, q_6, q_7\}$

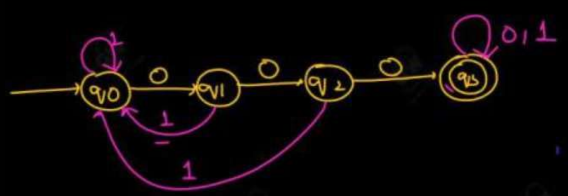
TRANSITION TABLE

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

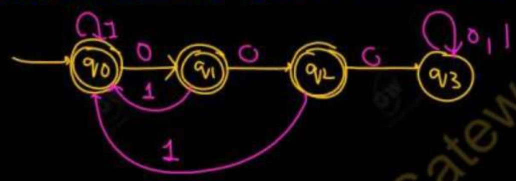
Design a DFA that accept all string over the $\Sigma=\{0,1\}$ such that every accepted all string which has three consecutive ~~three~~ zeros

TRANSITION DAIGRAM

L={000,11010000,000111,101000111....}



Does not contain consecutive three zero



$\bar{L}$ = {0, 00, 11, 10, 01

11101001

Complement of L

TRANSITION TABLE

	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
$^*q_3$	q_3	q_3

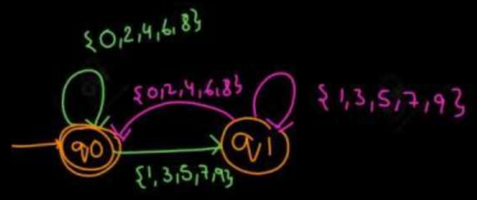
$\{Q, \Sigma, \delta, q_o, F\}$

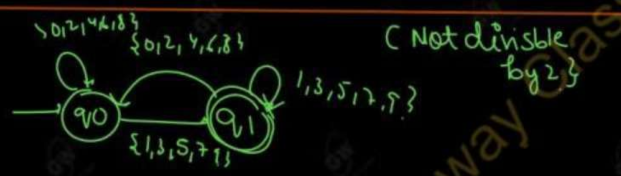
$\{\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_o, \{q_3\}\}$

Construct a DFA for the decimal number{0,1,2.....,9} divisible by 2/ multiple 0f 2/|w| mod 2=0

TRANSITION DAIGRAM

L={2,4 8, 10, 12, 14,20,22,24,.....}





TRANSITION TABLE

	{0,2,4,6,8}	{1,3,5,7,9}
$\rightarrow {}^{*}q_0$	q_0	q_1
q_1	q_0	q_1

$\{Q, \Sigma, \delta, q_o, F\}$

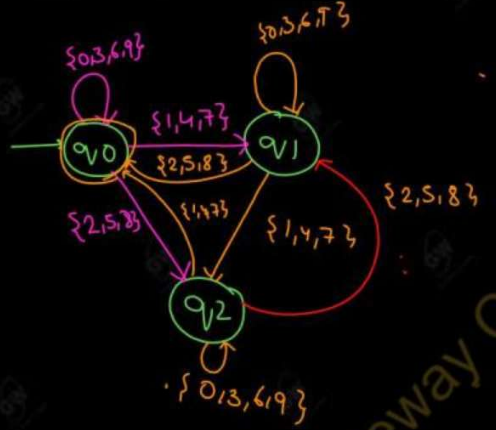
$\{\{q_0, q_1\},$ δ, q_o $, \{q_0\}\}$

{0,1,2,3,4,5,6,7,8,9}

Construct a DFA for the decimal number{0,1,2.....,9} divisible by 3

TRANSITION DAIGRAM

L={0,3,9......}



TRANSITION TABLE

	{0,3,6,9}	{1,4,7}	{2,5,8}
$\rightarrow {}^{*}q_0$	q_0	q_1	q_2
q_1	q_1	q_2	q_0
q_2	q_2	q_0	q_1

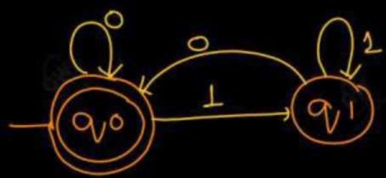
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2\}, \{0,1,2....9\}, \delta, q_o, \{q_0\}\}$

Construct a DFA for the which accept the ste of all string over{0,1} which interpreted as binary number divisible by 2

TRANSITION DAIGRAM

L={0,00,10,000,010,100,......}



TRANSITION TABLE

	0	1
→* q_0	q_0	q_1
q_1	q_0	q_1

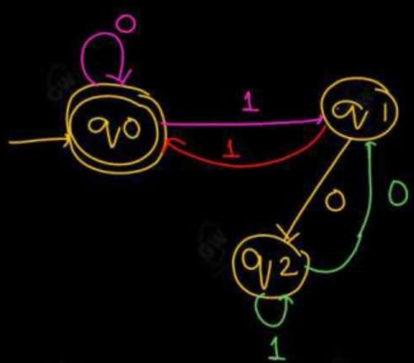
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, \}, \{0,1\}, \delta, q_o, \{q_0\}\}$

Construct a DFA for the which accept the all string over{0,1} which interpreted as binary number divisible by 3

TRANSITION DAIGRAM

L={0,11,110, ...}



TRANSITION TABLE

	0	1
$\rightarrow {}^*q_0$	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

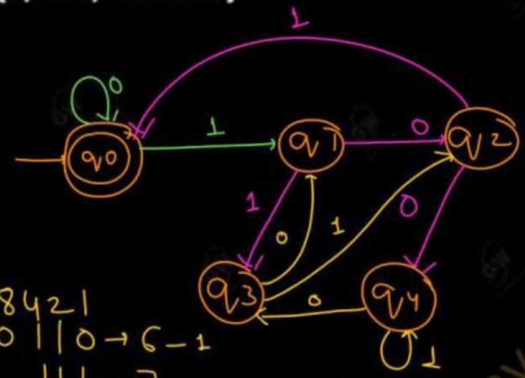
$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_o, \{q_0\}\}$

Construct a DFA for the which accept the string of all string over{0,1} which interpreted as binary number divisible by 5

TRANSITION DAIGRAM

L={0, 101, 1010.... }



8421
0110 → 6 − 1
111 − 7 2
1010 − 10 → 0
1111

TRANSITION TABLE

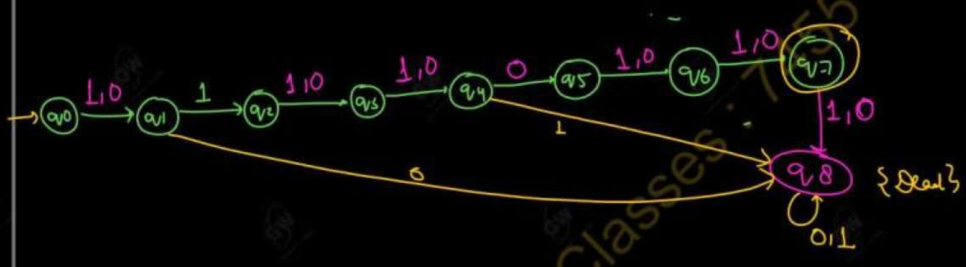
	0	1
→* q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_o, \{q_0\}\}$

Construct a DFA for the which accept the string of a~~ll string~~ over{0,1} having length 7 provided 2 digit 1 from left and 3rd digit from right is 0

TRANSITION DAIGRAM



$\{Q, \Sigma, \delta, q_0, F\}$

$\{\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}, \{0,1\}, \delta, q_0, \{q_7\}\}$

TRANSITION TABLE

	0	1
$\rightarrow q_0$	q_1	q_1
q_1	q_8	q_2
q_2	q_3	q_3
q_3	q_4	q_4
q_4	q_5	q_8
q_5	q_6	q_6
$^*q_7$	q_8	q_8
q_8	q_8	q_8

MINIMIZATION OF DFA

➢The process of elimination of state whose presence or absence does not affect the language accepting capability of DFA is called minimization of DFA and the result is minimal deterministic finite automata commonly known as MFA

> NOTE : MFA is unique for a language

DEAD STATE

It is basically created to make the system complete, can be defined as a state from which there is no transition possible to the final state

> NOTE : IN DFA , there can be more than one dead state but logically always one dead state is sufficient to complete the functionality

MINIMIZATION OF DFA

➢ **UNREACHABLE STATE**

Unreachable states are the states that are not reachable from the initial state of the DFA, for any input string.

➢ **EQIVALENT STATE**

In the minimization of a DFA, equivalent states are states that can be merged without changing the language recognized by the DFA.

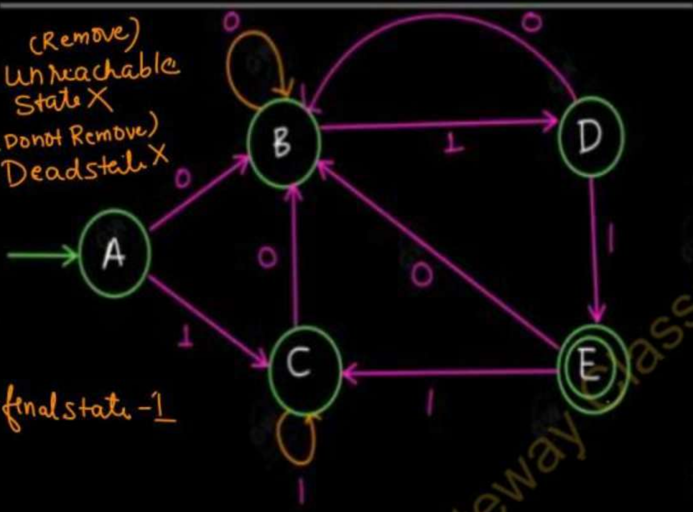
➢ **EQUIVALENCE CLASS**

An equivalent class (or equivalence class) refers to a grouping of states that are equivalent to each other. This means that all states within an equivalence class behave identically with respect to the language recognized by the DFA.

1. MINIMIZATION OF DFA

(Remove)
→ Unreachable State X
(Donot Remove)
→ Deadstate X

→ final state - 1



TRANSITION TABLE

	0	1
→A	B	C
B	B	D
C	B	C
D	B	E
*E | E	B	C

1. MINIMIZATION OF DFA

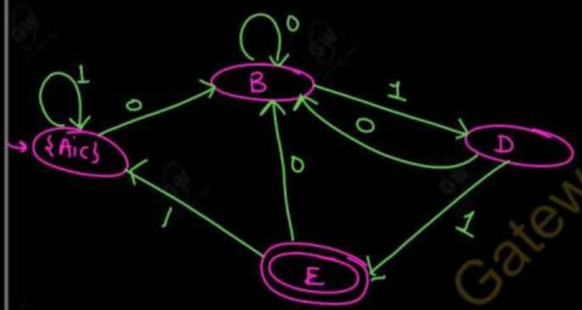
EQUIVALENCE CLASS

$\pi_0 = \{A, B, C, D\} \{E\}$ 0-equivalence class

$\pi_1 = \{A, B, C\} \{D\} \{E\}$ 1-equivalne Clas

$\pi_2 = \{A, C\} \{B\} \{D\}, \{E\}$ – 2-equivalena class

$\pi_3 = \{A, C\} \{B\}, \{D\}, \{E\}$ 3-equivalince



$\{\{A,C\}, B, D, E\}$
$\{A,C\}, E, \{0,1\}$
$\delta, \}$

$M_1 = \{Q_1, q_0, F, \delta_1, \Sigma,$

TRANSITION TABLE

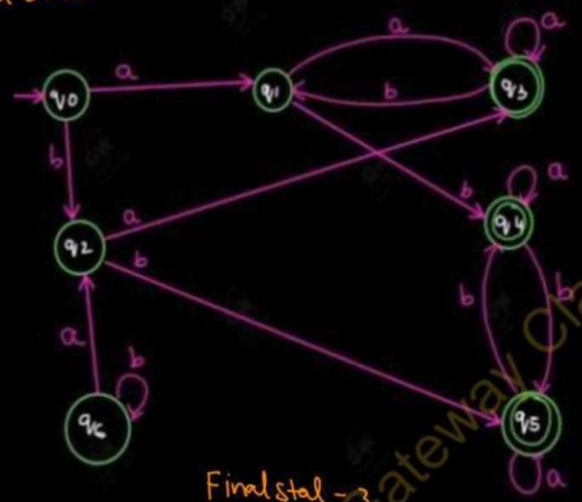
	0	1
→A	B	C
B	B	D
C	B	C
D	B	E
*E	B	C

MINIMIZED TRANSITION TABLE

	0	1
→{A,C}	B	{A,C}
B	B	D
D	B	E
*E	B	{A,C}

2. MINIMIZATION OF DFA

→ unreachable state (Removed
→ Dead State ×



Final state - 3

TRANSITION TABLE

	a	b
→ q_0	q_1	q_2
q_1	q_3	q_4
q_2	q_3	q_5
* q_3	q_3	q_1
* q_4	q_4	q_5
* q_5	q_5	q_4
~~q_6~~	~~q_2~~	~~q_6~~

q_6 unreachable State

2.MINIMIZATION OF DFA

EQUIVALENCE CLASS

$\pi_0 = \{q0, q1, q2\}$ $\{q3, q4, q5\}$ 0-equivalence class

$\pi_1 = \{q0\}, \{q1, q2\}, \{q3\}, \{q4, q5\}$ (1-equivalence class

$\pi_2 = \{q0\}, \{q1\ q2\}, \{q3\}, \{q4, q5\}$

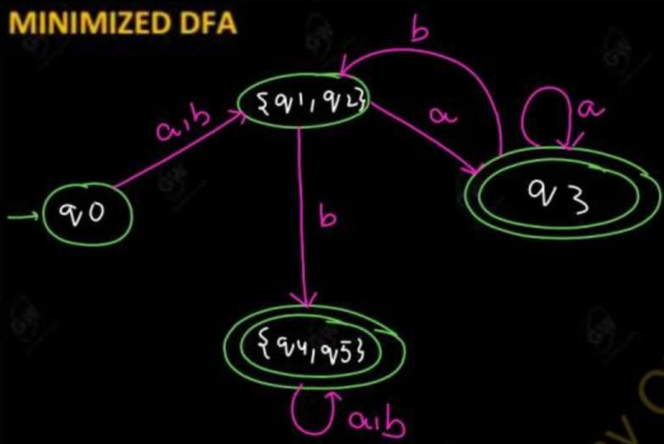
(2 equivalence class

TRANSITION TABLE

	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_3	q_4
q_2	q_3	q_5
$*q_3$	q_3	q_1
$*q_4$	q_4	q_5
$*q_5$	q_5	q_4

2. MINIMIZATION OF DFA

MINIMIZED DFA

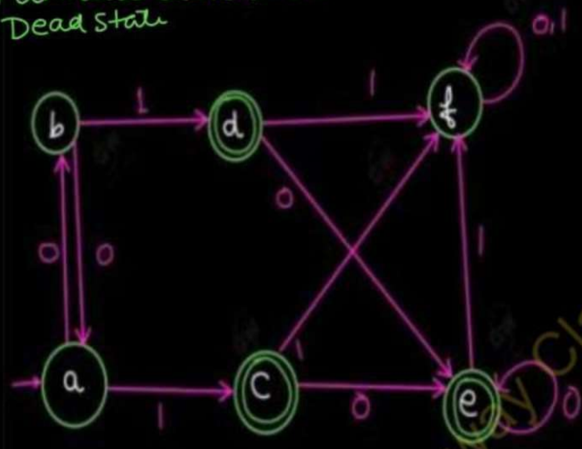


TRANSITION TABLE

	a	b
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	q_3	$\{q_4, q_5\}$
$*q_3$ | q_3	q_3	$\{q_1, q_2\}$
$*\{q_4, q_5\}$ | $\{q_4, q_5\}$	$\{q_4, q_5\}$	$\{q_4, q_5\}$

3 MINIMIZATION OF DFA

→ unreachable state X
→ Dead state



→ 3-final states

TRANSITION TABLE

	0	1
→a	b	c
b	a	d
*c | c	e	f
*d | d	e	f
*e | e	e	f
f ——	f	f

f dead state

3. MINIMIZATION OF DFA

EQUIVALENCE CLASS

$\pi_0 = \{a, b, f\} \{c, d, e\}$ 0-equivalence class

$\pi_1 = \{a, b\} \{f\} \{c, d, e\}$ 1-equivalence class

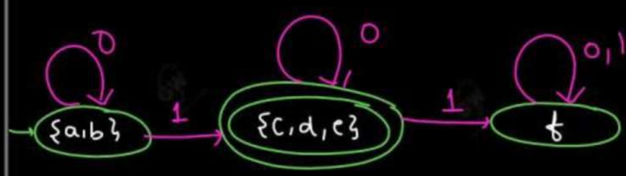
$\pi_2 = \{a, b\}, \{f\} \{c, d, e\}$ 2-equivalence class

TRANSITION TABLE

	0	1
→a	b	c
b	a	d
*c	e	f
*d	e	f
*e	e	f
f	f	f

3 MINIMIZATION OF DFA

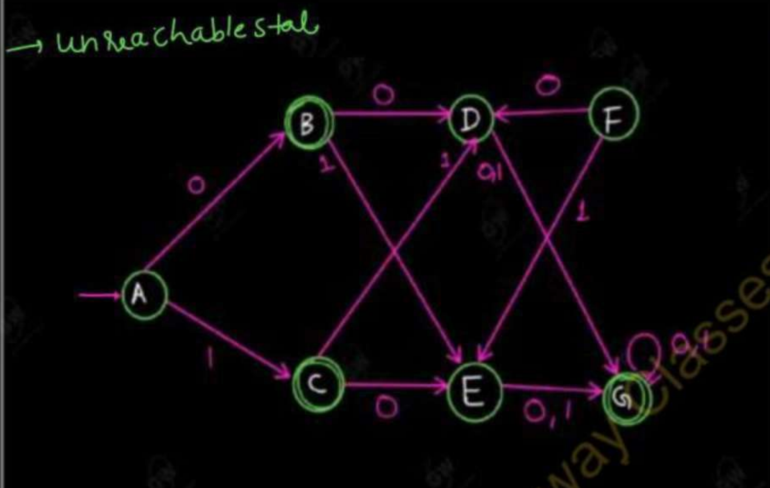
MINIMIZED DFA



TRANSITION TABLE

	0	1
→{a , b}	{a , b}	{c ,d ,e}
*{c, d , e}	{c , d , e}	f
f	f	f

4 MINIMIZATION OF DFA



TRANSITION TABLE

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

F is unreachable state

4 MINIMIZATION OF DFA

EQIVALENCE CLASS

$\pi_0 = \{A, D, E\}\ \{B, C, G\}$ 0-equivalence class

$\pi_1 = \{A, D, E\}\ \{B, C\}\ \{G\}$ 1-equivalence class

$\pi_2 = \{A\}, \{D, E\}\ \{B, C\}, \{G\}$

2-equivalence class

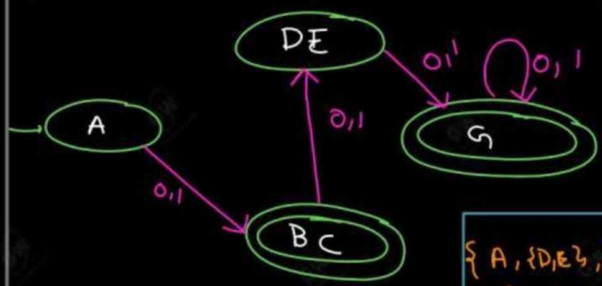
$\pi_3 = \{A\}. \{D, E\}. \{B, C\}. \{G\}$

3-equivalence class

TRANSITION TABLE

	0	1
→A	B	C
*B	D	E
*C	E	D
D	G	G
E	G	G
*G	G	G

4 MINIMIZATION OF DFA



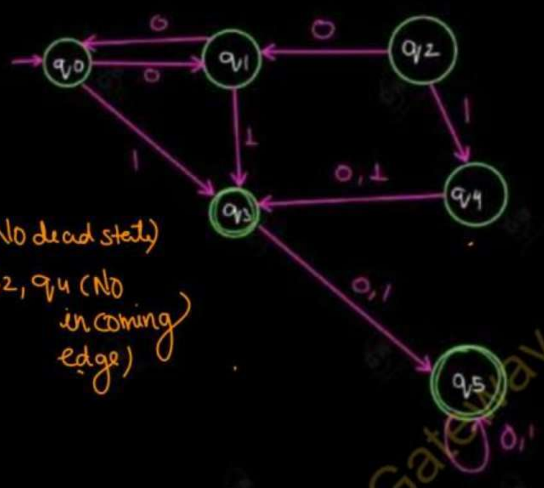
(Minimized dFA)

{A, {D,E}, {B,C}, G, δ, A, {0,1}, {{B,C}, G}}

(Minimized)
TRANSITION TABLE

	0	1
→A	{B,C}	{B,C}
{D,E}	G	G
*{B,C} / BC	{D,E}	{D,E}
*G / G	G	G

5 MINIMIZATION OF DFA



(No dead state)

q_2, q_4 (No incoming edge)

TRANSITION TABLE

	0	1
→q_0	q_1	q_3
q_1	q_0	q_3
× q_2	q_1	q_4
*q_3 | q_3	q_5	q_5
× q_4	q_3	q_3
*q_5 | q_5	q_5	q_5

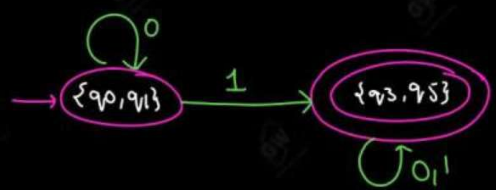
Note

q_2, q_4 are unreachable state

EQUIVALENCE CLASS

$\pi_0 = \{q_0, q_1\} \{q_3, q_5\}$ 0-equivalence class

$\pi_1 = \{q_0, q_1\}, \{q_3, q_5\}$ 1-equivalence class



$\{\{q_0, q_1\} \{q_3, q_5\}\}, \delta', \{0,1\} \{\{q_0, q_1\}\}, \{\{q_3, q_5\}\}$

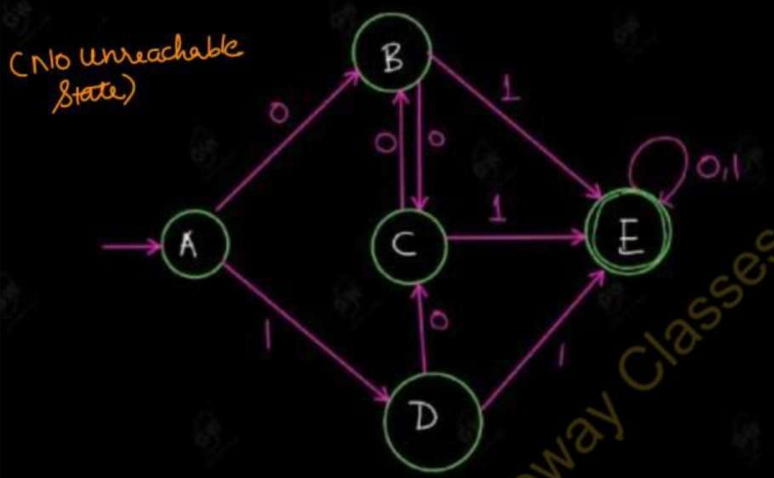
TRANSITION TABLE

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

MINIMIZED TRANSITION TABLE

	0	1
$\rightarrow\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_3, q_5\}$
$*\{q_3, q_5\}$	$\{q_3, q_5\}$	$\{q_3, q_5\}$

6 MINIMIZATION OF DFA



TRANSITION TABLE

	0	1
→A	B	D
B	C	E
C	B	E
D	C	E
*E	E	E

6 MINIMIZATION OF DFA

EQUIVALENCE CLASS

$\pi_0 = \{A, B, C, D\} \{E\}$ 0-equivalence class

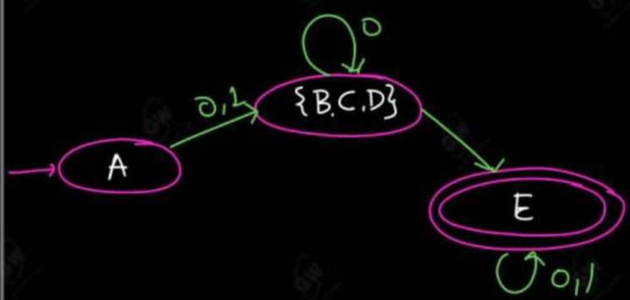
$\pi_1 = \{A\}, \{B, C, D\}, \{E\}$ 1-equivalence Class

$\pi_2 = \{A\}, \{B, C, D\}, \{E\}$ 2-equivalence Class

TRANSITION TABLE

	0	1
→A	B	D
B	C	E
C	B	E
D	C	E
*E	E	E

MINIMIZED DFA

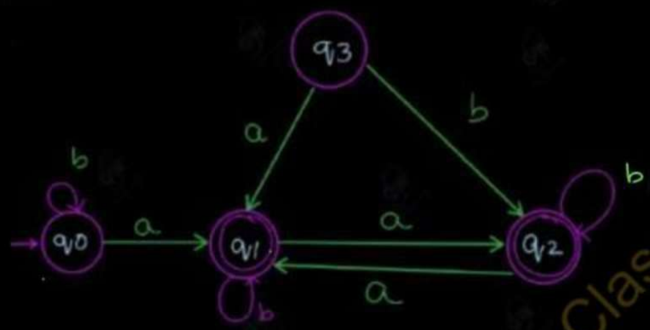


MINIMIZED TRANSITION TABLE

	0	1
→A	{B,C ,D}	{B,C ,D}
{B,C ,D}	{B,C ,D}	E
*E | E	E	E

7 MINIMIZATION OF DFA

unreachable state



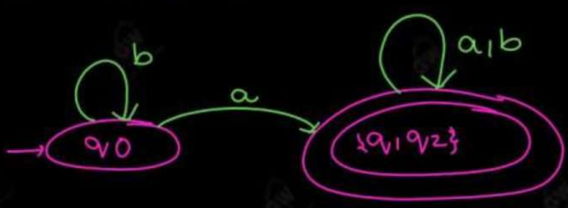
TRANSITION TABLE

	a	b
→q_0	q_1	q_0
*q_1	q_2	q_1
*q_2	q_1	q_2
X q_3	q_1	q_2

q_3 is unreachable state

7 MINIMIZATION OF DFA

$\pi_0 = \{q_0\}\ \{q_1 q_2\}$ 0-equivalence class
$\pi_1 = \{q_0\}\ \{q_1 q_2\}$ 1-equivalence class

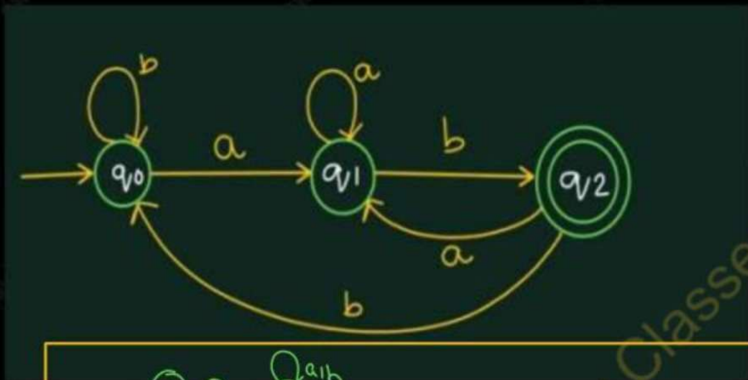


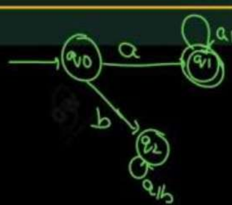
TRANSITION TABLE

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

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

What language accepted by this DFA





a

$\{W \in \{a,b\}^* \mid W \text{ is a substring start with } a\}$

$\{W \in \{a,b\}^* \mid W \text{ is ending with } ab\}$

Explanation

$\in$ belong to

$W \rightarrow ab$ $\quad$ $W \rightarrow abab$ $\quad$ $W \rightarrow abaaabab$

→ dead state → NO → Starting with ab ×

✓ ending with ab

→ Substring ab → × q_2 loop a,b

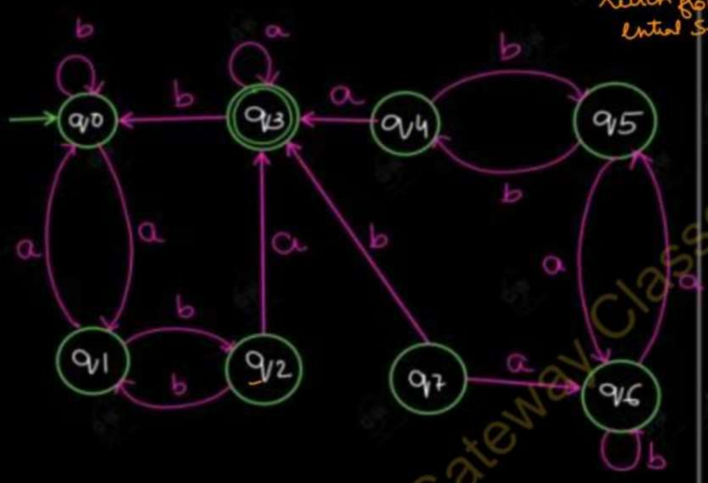
$\Sigma = \{a,b\}$

$\Sigma^* = \{\epsilon, a, b, aa, bb, ab, ba, bbb \ldots\}$

$\{a,b\}^* = \{\epsilon, a, b, aa, bb, ba, bbb \ldots\}$

8.MINIMIZATION OF DFA

unreachable state q4, q5, q6, q7
b/c no way to reach from initial state



TRANSITION TABLE

	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

8.MINIMIZATION OF DFA

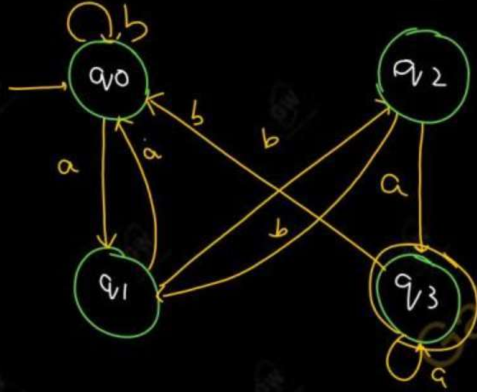
EQUIVALENCE CLASS

$\pi_0 = \{q_0, q_1, q_2\}\ \{q_3\}$ 0-equivalence class

$\pi_1 = \{q_0, q_1\}\ \{q_2\},\ \{q_3\}$ 1-equivalence class

$\pi_2 = \{q_0\}\ \{q_1\}\ \{q_2\},\ \{q_3\}$ 2-equivalence cl

$\pi_3 = \{q_0\},\ \{q_1\},\ \{q_2\}.\ \{q_3\}$ 3-equivalence class



TRANSITION TABLE

	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

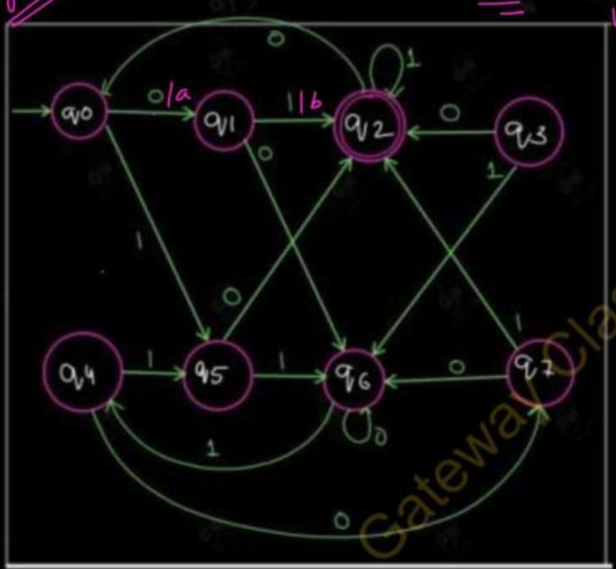
MINIMIZED TRANSITION TABLE

	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

8.MINIMIZATION OF DFA

dfa

let a=0, b=1



TRANSITION TABLE

	a|0	b|1
→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

9.MINIMIZATION OF DFA

EQUIVALENCE CLASS

$\pi_0 = \{q_0, q_1, q_4, q_5, q_6, q_7\}\ \{q_2\}$

$\pi_1 = \{q_0, q_4, q_6\}\{q_1, q_7\}\ \{q_5\}\ \{q_2\}$

$\pi_2 = \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_5\}, \{q_2\}$

$\pi_3 = \{q_0, q_4\}, \{q_6\}\ \{q_1, q_7\}\{q_5\}\ \{q_2\}$

NOTE - if DFA table is Given first make DFA from table Why?

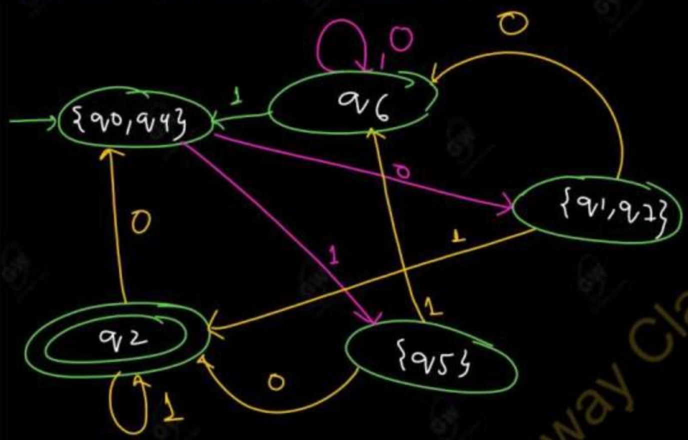
So that you can find unreachable State

TRANSITION TABLE

	a / 0	b / 1
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
$^*q_2$	q_0	q_2
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

9.MINIMIZATION OF DFA

MINIMIZED TRANSITION Daigram



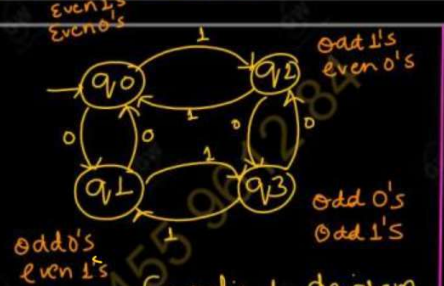
MINIMIZED TRANSITION TABLE

	0/a	1/b
→$\{q_0, q_4\}$	$\{q_1, q_7\}$	q_5
q_6	q_6	$\{q_0, q_4\}$
$\{q_1, q_7\}$	q_6	q_2
q_5	q_2	q_6
*q_2	$\{q_0, q_4\}$	q_2

Construct a DFA that accept the string over Σ ={0,1} which contain

Final

1. Even number of 0 and even number of 1 $\{q_0\}$
2. Odd number of 0 and even number of 1 $\{q_1\}$
3. Even number of 0 or even number of 1 $\{q_0, q_1, q_2\}$
4. Odd number of 0 or even number of 1 $\{q_0, q_1, q_3\}$
5. either odd number of 0 or even number of 1But not both together $\{q_0, q_3\}$
6. odd number of 0 and odd number of 1 $\{q_3\}$
7. Odd number of 0 or odd number of 1 $\{q_1, q_2, q_3\}$

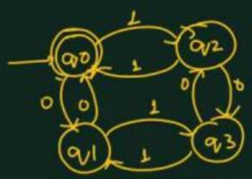


Generalised daigram

Generalised transition table

	0	1
$\rightarrow q_0$	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

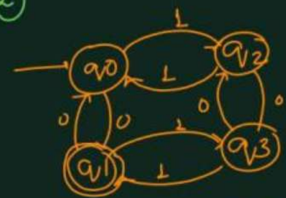
1



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

$\{\{q_0, q_1, q_2, q_3\}, \delta, q_0, q_0, \{0,1\}\}$

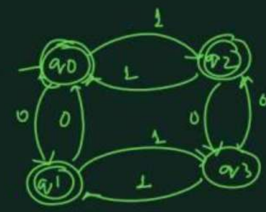
2



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

$\{\{q_0, q_1, q_2, q_3\}, \delta, q_0, q_1, \{0,1\}\}$

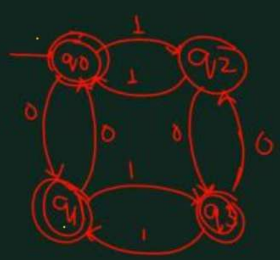
3



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

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

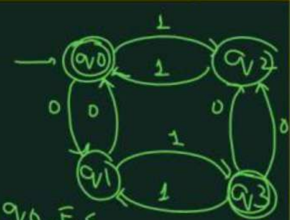
4



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

$\{q_0, q_1 q_2, q_3\}, \delta, q_0, \{q_0, q_1, q_3\}, \{0,1\}\}$

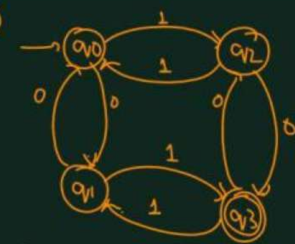
5



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

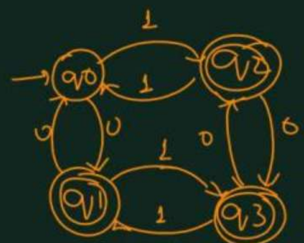
$\{\{q_0, q_1, q_2, q_3\}\}, \delta, q_0, \{q_0, q_3\}\ \{0,1\}\}$

6



$M.\ Q, \delta, q_0, F, \Sigma$

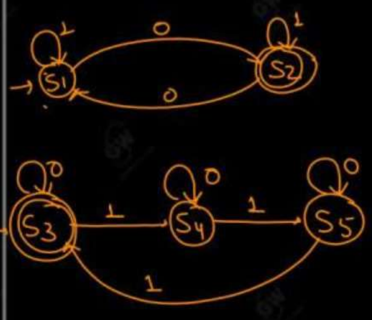
$\{\{q_0, q_1, q_2, q_3\}, \delta, q_0, \{q_3\}\}$

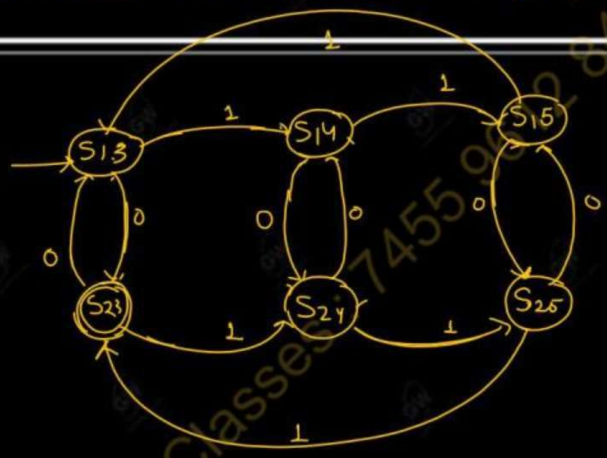


$M = Q, \delta, q_0, F, \Sigma$

$\{q_0, q_1, q_2, q_3\}\ \delta,\ q_0\ \{q_1 q_2, q_3\},\ \{0, 1\}$

Construct a DFA that contain odd number of 0 and 1 is multiple of 3





Construct a DFA that contain odd number of 0 and 1 is multiple of 3 over{0,1}

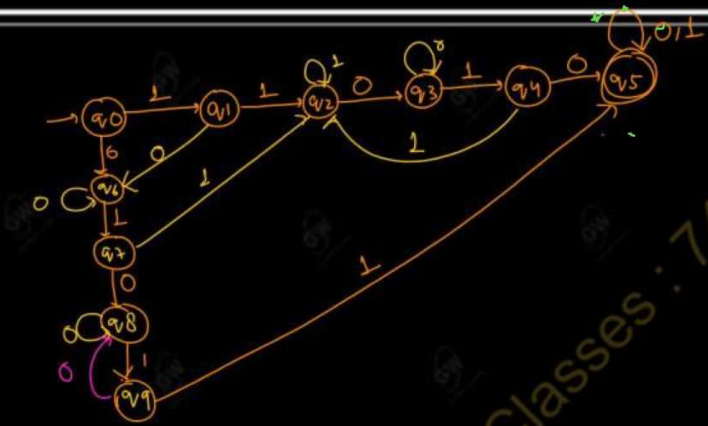
	0	1
→S_{13}	S_{23}	S_{14}
S_{14}	S_{24}	S_{15}
S_{15}	S_{25}	S_{13}
*S_{23}	S_{13}	S_{24}
S_{24}	S_{14}	S_{25}
S_{25}	S_{15}	S_{23}

$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{S_{13}, S_{14}, S_{13}, S_{23}, S_{24}, S_{25}\}, \{0,1\}, \delta, S_{13}, \{S_{23}\}\}$

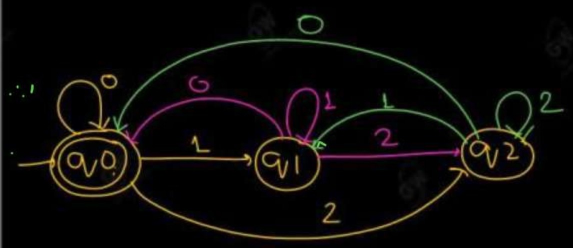
Construct the finite automata that accept all the string containing 11 and 010 as substring

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



Construct a DFA for ternary number divisible by 3

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



TRANSITION TABLE

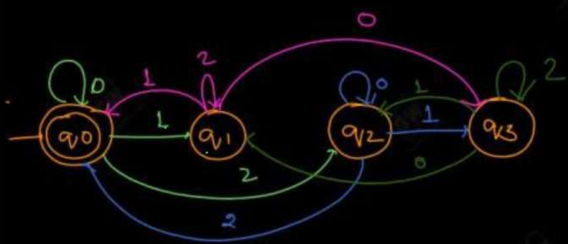
	0	1	2
$\rightarrow {}^{*}q_0$	q_0	q_1	q_2
q_1	q_0	q_1	q_2
q_2	q_0	q_1	q_2

$\{Q, \Sigma, \delta, q_0, F\}$

$\{\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_0\}\}$

Construct a DFA for ternary number divisible by 4

$\Sigma: 0,1,2$



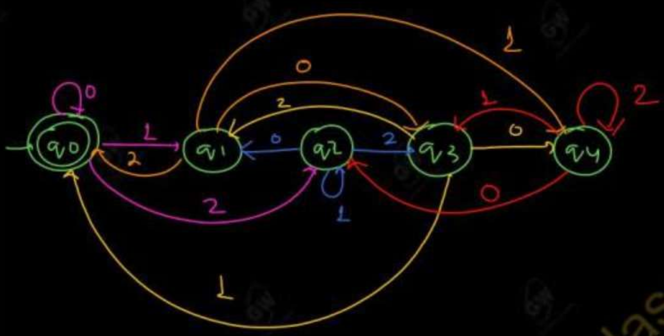
TRANSITION TABLE

	0	1	2
$\rightarrow {}^{*}q_0$	q_0	q_1	q_2
q_1	q_3	q_0	q_1
q_2	q_2	q_3	q_0
q_3	q_1	q_2	q_3

$\{Q, \Sigma, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3,\},\{0,1\}, \delta, q_o, \{q_0\}\}$

Construct a DFA for ternary number divisible by 5.



TRANSITION TABLE

	0	1	2
$\rightarrow {}^*q_0$	q_0	q_1	q_2
q_1	q_3	q_4	q_0
q_2	q_1	q_2	q_3
q_3	q_4	q_0	q_1
q_4	q_2	q_3	q_4

$\{Q, \sum, \delta, q_o, F\}$

$\{\{q_0, q_1, q_2, q_3, q_4\},\{0,1\}, \delta, q_o , \{q_0\}\}$

NFA /NDFA (NON DETERMINISTIC FINITE AUTOMATA)

- It is defined as a 5 tuple, $M=(Q, \Sigma, \delta, q0, F)$

Q: Finite set of states

Σ: Finite set of the input symbol

q0: Initial state

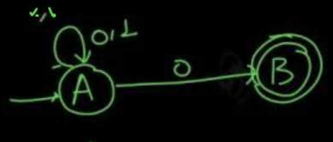
F: Final state $F \subseteq Q$

δ: Transition function: $Q \times \Sigma \rightarrow 2^Q$

NOTE

- In NFA, given the current state there could be multiple next state
- The next state may be chosen at random
- All next state may be chosen in parallel

Construct the NFA over $\underset{\Sigma}{\{0,1\}}$ accept all set of string end with zero



(transition daigram)

Transition table

	0	1
→A	{A,B}	A
B / *B	--	--

M=(Q, Σ, δ,q0,F)= {{A,B}, {0,1}, δ,A,B}

Construct the NFA over {0,1) accept all set of string end with 1

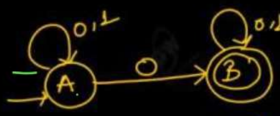
(transition Daigram)

transition table

	0	1
→A	A	{A,B}
*B | B	--	--

M=(Q, Σ, δ,q0,F)= {{A,B}, {0,1}, δ,A,B}

Construct the NFA over {0,1} accept all set of string containing zero



transition Daigram

transition table

	0	1
A	{A,B} A,B	{A}
*B	{B} / B	{B}

M=(Q, Σ, δ,q0,F)

{{A,B}, {0,1}, δ,A,B}

Construct the NFA over {0,1} accept all set of string start with 10

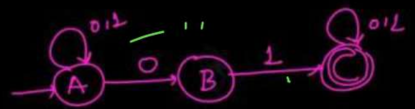
transition Daigram

transition table

	0	1
→A	--	B
B	C	--
*C	C	C

M=(Q, Σ, δ,q0,F) = {{A,B,C}, {0,1}, δ,A,C}

Construct the NFA over {0,1} accept all set of string containing 01 as a substring

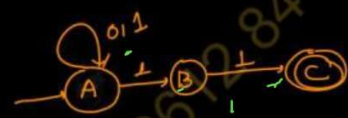


	0	1
→A	A,B	A
B	--	C
*C	C	C

M=(Q, Σ, δ,q0,F)

{{A,B,C}, {0,1}, δ,A,C}

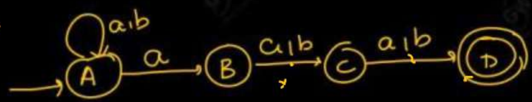
Construct the NFA over {0,1} accept all set of string end with 11



	0	1
→A	{A}	A,B
B	--	C
*C	---	--

M=(Q, Σ, δ,q0,F) = {{A,B,C}, {0,1}, δ,A,C}

Construct a NFA for the language which accept all the string in which third symbol from the right end is always a over{a , b}



1. NFA
2. NFA to DFA conversion – DFA → 90 - 95%
3. Minimization of DFA → 90%

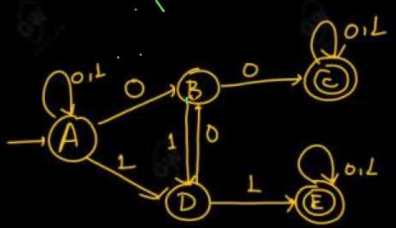
Construct NFA —— DFA (Minimization)
20 - 30%

TRANSITION TABLE

	a	b
→A	A,B	A
B	C	C
C	D	D
*D	-	–

M=(Q, Σ, δ,q0,F)= {{A,B,C,D}, {0,1}, δ,A,D}

Construct a NFA for the language which accept all the string over{0,1} that have at least two consecutive zero or one



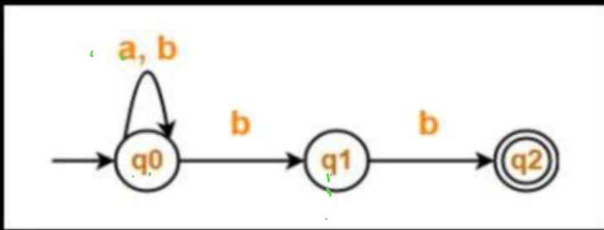
NFA → incomplete

TRANSITION TABLE

	0	1
→A	A,B	A,D
B	C	D
*C	C	C
D	B	E
*E	E	E

M=(Q, Σ, δ,q0,F)= {{A,B,C,D,E}, {0,1}, δ,A, {E,C}}

Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-



(NFA)

Step 1

TRANSITION TABLE OF NFA

State / Alphabet	a	b
→q0	q0	q0, q1
q1	–	*q2
*q2	–	–

TRANSITION TABLE FOR DFA

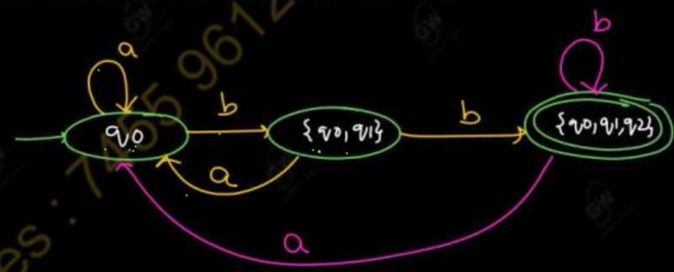
State / Alphabet	a	b
→ q0	q0	{q0, q1}
{q0, q1}	q0	{q0, q1, q2}
{q0, q1, q2}	q0	{q0, q1, q2}

Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-

TRANSITION TABLE FOR DFA

State / Alphabet	a	b
→q0 A	q0	{q0, q1}
{q0, q1} B	q0	{q0, q1, q2}
*{q0, q1, q2} C	q0	{q0, q1, q2}

TRANSITION DAIGRAM FOR DFA



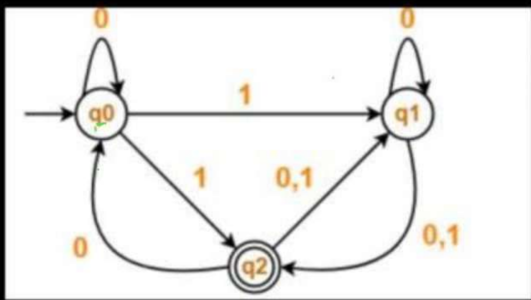
	a	b
→A	A	B
B	A	C
*C	A	C

{A, B} {C} = π_0

{A} {B} {C} = π_1

{A} {B} {C} π_2

Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-



(NFA)

TRANSITION TABLE OF NFA

State / Alphabet	0	1
→q0	q0	q1, q2
q1	q1, q2	q2
*q2	q0, q1	q1

TRANSITION TABLE FOR DFA

State / Alphabet	0	1
→ q0	q0	{q1 q2}
{q1,q2}	{q0, q1, q2}	{q1, q2}
{q0, q1, q2}	{q0, q1, q2}	{q1, q2}

Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-

TRANSITION TABLE FOR DFA

State / Alphabet	0	1
→q0 A	q0 A	{q1, q2} B
*{q1, q2} B	*{q0, q1, q2} C	{q1, q2} B
*{q0, q1, q2} C	{q0, q1, q2} C	{q1, q2} B

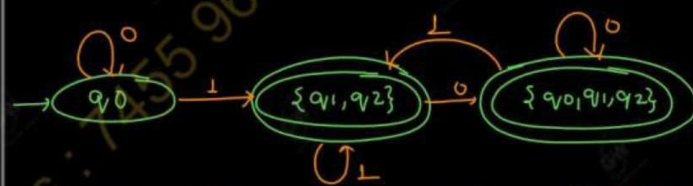
	0	1
A	A	{B,C}
{B,C}	{B,C}	{B,C}

$\pi_0 = \{A\}\ \{B, C\}$

$\pi_1 = \{A\}\ \{B, C\}$

	0	1
A	A	B
B	C	B
C	C	B

TRANSITION DAIGRAM FOR DFA



Minimization DFA (NFA → MinDFA)

① NFA → DFA

DFA → MDFA

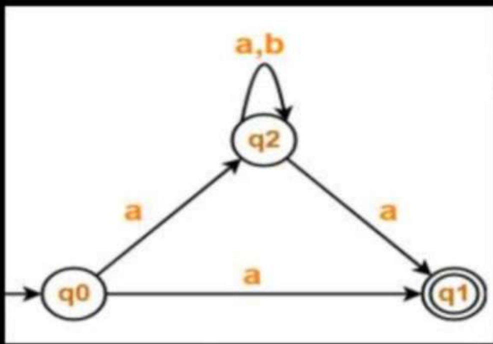
$\pi_0 = \{q_0\}\ \{\{q_1, q_2\}, \{q_0, q_1, q_2\}\}$

$\pi_1 = \{q_0\}\ \{\{q_1, q_2\}, \{q_0, q_1, q_2\}\}$

(MDFA)

	0	1
→q0	q0	{{q1 q2}, {q0, q1, q2}}
*{{q1, q2}, {q0 q1 q2}}	{{q1, q2}, {q0, q1, q2}}	{{q0, q1}, {q0, q1, q2}}

Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-



(NFA)

TRANSITION TABLE OF NFA

State / Alphabet	a	b
→q0	q1, q2	–
*q1	–	–
q2	q1, q2	q2

TRANSITION TABLE FOR DFA

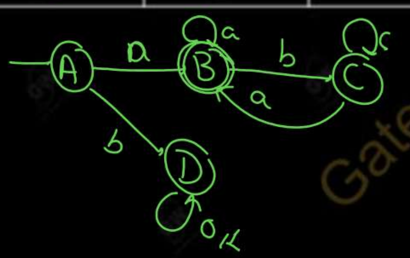
State / Alphabet	a	b
→q0	{q1,q2}	ϕ
{q1,q2}	{q1 q2}	q2
q2	{q1,q2}	q2
ϕ	ϕ	ϕ

ϕ (Dead State)

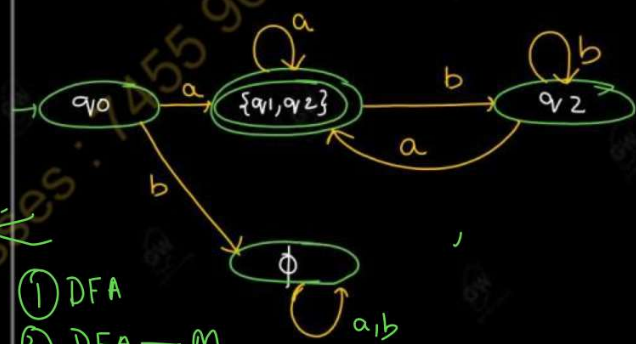
Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-

TRANSITION TABLE FOR DFA

State / Alphabet	a	b
→q0 A	{q1, q2} B	Ø D
*{q1, q2} B	{q1, q2} B	q2 C
q2 C	{q1, q2} B	q2 C
Ø D	Ø D	Ø D

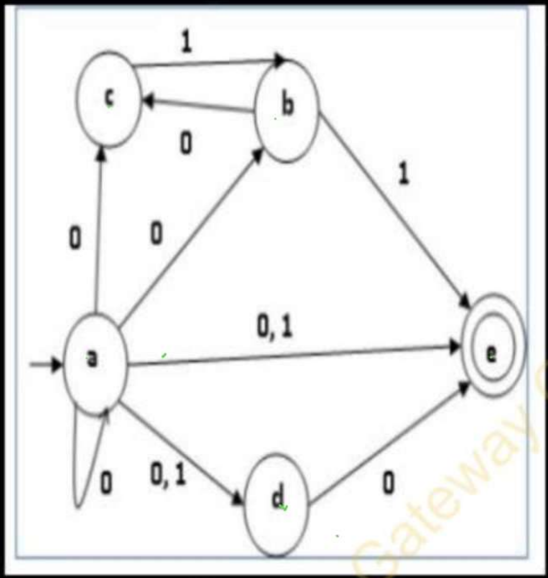


TRANSITION DAIGRAM FOR DFA



① DFA
② DFA — M.
NFA → DFA → m

Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-



(NFA)

TRANSITION TABLE FOR NDFA

State / Alphabet	0	1
→a	a,b,c,d,e	d,e
b	c	e
c	∅	b
d	e	∅
*e	∅	∅

Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)

Transition table for DFA

State / Alphabet	0	1
→ a	{a,b,c,d,e}	{d,e}
*{a,b,c,d,e}	{a,b,c,d,e}	{b,d,e}
*{d,e}	{e}	ϕ
*{b,d,e}	{c,e}	{e}
*{e}	ϕ	ϕ
*{c,e}	ϕ	{b}
{b}	{c}	{e}
{c}	ϕ	{b}
ϕ	ϕ	ϕ

TRANSITION TABLE FOR NDFA

State / Alphabet	0	1
→a	a,b,c,d,e	d,e
b	c	e
c	Ø	b
d	e	Ø
*e	Ø	Ø

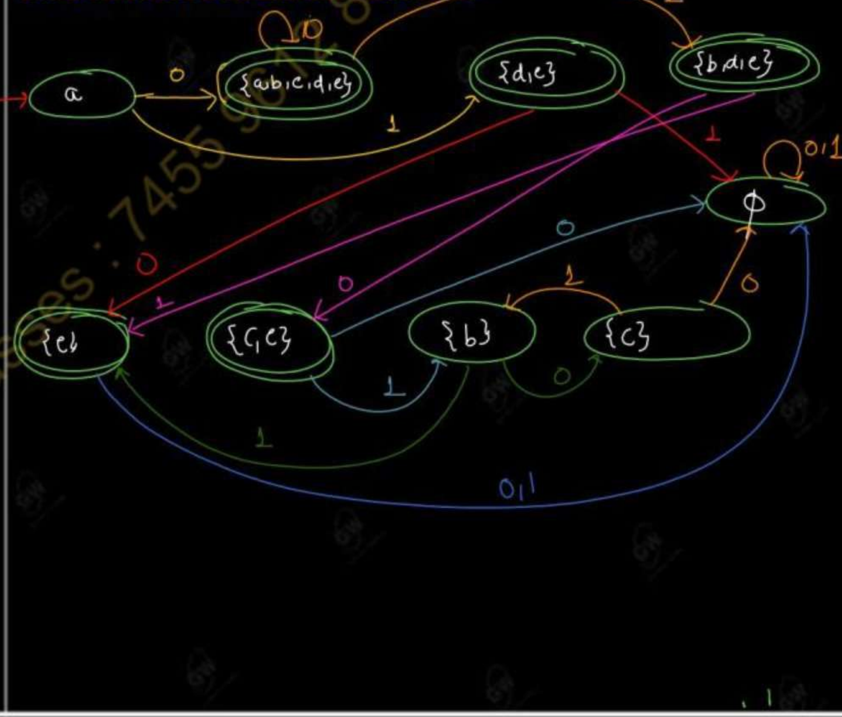
Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-

TRANSITION TABLE FOR DFA

NFA to DFA
DFA to MDF
NFA → MDFA

	0	1
a	{a,b,c,d,e}	{d,e}
*{a,b,c,d,e}	{a,b,c,d,e}	{b,d,e}
*{d,e}	{e}	Ø
*{b,d,e}	{c,e}	{e}
*{e}	Ø	Ø
*{c, e}	Ø	{b}
{b}	{c}	{e}
{c}	Ø	{b}
Ø	Ø	Ø

TRANSITION DAIGRAM FOR DFA



Imp

1. DFA
2. Minization of DFA
3. NFA to DFA
4. ε-NFA to NFA

ε-NFA → Min DF

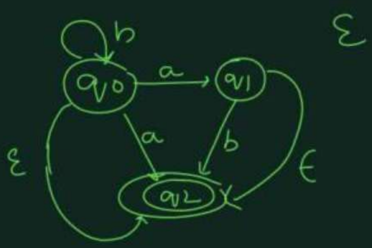
ε-NFA to NFA
NFA to DFA
Min

ε-NFA to DFA
↓
Min

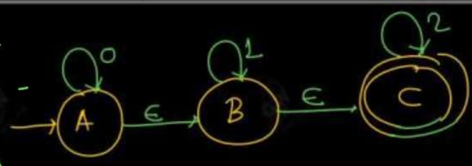
1. DFA
2. Min DFA
3. NFA to DFA
4. NFA to Min DFA
   * NFA to DFA
   * Min of DFA
5. ε-NFA to NFA
6. ε NFA to DFA → ε-NFA to NFA, NFA to DFA
7. ε-NFA to Minimized DFA
   * ε-NFA to NFA
     NFA to DFA
     DFA to Minimized DFA

$\underline{\varepsilon\text{-NFA}}$

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



Q1 EPSILON NFA TO NFA CONVERSION



$\epsilon\text{-closure}(\delta'(\epsilon\text{-closure}(A), 0))$

$\epsilon\text{-closure}(\delta'((A, B, C), 0))$

$(\delta'(A,0) \cup \delta'(B,0) \cup \delta'(C,0))$

$\epsilon\text{-closure}(A \cup \phi \cup \phi)$

A, B, C

TRANSITION TABLE OF εNFA

State / Alphabet	0	1	2	ε
→A	A	-	-	B
B	-	B	-	C
*C	–	-	C	-

EPSILON CLOSURE	
A	A, B, C
B	B, C
C	C

	ε closure	0	ε
A	A B C	A — —	A B C

$\boxed{A \xrightarrow{0} \{A, B, C\}}$

Step1

epsilon closure A

$\{A, B, C\}$

$\delta'(A,0) = A$

$\delta(B,0) = \phi$

$\delta(B,1) = \phi$

$A \cup \phi \cup \phi$

$\{A\}$

epsilon closure(A)

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

Q1 EPSILON NFA TO NFA CONVERSION

State / Alphabet	0	1	2	ε
→A	A	-	-	B
B	–	B	-	C
*C	–	-	C	-

	ε closure	1	ε
A	A $\xrightarrow{1}$ — B $\xrightarrow{1}$ B C $\xrightarrow{1}$ —	— B → —	B, C

epsilon closure (A)
{A, B, C}
δ'(A, 1) = ϕ
δ(B, 1) = B
δ(C, 1) = ϕ
ϕ ∪ B ∪ ϕ
{B}
eplison closure (B)
= B, C

	ε Closure	2 transition of A, B, C on 2	ε
A	A B C	— — C	C

epsilon closure (A)
A, B, C
δ'(A, 2) = ϕ
δ'(B, 2) = ϕ
δ(C, 2) = C
{ϕ ∪ ϕ ∪ C}
C
epsilon (C) = C

Q1 EPSILON NFA TO NFA CONVERSION

State / Alphabet	0	1	2	ε
→A	A	-	-	B
B	–	B	-	C
*C	–	-	C	-

	ε closure	0	ε
B	B $\xrightarrow{0}$ C $\xrightarrow{0}$	ϕ / – ϕ / –	–

	ε closure	1	ε
B	B $\xrightarrow{1}$ C $\xrightarrow{1}$	B $\xrightarrow{ε}$ –	B, C

	ε closure	2	ε
B	B $\xrightarrow{2}$ C $\xrightarrow{2}$	– C →	C

Q1 EPSILON NFA TO NFA CONVERSION

State / Alphabet	0	1	2	ε
→A	A	-	-	B
B	–	B	-	C
*C	–	-	C	-

	ε	0	ε
C	C	ϕ	–

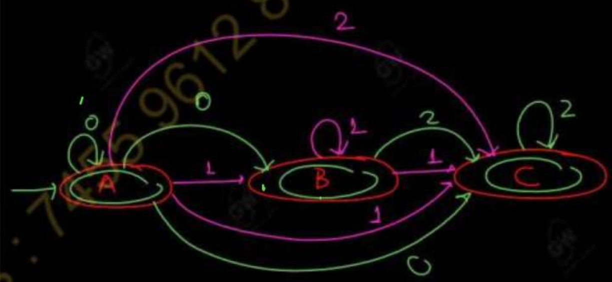
	ε	1	ε
C	C	ϕ	—

	ε	2	ε
C	C	C	C

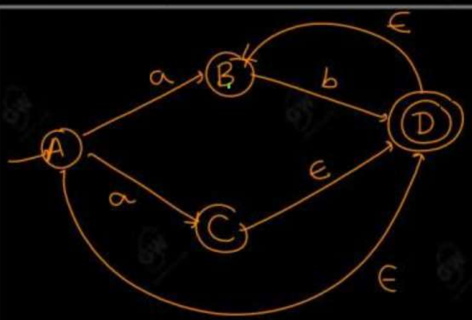
Q1 EPSILON NFA TO NFA CONVERSION

State / Alphabet	0	1	2
→*A	A,B,C	B,C	C
*B	—	B,C	C
*C	—	—	C

(NFA) transition table



Q2 EPSILON NFA TO NFA CONVERSION



	a	b	ε
→A	{B,C}	-	D
B	–	D	-
C	–	-	D
*D	-	-	B

EPSILON CLOSURE	
A	A,D,B
B	B
C	C,D B
D	D, B

Q2 EPSILON NFA TO NFA CONVERSION

State / Alphabet	a	b	ε
→A	{B,C}	-	D
B	–	D	-
C	–	-	D
*D	-	-	B

	ε Closure	a	ε
A	$A \xrightarrow{a}$ $D \xrightarrow{a}$ $B \xrightarrow{a}$	B,C → — —	B,C,D

	ε*	b	ε*
A	$A \xrightarrow{b}$ $D \xrightarrow{b}$ $B \xrightarrow{b}$	ϕ ϕ D →	B,D

	ε* Closure	a	ε*
B	$B \xrightarrow{a}$	ϕ	—

Q2 EPSILON NFA TO NFA CONVERSION

State / Alphabet	a	b	ε
→A	{B,C}	-	D
B	–	D	-
C	–	-	D
*D	-	-	B

	ε	b	ε
B	$B \xrightarrow{b} D \xrightarrow{\epsilon} B, D$		

	ε Closure	a	ε Clos
C	$C \xrightarrow{a} \phi$ $D \xrightarrow{a} \phi$ $B \xrightarrow{a} \phi$		— — —

	ε Closure	b	ε Closure
C	$C \xrightarrow{b} \phi$ $D \xrightarrow{b} \phi$ $B \xrightarrow{b} D \xrightarrow{\epsilon} D, B$		

Q2 EPSILON NFA TO NFA CONVERSION

State / Alphabet	a	b	ε
→A	{B,C}	-	D
B	–	D	-
C	–	-	D
*D	-	-	B

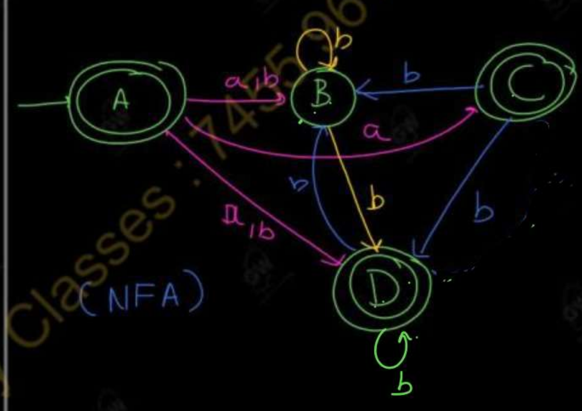
	① ε Closure	② a	③ ε Closure
D	$D \xrightarrow{a}$ — $B \xrightarrow{a}$ —		—

	ε	b	ε
D	$D \xrightarrow{b}$ — $B \xrightarrow{b}$	D	$\xrightarrow{\epsilon}$ B,D

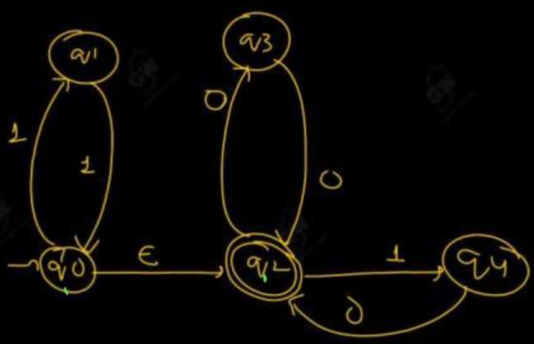
Q2 EPSILON NFA TO NFA CONVERSION

State / Alphabet	a	b
→A	B,C,D	B,D
B	—	B,D
C	—	B,D
D	—	B,D

(NFA)



Q3 EPSILON NFA TO NFA CONVERSION



transition table of ε-NFA

	0	1	ε
q0	-	q1	q2
q1	-	q0	-
*q2	q3	q4	-
q3	q2	-	-
q4	q2	-	-

Q3 EPSILON NFA TO NFA CONVERSION

	Epsilon closure
q0	{q0,q2}
q1	q1
q2	q2
q3	q3
q4	q4

	0	1	ε
q0	-	q1	q2
q1	-	q0	-
*q2	q3	q4	-
q3	q2	-	-
q4	q2	-	-

Q3 EPSILON NFA TO NFA CONVERSION

	ε closure	0	ε closure
q0	q0 q2	— q3	 → q3

	ε Closure	1	ε Closure
q0	q0 q2	q1 q4	q1, q4

	0	1	ε
q0	-	q1	q2
q1	-	q0	-
*q2	q3	q4	-
q3	q2	-	-
q4	q2	-	-

Q3 EPSILON NFA TO NFA CONVERSION

	ε Closure	0	ε
q1	q1	φ	—

	ε Closure	1	ε Closure
q1	$q_1 \xrightarrow{1} q_0$		{q0, q2}

	0	1	ε
q0	-	q1	q2
q1	-	q0	-
*q2	q3	q4	-
q3	q2	-	-
q4	q2	-	-

Q3 EPSILON NFA TO NFA CONVERSION

	ε	0	ε Closure
q2	$q_2 \xrightarrow{0}$	q3	q3

	ε	1	ε Closure
q2	$q_2 \xrightarrow{1}$	q4	q4

	0	1	ε
q0	-	q1	q2
q1	-	q0	-
*q2	q3	q4	-
q3	q2	-	-
q4	q2	-	-

Q3 EPSILON NFA TO NFA CONVERSION

	ε	0	ε
q3	q3	q2	q2

	ε	1	ε
q3	q3	—	—

	0	1	ε
q0	-	q1	q2
q1	-	q0	-
*q2	q3	q4	-
q3	q2	-	-
q4	q2	-	-

Q3 EPSILON NFA TO NFA CONVERSION

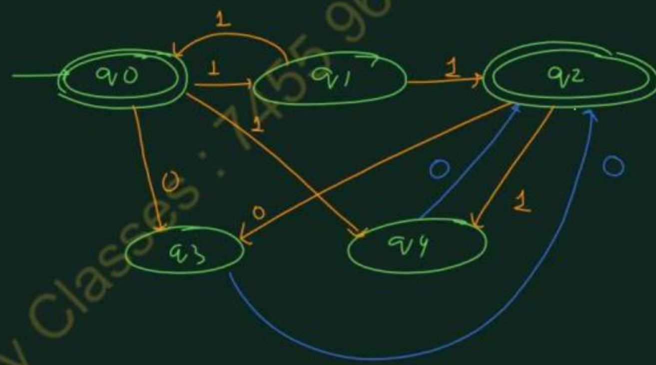
	ε Closure	0	ε Closure
q4	q4	q2	q2

	ε Closure	1	ε Closure
q4	q4 $\xrightarrow{1}$	–	–

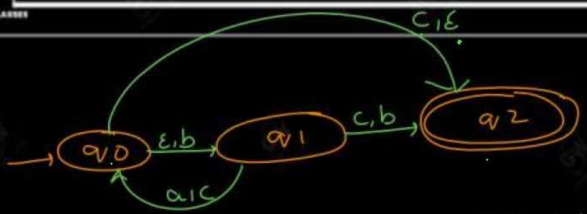
	0	1	ε
q0	-	q1	q2
q1	-	q0	-
*q2	q3	q4	-
q3	q2	-	-
q4	q2	-	-

NFA

	0	1
→ q0	q3	q1, q4
q1	–	q0, q2
q2	q3	q4
q3	q2	–
q4	q2	–



Convert following EPSIILON NFA to equivalent MDFA



q0,q2 – Final State

(transition diagram for ε-NFA)

Σ = {a,b,c}

TRANSITION TABLE OF εNFA

State / Alphabet	a	b	c	ε
→q0	_	q1	q2	q1,q2
q1	q0	q2	q0,q2	_
*q2	–	--	--	-

EPSILON CLOSURE	
q0	q0,q1,q2
q1	q1
q2	q2

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF εNFA

	a	b	c	ε
→q0	_	q1	q2	q1,q2
q1	q0	q2	q0,q2	_
*q2	–	--	--	-

	ε closure	a		ε closure
q0	$q_0 \xrightarrow{a} \phi$ $q_1 \xrightarrow{a} q_0$ $q_2 \xrightarrow{a} \phi$		$\phi \cup q_0 \cup \phi$ $= q_0 \rightarrow$	$q_0 q_1 q_2$

	ε closure	b	ε closure
q0	$q_0 \xrightarrow{b} q_1$ $q_1 \xrightarrow{b} q_2$ $q_2 \xrightarrow{b} \phi$	$q_1 \xrightarrow{\epsilon\ closure}$ $q_2 \xrightarrow{\epsilon\ closure}$	q_1 q_2 $\{q_1, q_2\}$

	ε closure	c		ε closure
q0	$q_0 \xrightarrow{c} q_2$ $q_1 \xrightarrow{c} \{q_0, q_2\}$ $q_2 \xrightarrow{c}$	$q_2 \cup \{q_0, q_2\}$	$q_0 \xrightarrow{\epsilon\ closure}$ $q_2 \xrightarrow{\epsilon\ closure}$	q_0, q_1, q_2 q_2 $\downarrow$ q_0, q_1, q_2

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF εNFA

	a	b	c	ε
→q0	_	q1	q2	q1,q2
q1	q0	q2	q0,q2	_
*q2	–	--	--	-

	ε closure	a	ε closure
q1	$q1 \xrightarrow{a} q0 \xrightarrow{\epsilon\ closure} \{q0,q1,q2\}$		

	ε closure	b	ε closure
q1	$q1 \xrightarrow{b} q2$		q2

	ε closure	c	ε closure
q1	$q1 \xrightarrow{c} q0 \xrightarrow{\epsilon\ closure} q0,q1,q2$		
	$q2 \xrightarrow{\epsilon\ closure} q2$		

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF εNFA

	a	b	c	ε
→q0	_	q1	q2	q1,q2
q1	q0	q2	q0,q2	_
*q2	–	--	--	-

	ε closure	a	ε closure
q2	q2 $\xrightarrow{a}$	φ	—

	ε closure	b	ε closure
q2	q2 →	φ	—

	ε closure	c	ε closure
q2	q2 →	φ	—

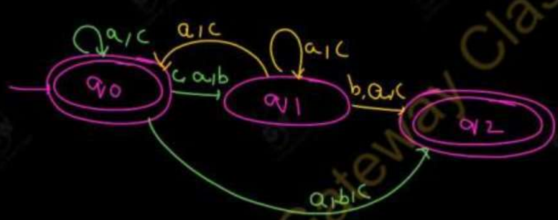
Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF NFA

Input → / State ↓	a	b	c
→* q_0	{q_0, q_1, q_2}	{q_1, q_2}	{q_0, q_1 q_2}
q_1	{q_0, q_1, q_2}	q_2	{q_0, q_1, q_2}
* q_2	—	—	—

ε-NFA to MDFA

1. ε NFA to NFA
2. NFA to DFA
3. DFA to MDFA



(Transition Diagram for (NFA)

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF NFA

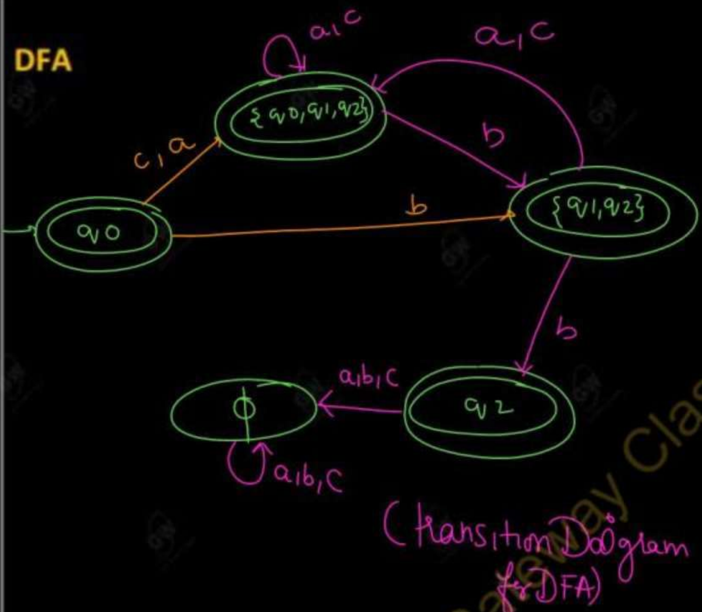
	a	b	c
→*q0	{q0,q1,q2}	{q1,q2}	{q0,q1,q2}
q1	{q0,q1,q2}	q2	{q0,q1,q2}
*q2	–	--	--

DFA (NFA TO DFA)

	a	b	c
→*q0	{q0,q1,q2}	{q1,q2}	{q0,q1,q2}
*{q0,q1,q2}	{q0,q1,q2}	{q1,q2}	{q0,q1,q2}
*{q1,q2}	{q0,q1,q2}	q2	{q0,q1,q2}
*q2	ϕ	ϕ	ϕ
ϕ	ϕ	ϕ	ϕ

Convert following NFA to equivalent DFA and MDFA

DFA



DFA (NFA TO DFA)

(transition table DFA)

	a	b	c
→*q0	{q0,q1,q2}	{q1,q2}	{q0,q1,q2}
*{q0,q1,q2}	{q0,q1,q2}	{q1,q2}	{q0,q1,q2}
*{q1,q2}	{q0,q1,q2}	q2	{q0,q1,q2}
*q2	Φ	Φ	Φ
Φ	Φ	Φ	Φ

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF DFA

	a	b	c
→*A	B	C	B
*B	B	C	B
*C	B	D	B
*D	Φ	Φ	Φ
Φ	Φ	Φ	Φ

DFA (NFA TO DFA)

let	a	b	c
→*q0 (A)	{q0,q1,q2}	{q1,q2}	{q0,q1,q2}
*{q0,q1,q2} (B)	{q0,q1,q2}	{q1,q2}	{q0,q1,q2}
*{q1,q2} (C)	{q0,q1,q2}	q2	{q0,q1,q2}
*q2 (D)	Φ	Φ	Φ
Φ	Φ	Φ	Φ

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF DFA

	a	b	c
→ *A	B	C	B
*B	B	C	B
*C	B	D	B
*D	Φ	Φ	Φ
Φ	Φ	Φ	Φ

MINIMIZATION OF DFA

$\pi_0 = \underbrace{\{A, B, C, D\}}_{q_1} \; \underbrace{\{\phi\}}_{q_2}$

$\pi_1 = \{A, B, C\} \{D\} \{\phi\}$

$\pi_2 = \{A, B\} \{C\}, \{D\} \; \{\phi\}$

$\pi_2 = \{A, B\} \; \{C\}, \{D\}, \{\phi\}$

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF DFA

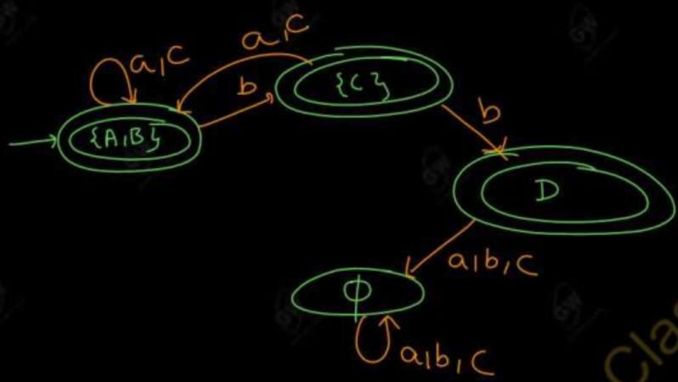
	a	b	c
*A	B	C	B
*B	B	C	B
*C	B	D	B
*D	Φ	Φ	Φ
Φ	Φ	Φ	Φ

	a	b	c
→*{A,B}	{A,B}	C	{A,B}
*C	{A,B}	D	{A,B}
*D	φ	φ	φ
φ	φ	φ	φ

{minimized DFA table}

Convert following NFA to equivalent DFA and MDFA

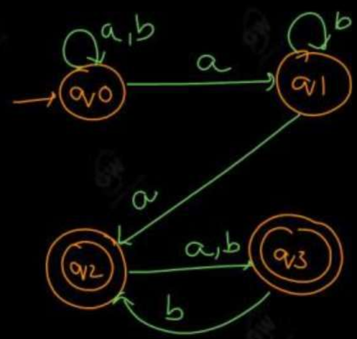
MINIMIZED DFA



	a	b	c
→ * {A,B}	{A,B}	{C}	{A,B}
* {C}	{A,B}	{D}	{A,B}
* {D}	Φ	Φ	Φ
Φ	Φ	Φ	Φ

{minimized DFA table}

Construct the minimum automata equivalent to NFA



(transition Diagram for (NFA))

Transition table for NFA

	a	b
→q0	q0,q1	q0
q1	q2	q1
*q2	q3	q3
*q3	-	q2

Construct the minimum automata equivalent to NFA

Transition table DFA

	a	b
→ q0	{q0, q1}	q0
{q0, q1}	{q0, q1, q2}	{q0, q1}
*{q0, q1, q2}	{q0, q1, q2, q3}	{q0, q1, q3}
*{q0, q1, q2, q3}	{q0, q1, q2, q3}	{q0, q1, q2, q3}
*{q0, q1, q3}	{q0, q1, q2}	{q0, q1, q2}

Transition table for NFA

	a	b
→q0	q0,q1	q0
q1	q2	q1
*q2	q3	q3
*q3	-	q2

Construct the minimum automata equivalent to NFA

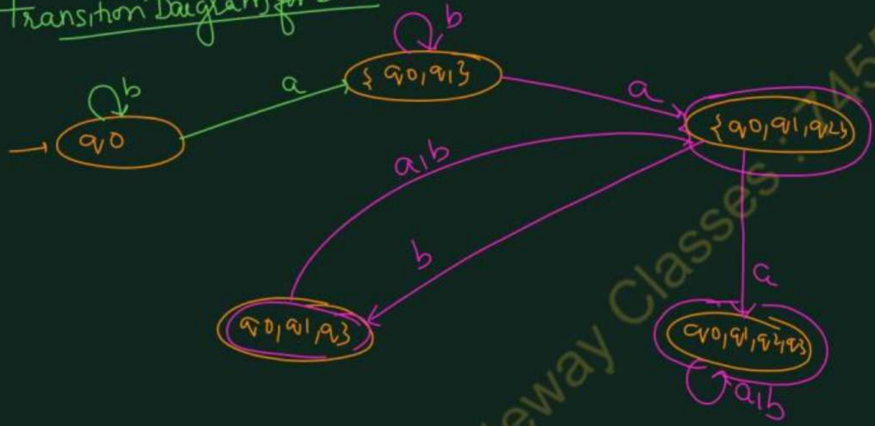
Transition table DFA

	a	b
→q0 0	{q0,q1}	q0
{q0,q1} 1	{q0,q1,q2}	{q0,q1}
*{q0,q1,q2} 2	{q0,q1,q2,q3}	{q0,q1,q3}
*{q0,q1,q2,q3} 3	{q0,q1,q2,q3}	{q0,q1,q2,q3}
*{q0,q1,q3} 4	{q0,q1,q2)	{q0,q1,q2}

Transition table for DFA

	a	b
→0	1	0
1	2	1
*2	3	4
*3	3	3
*4	2	2

Transition Daigram for DFA



Construct the minimum automata equivalent to NFA

MINIMIZATION OF DFA

$\pi_0 = \{0,1\}, \{2,3,4\}$ 0-equivalence class

$\pi_1 = \{0\}, \{1\}, \{2,3,4\}$ 1-equivalence class

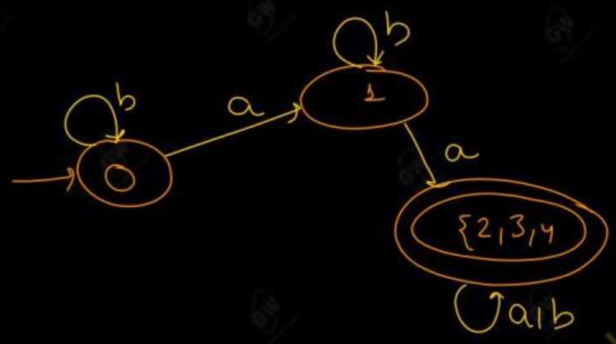
$\pi_2 = \{0\} \{1\}, \{2,3,4\}$ 2-equivalence class

Transition table for DFA

	a	b
→0	1	0
1	2	1
*2	3	4
*3	3	3
*4	2	2

Construct the minimum automata equivalent to NFA

MINIMIZATION OF DFA (Transition Diagram)

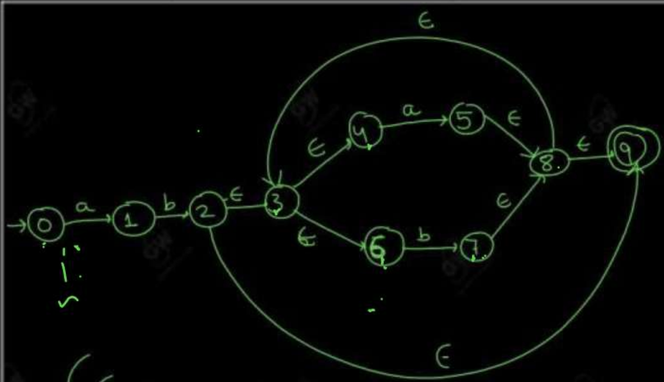


Transition table for DFA

	a	b
→0	1	0
1	{2,3,4}	1
*{2,3,4}	{2,3,4}	{2,3,4}

Question. EPSILON NFA TO DFA CONVERSION

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



(Epsilon NFA transition Daigram)

	a	b	ε (Empty string
→0	1	-	-
1	-	2	-
2	-	-	3,9
3	-	-	4,6
4	5	-	-
5	-	-	8
6	-	7	-
7	-	-	8
8			9,3
*9	-	-	-

Epsilon NFA table

	Epsilon closure
0	0
1	1
2	2,3,9,4,6
3	3,4,6
4	4
5	5,8,9,3,4,6
6	6
7	7,8,9,3,4,6
8	8,9,3,4,6
9	9

	a	b	ε
→0	1	-	-
1	-	2	-
2	-	-	3,9
3	-	-	4,6
4	5	-	-
5	-	-	8
6	-	7	-
7	-	-	8
8			9,3
*9	-	-	-

Initial State Of NFA	Epsilon closure	a	Epsilon closure	b	Epsilon closure
0	0	1	1	ϕ	—

	a	b	ε
→0	1	-	-
1	-	2	-
2	-	-	3,9
3	-	-	4,6
4	5	-	-
5	-	-	8
6	-	7	-
7	-	-	8
8			9,3
*9	-	-	-

State	a	Epsilon closure	b	Epsilon closure
1	ϕ	—	2	{2,3,9,4,6}
{2,3,9,4,6}	5	5,8,9,3,4,6	7	7,8,9,3,4,6
{5,8,9,3,4,6	5	5,8,9,3,4,6	7	7,8,9,3,4,6
7,8,9,3,4,6	5	5,8,9,3,4,6	7,8	7,8,9,3,4,6
ϕ	—	—	—	—

	a	b	ε
→0	1	-	-
1	-	2	-
2	-	-	3,9
3	-	-	4,6
4	5	-	-
5	-	-	8
6	-	7	-
7	-	-	8
8			9,3
*9	-	-	-

	a	b
→ 0	1	ϕ
1	ϕ	{2,3,9,4,6}
* {2,3,9,4,6}	{5,8,9,3,4,6}	{7,8,9,3,4,6}
* {5,8,9,3,4,6}	{5,8,9,3,4,6}	{7,8,9,3,4,6}
* {7,8,9,3,4,6}	5,8,9,3,4,6	{7,8,9 3,4,6}
ϕ	ϕ	ϕ

DFA table

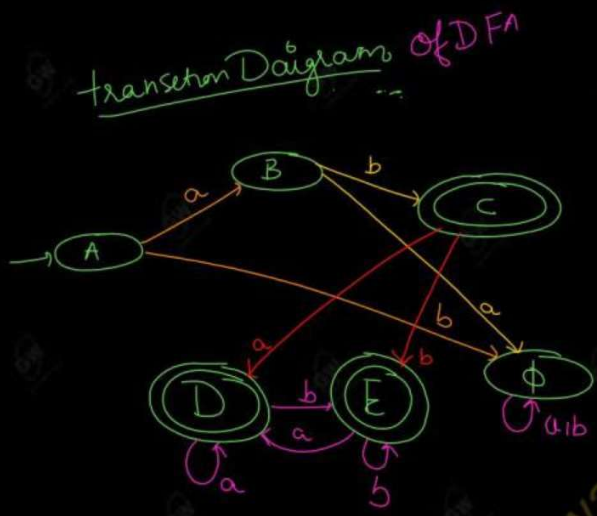
ϵNFA → 9

DFA → Final state - 9 present

		a	b
A	→0	1	ϕ
B	1	ϕ	{2,3,9,4,6}
C	*{2,3,9,4,6}	{5,8,3,4,6,9}	{7,8,3,9,4,6}
D	*{5,8,3,4,6, 9}	{5,8,3,4,6,9}	{7,8,3,9,4,6}
E	*{7,8,3,9,4, 6}	{5,8,3,4,6,9}	{7,8,3,9,4,6}
	ϕ	ϕ	ϕ

DFA table

	a	b
→A	B	ϕ
B	ϕ	C
*C	D	E
*D	D	E
*E	D	E
ϕ	ϕ	ϕ



	a	b
→A	B	ϕ
B	ϕ	C
*C	D	E
*D	D	E
*E	D	E
ϕ	ϕ	ϕ

Only if ENFA to Minimized DFA Asked?

MINIMIZED DFA

$\pi_0 = \{A, B, \phi\} \quad \{C, D, E\}$ 0-equivalence class

$\pi_1 = \{A, \phi\}, \{B\}, \{C, D, E\}$ 1-equivalence class

$\pi_2 = \{A\}, \{B\} \{\phi\}, \{C, D, E\}$ 2-equivalence class

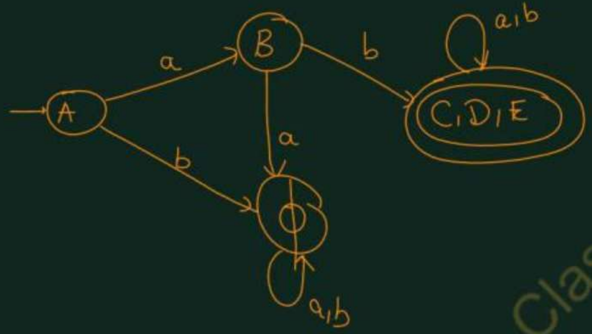
$\pi_3 = \{A\}, \{B\}, \{\phi\}, \{C, D, E\}$ 3-equivalence class

	a	b
→A	B	ϕ
B	ϕ	C
*C	D	E
*D	D	E
*E	D	E
ϕ	ϕ	ϕ

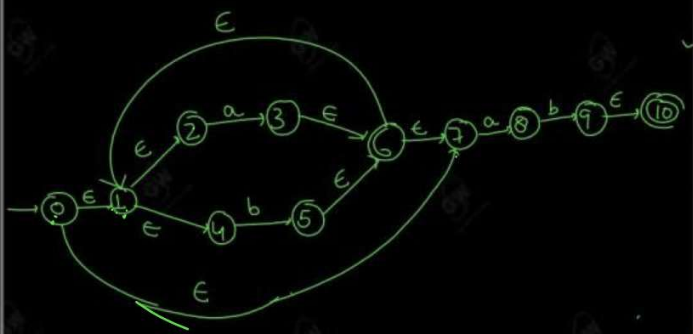
	a	b
→A	B	ϕ
B	ϕ	{C,D,E}
*{C,D,E}	{C,D,E}	{C,D,E}
ϕ	ϕ	ϕ

Minimized DFA table

	a	b
→A	B	ϕ
B	ϕ	C
*C	D	E
*D	D	E
*E	D	E
ϕ	ϕ	ϕ


A
B
C,D,E
φ
a
b
a
b
a,b
a,b

Question. EPSILON NFA TO DFA CONVERSION



	a	b	ε
→0	-	-	1,7
1	-	-	2,4
2	3	-	-
3	-	-	6
4	-	5	-
5	-	--	6
6	-	-	1,7
7	8	-	-
8	-	9	-
9	-	10	-
10	-	--	-

	EPSILON CLOSURE
0	0,1,2,4,7
1	1,2,4
2	2
3	3,6,1,2,4,7// 1,2,3,4,6,7
4	4
5	5,6,7,1,2,4//1,2,4,5,6,7
6	6,7,1,2,4//1,2,4,6,7
7	7
8	8
9	9
10	10

	a	b	ε
→0	-	-	1,7
1	-	-	2,4
2	3	-	-
3	-	-	6
4	-	5	-
5	-	--	6
6	-	-	1,7
7	8	-	-
8	-	9	-
9	-	10	-
10	-	--	-

Initial State Of NFA	Epsilon closure	a	Epsilon closure	b	Epsilon closure
0	{0,1,2,4,7}	3,8	1,2,3,4,6,7,8	5	{1,2,4,5,6,7}

1,2,3,4,6,7 U 8

	a	b	ε
→0	-	-	1 ,7
1	-	-	2,4
2	3	-	-
3	-	-	6
4	-	5	-
5	-	--	6
6	-	-	1,7
7	8	-	-
8	-	9	-
9	-	10	-
10	-	--	-

State	a	Epsilon closure	b	Epsilon closure
1,2,3,4,6,7,8	3,8	1,2,3,4,6,7,8	5,9	1,2,4,5,6,7,9
1,2,4,5,6,7	3,8	1,2,3,4,6,7,8	5	1,2,4,5,6,7
1,2,4,5,6,7,9	3,8	1,2,3,4,6,7,8	5,10	1,2,4,5,6,7,10
1,2,4,5,6,7,10	3,8	1,2,3,4,6,7,8	5	1,2,4,5,6,7,

	a	b	ε
→0	-	-	1,7
1	-	-	2,4
2	3	-	-
3	-	-	6
4	-	5	-
5	-	--	6
6	-	-	1,7
7	8	-	-
8	-	9	-
9	-	10	-
10	-	--	-

DFA FINAL TABLE

	a	b
→{0,1,2,4,7}	{1,2,3,4,6,7,8}	{1,2,4,5,6,7}
{1,2,3,4,6,7,8}	{1,2,3,4,6,7,8}	{1,2,4,5,6,7,9}
{1,2,4,5,6,7}	{1,2,3,4,6,7,8}	{1,2,4,5,6,7}
{1,2,4,5,6,7,9}	{1,2,3,4,6,7,8}	{1,2,4,5,6,7,10}
*{1,2,4,5,6,7,10}	{1,2,3,4,6,7,8}	{1,2,4,5,6,7}

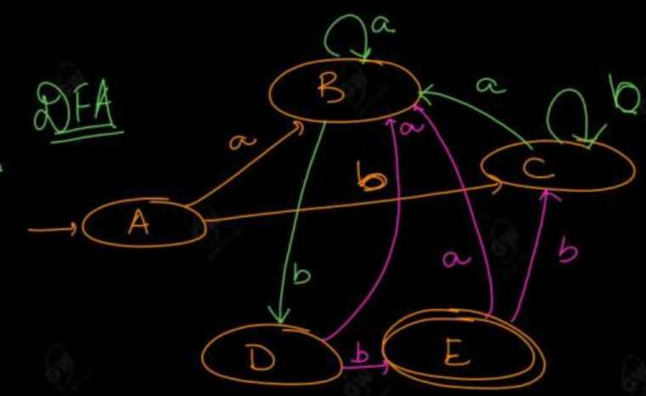
DFA table

	a	b
->{0,1,2,4,7} let A	{1,2,3,4,6,7,8}	{1,2,4,5,6,7}
{1,2,3,4,6,7,8} B	{1,2,3,4,6,7,8}	{1,2,4,5,6,7,9}
{1,2,4,5,6,7} C	{1,2,3,4,6,7,8}	{1,2,4,5,6,7}
{1,2,4,5,6,7,9} D	{1,2,3,4,6,7,8}	{1,2,4,5,6,7,10}
* {1,2,4,5,6,7,10} E	{1,2,3,4,6,7,8}	{1,2,4,5,6,7}

DFA table

	a	b
→A	B	C
B	B	D
C	B	C
D	B	E
*E	B	C

DFA



	a	b
→A	B	C
B	B	D
C	B	C
D	B	E
*E	B	C

	a	b
→A	B	C
B	B	D
C	B	C
D	B	E
*E	B	C

Minimized DFA

$\pi_0 = \{A, B, C, D\} \{E\}$

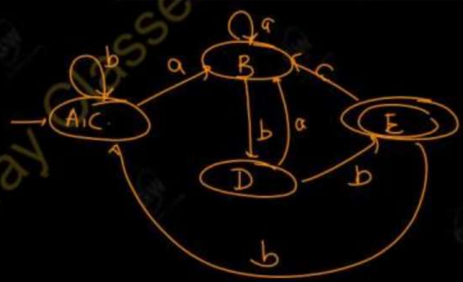
$\pi_1 = \{A, B, C\}, \{D\}, \{E\}$

$\pi_2 = \{A, C\} \{B\}, \{D\} \ \{E\}$

$\pi_3 = \{A, C\} \{B\}, \{D\}, \{E\}$

	a	b
→{A,C}	B	{A,C}
B	B	D
D	B	E
*E	B	{A,C}

(Minimized DFA Transition table)



Moore and mealy machine

- Both Moore and mealy machine are the special case of DFA
- Both act like the output producers rather than the string acceptors
- In Moore and mealy machine no need to define the final state.
- No concept of final state and dead state
- Mealy and Moore machine is equivalent in power

Moore machine

Moore's machine is defined as a machine in the theory of computation whose output values are determined only by its current state. It has also 6 tuples

$(Q, q0, \Sigma, \blacktriangle, \delta, \lambda)$

Q is a finite set of states $\{q0, q1, q2, q3\}$

q0 is the initial state $q0$

Σ is the input alphabet $\{0,1\}$

$\blacktriangle$ is the output alphabet $\{0,1\}$

δ is the transition function that maps $Q \times \Sigma \rightarrow Q$

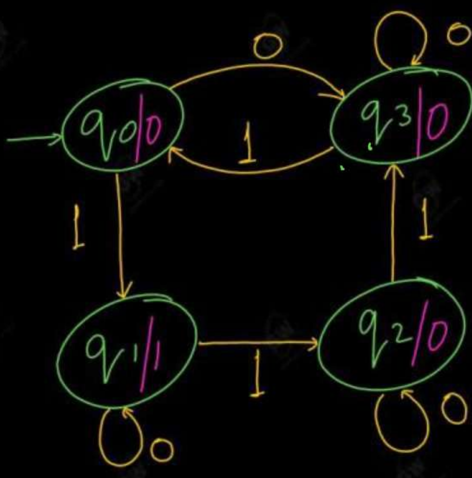
λ is the output function that maps $Q \rightarrow \blacktriangle$

NOTE

- In Moore machine for every state output is associated
- If length of input string is n, then length of output string will be n+1
- Moore machine response for empty string

Moore machine

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



Example

n 1110 - Output?

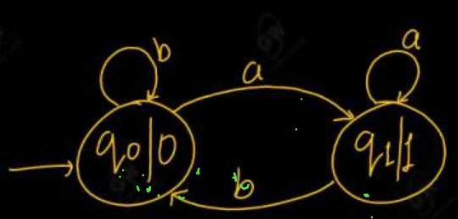
$q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_3$

0 1 0 0 0

01000 n+1

Present state	Next state a=0	a=1	Λ(output)
→q0	q3	q1	0
q1	q1	q2	1
q2	q2	q3	0
q3	q3	q0	0

Construct a Moore machine that takes all string of a and b as input and count number of a in the input string in term of 1 ∑={a,b}, ▲={0,1}



Present state	Next state a	b	Λ(output)
→q0	q1	q0	0
q1	q1	q0	1

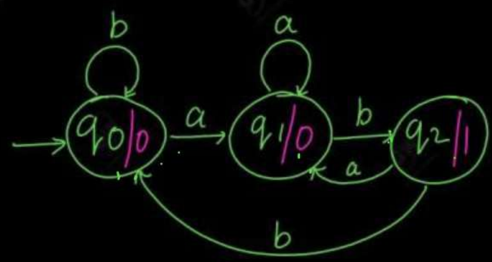
aaabbb

$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0$

0 1 1 1 0 0 0

0111000

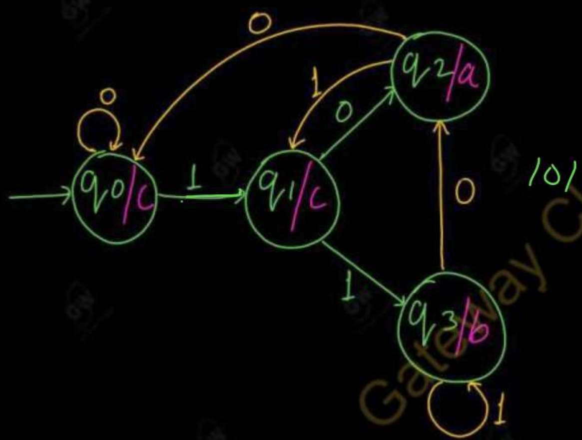
Construct a Moore machine that takes all string of a and b as input and count number of occurrence of sub-string 'ab' in term of 1 ∑={a,b}, ▲={0,1}



Present state	Next state a	b	Λ(output)
→q0	q1	q0	0
q1	q1	q2	0
q2	q1	q0	1

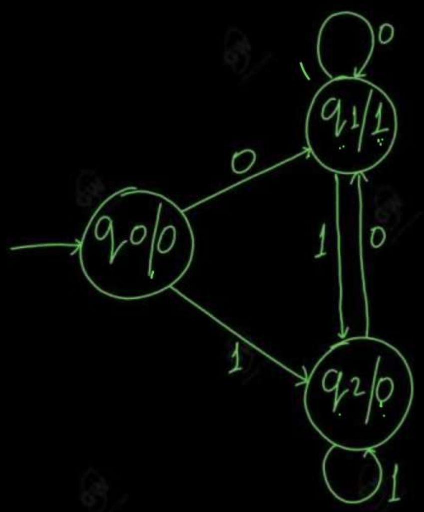
Construct a Moore machine that should give output a if the input string end with 10 ,b if the input string ends with 11 $\Sigma=\{0,1\}$, ▲={a,b,c}

10 ⟶ a
11 ⟶ b
other than this c



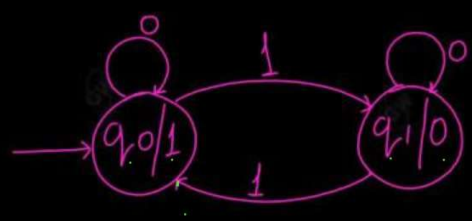
Present state	Next state 0	1	λ(output)
→q0	q0	q1	c
q1	q2	q3	c
q2	q0	q1	a
q3	q2	q3	b

Construct the Moore machine to generate the 1's complement of a given binary number $\Sigma = \{0, 1\}$



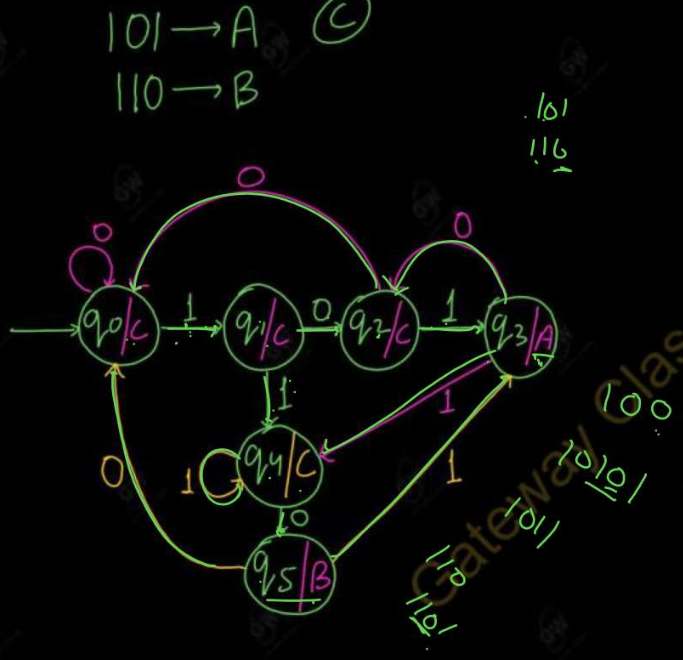
Present state	Next state 0	1	Λ(output)
→q0	q1	q2	0
q1	q1	q2	1
q2	q1	q2	0

Construct the Moore machine that determines whether an input string contain an even or odd number of 1 the machine generate 1 as a output if an even number of 1 are in the string and 0 otherwise



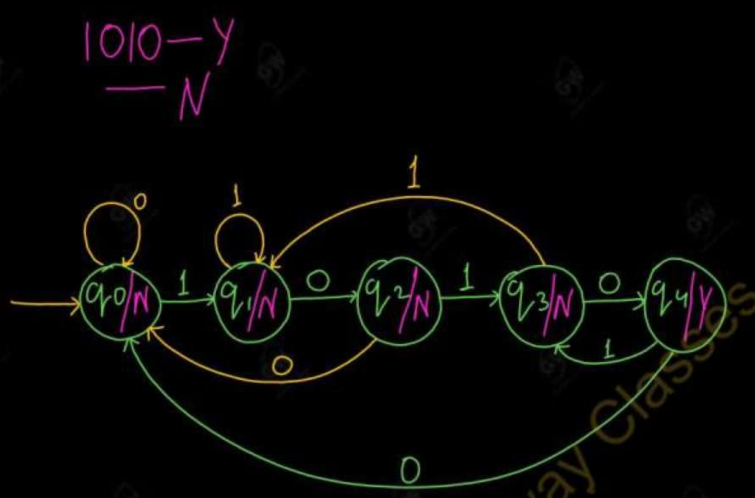
Present state	Next state 0	1	Λ(output)
→q0	q0	q1	1
q1	q1	q0	0

Design a Moore machine for a binary input sequence such that if it has a substring 101 then the machine output A, if the input has substring 110, it output B otherwise output c



Present state	Next state 0	1	Λ(output)
→q0	q0	q1	C
q1	q2	q4	C
q2	q0	q3	C
q3	q2	q4	A
q4	q5	q4	C
q5	q0	q3	B

Design a Moore machine with the input alphabet (0,1) an output alphabet{Y,N} which produces Y as output if input sequence contain 1010 as a substring otherwise ,it produces N as a output



Present state	Next state 0	1	Λ(output)
→q0	q0	q1	N
q1	q2	q1	N
q2	q0	q3	N
q3	q4	q1	N
q4	q0	q3	Y

Mealy machine

Mealy machines are also finite state machines with output value and their output depends on the present state and current input symbol. It can be defined as (Q, q0, ∑, O, δ, λ') where:

Q is a finite set of states.

q0 is the initial state. q_1

∑ is the input alphabet.

O is the output alphabet.

δ is the transition function which maps Q×∑ → Q.

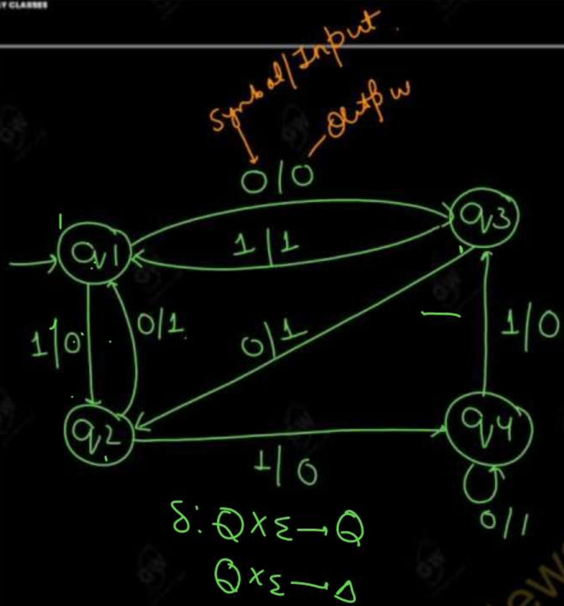
'λ' is the output function that maps Q×∑→ O.

λ

NOTE:

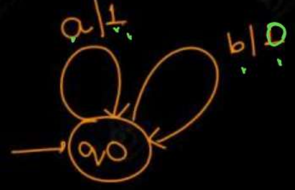
- If the length of input string is n, then length of output string will be n
- Mealy machine do not response for empty string

Mealy machine



	0		1	
	State	output	State	output
→q1	q3	0	q2	0
q2	q1	1	q4	0
q3	q2	1	q1	1
q4	q4	1	q3	0

Construct a Mealy machine that takes all string of a and b as input and count number of a in the input string in term of 1 $\Sigma=\{a,b\}$, $\blacktriangle=\{0,1\}$

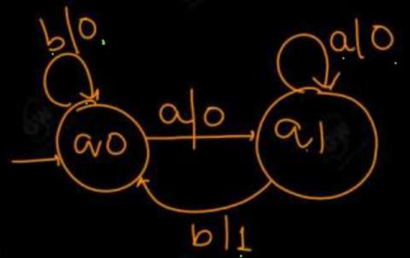


	a		b	
	State	output	State	output
q0	q0	1	q0	0

aab

$q0 \xrightarrow[\downarrow\,1]{a} q0 \xrightarrow[\downarrow\,1]{a} q0 \xrightarrow[\downarrow\,0]{b} q0$

Construct a Mealy machine that takes all string of a and b as input and count number of occurrence of sub-string 'ab' in term of 1 ∑={a,b}, ▲={0,1}



	a State	a output	b State	b output
->q0	q1	0	q0	0
q1	q1	0	q0	1

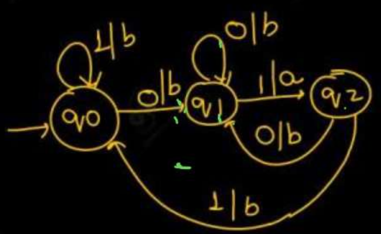
Construct the Mealy machine to generate the 1's complement of a given binary number

	0		1	
	State	output	State	output
->q0	q0	1	q0	0

Construct the Mealy machine that print a whenever the sequence 01 is encountered in any input binary string

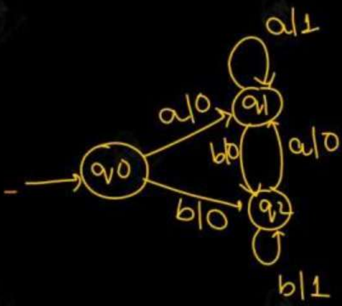
$\Sigma = \{0, 1\}$

Output = a | b



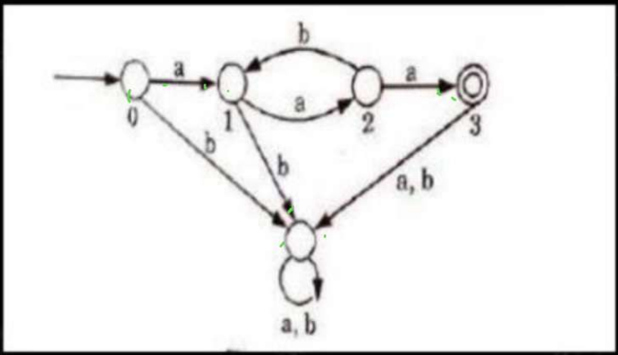
	0 State	output	1 State	output
->q0	q1	b	q0	b
q1	q1	b	q2	a
q2	q1	b	q0	b

Construct the Mealy machine that accept language consisting having alphabet{a,b} and the string should end with aa or bb this will counted by 1



	a State	output	b State	output
->q0	q1	0	q2	0
q1	q1	1	q2	0
q2	q1	0	q2	1

Consider the DFA given below fin out which language accepted by the DFA



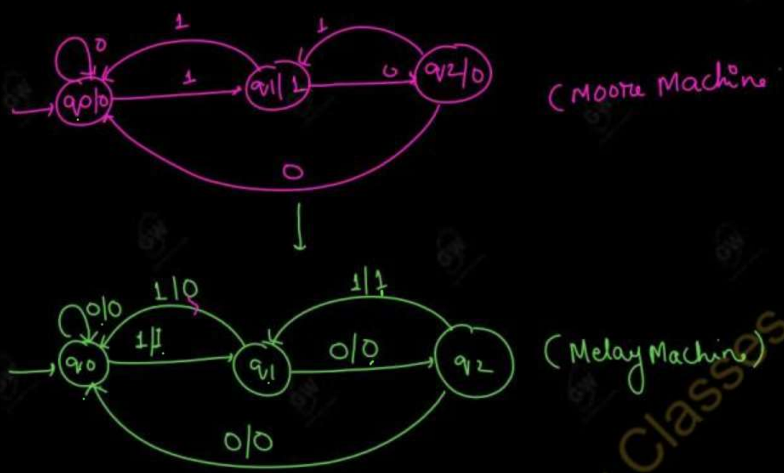
DFA that accept all string that start & end with a

L = {W ∈ {a,b}* | W accept all the string that start and end with a }

DIFFRENCE BETWEEN -

Aspect	Moore Machines	Mealy Machines
Output	Outputs depend only on the current state.	Outputs depend on the current state and input.
Number of States	Tends to require more states due to separate output behavior.	Might require fewer states as outputs are tied to transitions.
Response Time	Slower response to input changes as outputs update on state changes.	Faster response to input changes due to immediate output updates.
Complexity	Can be simpler due to separation of output behavior.	Can be more complex due to combined state-input cases.

DFA	NFA
Epsilon move is not allowed in DFA	Epsilon move is allowed in NFA
DFA allows only one move for single input alphabet.	There can be choice (more than one move) for single input alphabet.
δ: QxΣ -> Q i.e. next possible state belongs to Q.	δ: Qx(Σ U ε) -> 2^Q i.e. next possible state belongs to power set of Q
All DFA are NFA.	Not all NFA are DFA.
DFA requires more space.	NFA requires less space then DFA.
In DFA, the next possible state is distinctly set.	In NFA, each pair of state and input symbol can have many possible next states.
DFA is more difficult to construct.	NFA is easier to construct.
DFA cannot use Empty String transition.	NFA can use Empty String transition.



Present state	Next state 0	1	(output)
→q0	q0	q1	0
q1	q2	q0	1
q2	q0	q1	0

Moore machine transition table

Q1 Moore to mealy machine conversion

	0		1	
	State	**output**	**State**	**output**
->q0	q0	0	q1	1
q1	q2	0	q0	0
q2	q0	0	q1	1

Mealy machine transition table

Present state	**Next state** 0	**1**	(output)
→q0	q0	q1	0
q1	q2	q0	1
q2	q0	q1	0

Moore transition table

Q2 Moore to mealy machine conversion

	0		1	
	State	output	State	output
→q0	q1	0	q2	1
q1	q3	1	q2	1
q2	q2	1	q1	0
q3	q0	1	q3	1

Mealy machine transition table

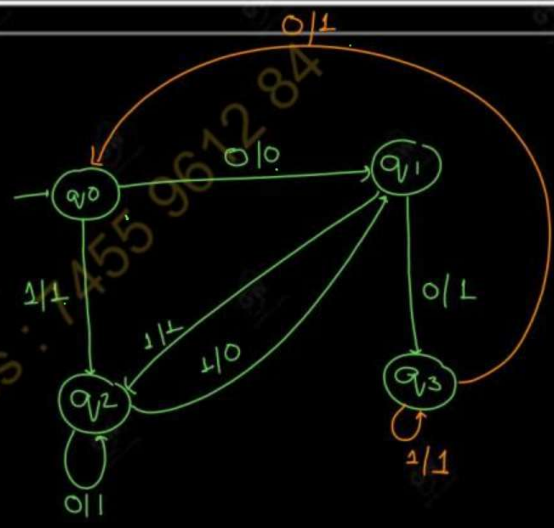
Present state	Next state 0	1	(output)
→q0	q1	q2	1
q1	q3	q2	0
q2	q2	q1	1
q3	q0	q3	1

Moore transition table

Q2 Moore to mealy machine conversion

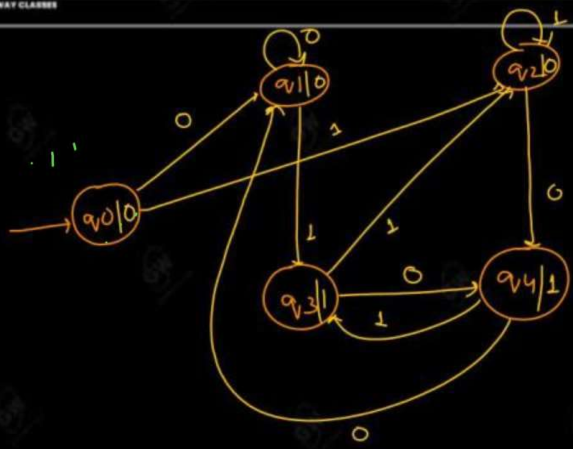
	0		1	
	State	output	State	output
→q0	q1	0	q2	1
q1	q3	1	q2	1
q2	q2	1	q1	0
q3	q0	1	q3	1

Mealy machine transition table



Mealy transition diagram

Q3Moore to mealy machine conversion



Present state	Next state 0	1	(output)
→q0	q1	q2	0
q1	q1	q3	0
q2	q4	q2	0
q3	q4	q2	1
q4	q1	q3	1

Moore machine transition table

Q3 Moore to mealy machine conversion

	0 State	0 output	1 State	1 output
q0	q1	0	q2	0
q1	q1	0	q3	1
q2	q4	1	q2	0
q3	q4	1	q2	0
q4	q1	0	q3	1

Mealy machine transition table

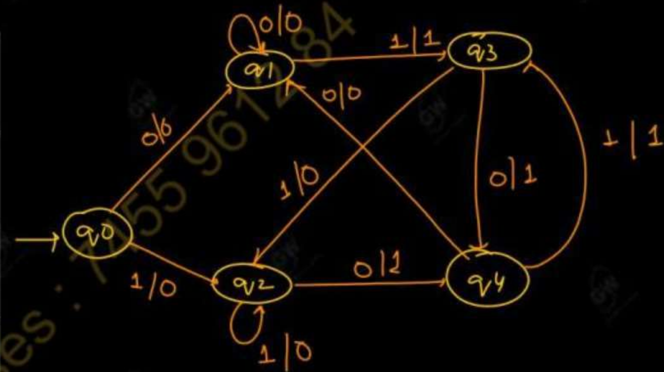
Present state	Next state 0	1	(output)
→q0	q1	q2	0
q1	q1	q3	0
q2	q4	q2	0
q3	q4	q2	1
q4	q1	q3	1

Moore transition table

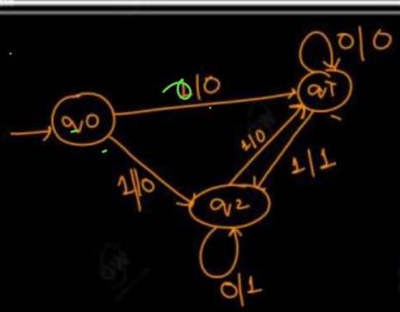
Q3 Moore to mealy machine conversion

	0		1	
	State	output	State	output
→q0	q1	0	q2	0
q1	q1	0	q3	1
q2	q4	1	q2	0
q3	q4	1	q2	0
q4	q1	0	q3	1

Mealy machine transition table



Q1Mealy to Moore conversion

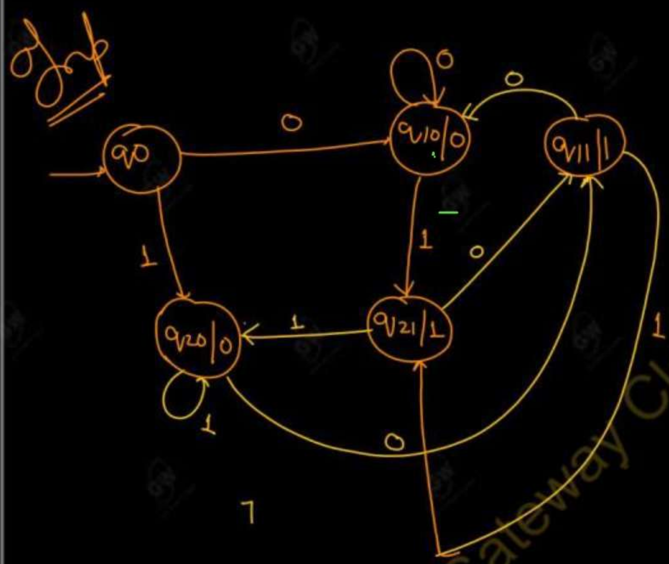


(Mlay Machine)

(Melay table)

	0 State	output	1 State	output
->q0	q1	0	q2	0
q1	q1	0	q2	1
q2	q2	1	q1	0

Q1Mealy to Moore conversion



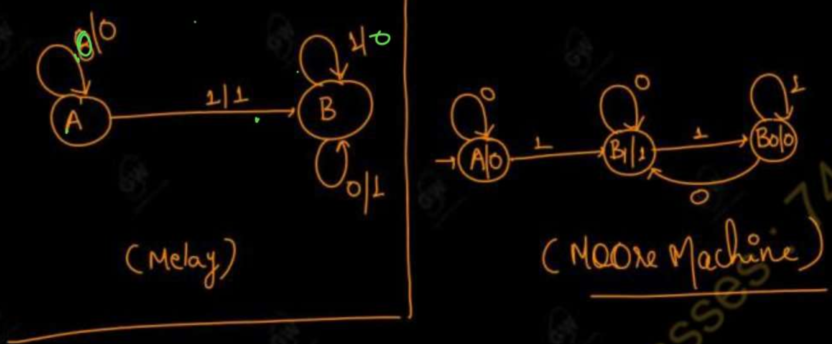
	0 State	output	1 State	output
->q0	q1	0	q2	0
q1	q1	0	q2	1
q2	q1	1	q2	0

Q1Mealy to Moore conversion

Present state	Next state 0	Next state 1	(output)
→q0	q10	q20	-
q10	q10	q21	0
q11	q10	q21	1
q20	q11	q20	0
q21	q11	q20	1

Moore transition table

Q1Mealy to Moore conversion



	0 State	output	1 State	output
->A	A	0	B	1
B	B	1	B	0

Q1Mealy to Moore conversion

Present state	Next state 0	1	(output)
→A	A	B1	0
B1	B1	B0	1
B0	B1	B0	0

Moore transition table