

UNIT 1

BASIC CONCEPT AND AUTOMATA THEORY

Todays' Target

- BASIC TERMINOLOGY

By PRAGYA RAJVANSI
B.Tech, M.Tech(C.S.E)

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.

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

$\Sigma = \{0, 1\}$ is an alphabet of binary digit

$\Sigma = \{0, 1, 2, 3, \dots, 9\}$ an alphabet of decimal digit

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

$\Sigma = \{A, B, C, D, \dots, Z\}$

Σ (SIGMA)

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

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

$\Sigma = \{a, b\}$ possible string over the alphabet $w = a$
 $a, b, ab, ba, aa, bb \dots$ $|w| = 1$

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

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 $\Sigma^* = \{\epsilon, 0, 1, 01, 10, 00, 11, 10101, \dots\}$.

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:

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

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

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

$$\Sigma^+ = \{0, 1, 00, \dots\}$$

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.

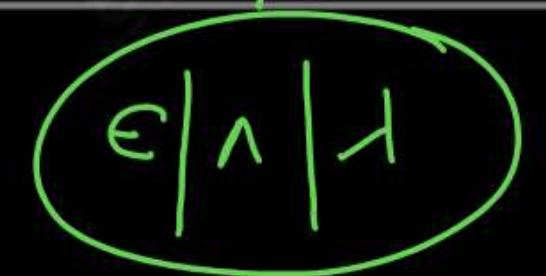
- For $\Sigma = \{0, 1\}$ we have set of strings as $\Sigma^* = \{\epsilon, 0, 1, 01, 10, 00, 11, 10101, \dots\}$.
- ✓ $L = \{ \text{set of all string end with 1 and start with 1} \}$
- $$L = \{1, 101, 1111, 100001\}$$
- If Σ is an alphabet, and $L \subseteq \Sigma^*$
- A language that can be formed over ' Σ ' can be Finite or Infinite.

SOME BASIC TERMINOLOGY

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

empty string

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



Set of all possible string of length p

$\Sigma^0 = \{\epsilon\}$ (LENGTH OF STRING MUST BE ZERO/NULL STRING/EMPTY STRING)

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

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

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

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

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

➤ **Representation**

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

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

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

➤ **Example** – If $\Sigma = \{a, b\}$, $\Sigma^+ = \{a, b, aa, ab, ba, bb, \dots\}$

➤ **Definition** – The set Σ^+ is the infinite set of all possible strings of all possible lengths over Σ excluding λ .

Kleene closure & Kleene positive

Q1	What is the value of following over <u>$\Sigma = \{a, b\}$</u> Σ^0 Σ^1 Σ^2 Σ^3
Q2	Differentiate between L^* and L^+

BASIC CONCEPT AND AUTOMATA THEORY

TODAY's TARGET

- Finite Automata and Types of Automata.
- Design some DFA
- Application and limitation of finite automata
- PYQ
- DPP

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

- 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

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

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

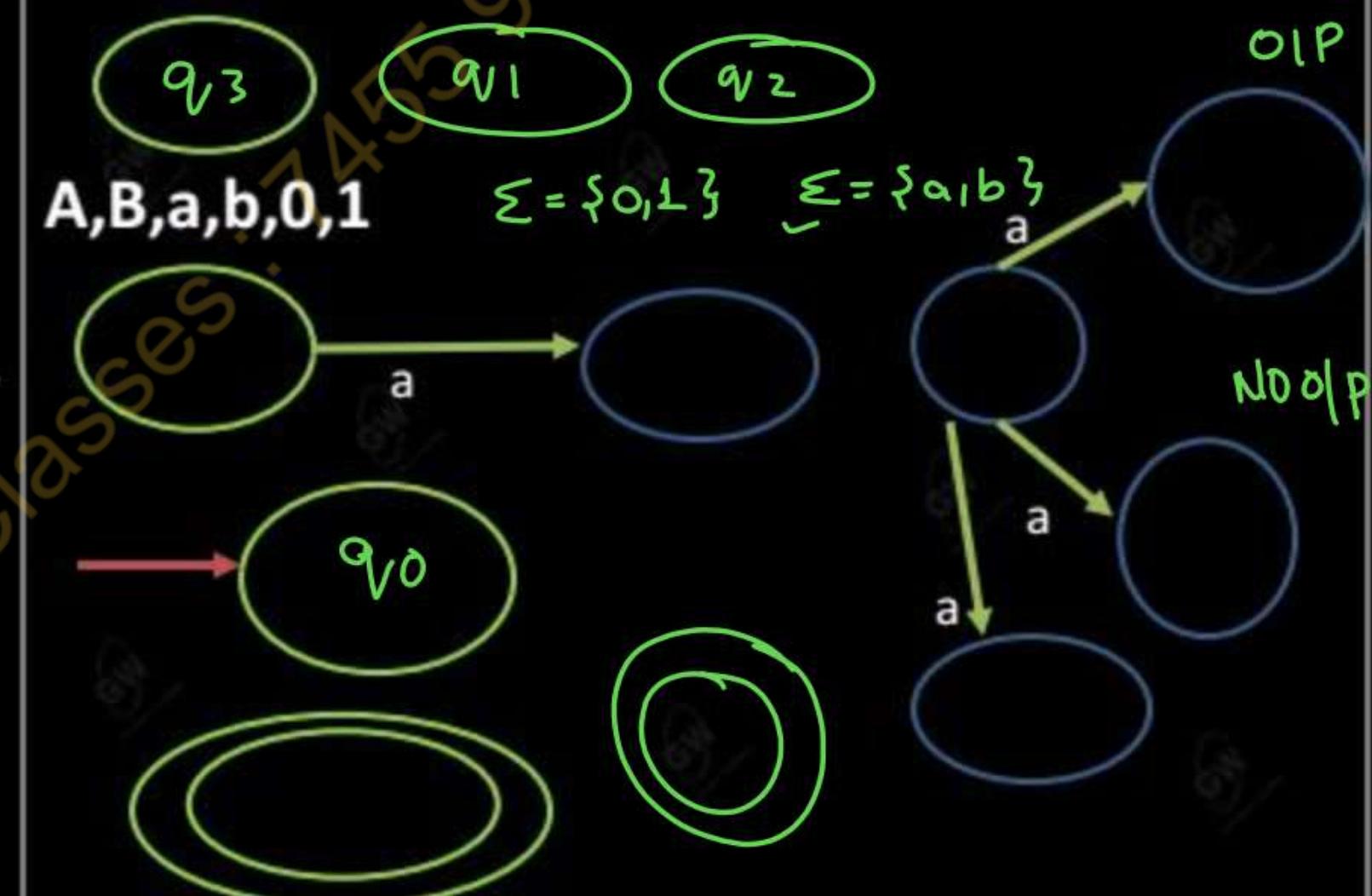
Σ : Finite set called alphabets.

δ : is the transition function.

$q_0 \in Q$ is the start or initial state.

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

Symbol Meaning: Σ (sigma), δ (delta)



✓ DIFFERENT TYPES OF FINITE AUTOMATA

Finite Automata without output

- Deterministic Finite Automata (DFA).
- Non-Deterministic Finite Automata (NFA or NDFA).
- Non-Deterministic Finite Automata with epsilon moves (e-NFA or e-NDFA).

Finite Automata with Output

- Moore machine.
- Mealy machine

Finite Automata Representation

- Graphical (Transition diagram)
- Tabular (Transition table)
- Mathematical (Transition function)

DETERMINISTIC 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 $(Q, \Sigma, \delta, q_0, F)$

- Q is a finite set of states. $\{q_0, q_1, q_3, q_4\}$
- Σ 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 ($F \subseteq Q$)

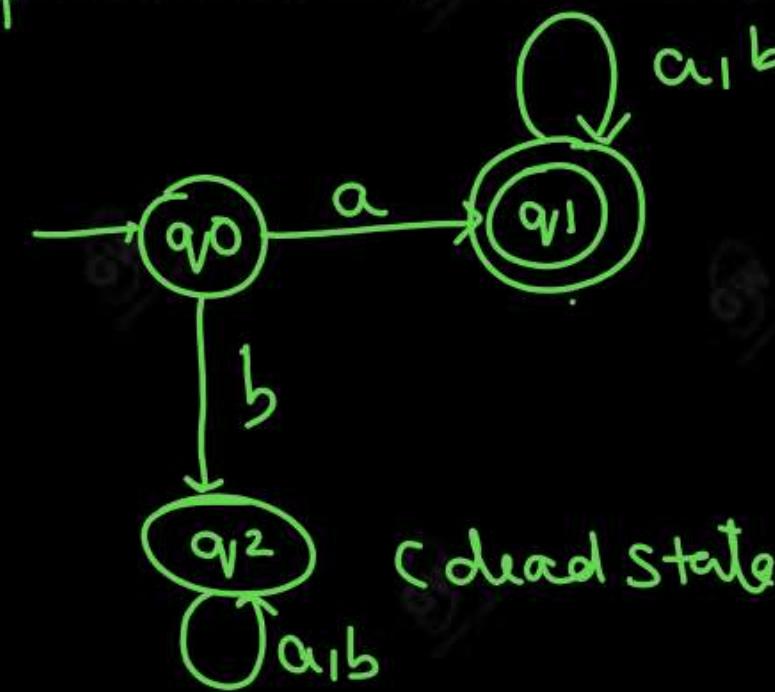
NOTE:

- *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 DIAGRAM

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


q_1 - final state
 $*q_1$ | q_1

Wrong String

x ba

x babaa

bababb

baaa

TRANSITION TABLE

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

{Q, Σ , δ , q_0 , F}

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

δ transition function is defined by the transition table

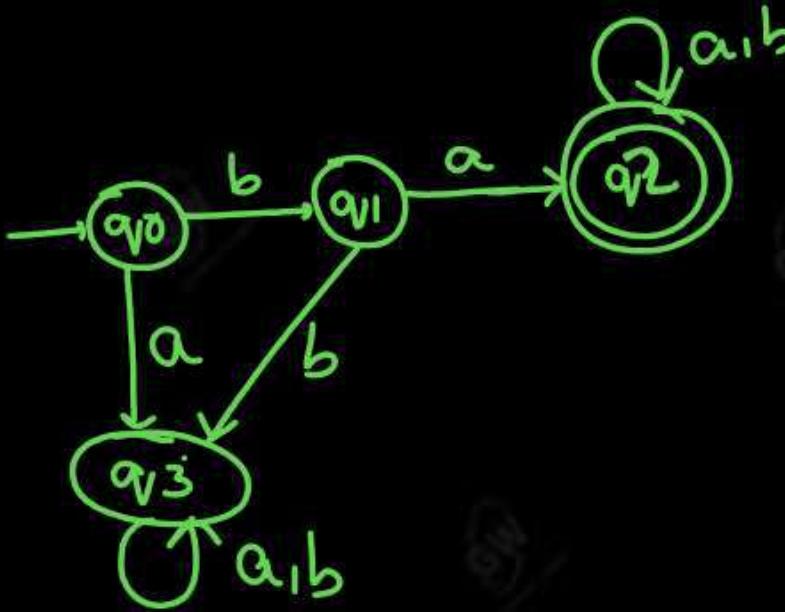
Note:

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

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

TRANSITION DIAGRAM

$L=\{ba, baaaaa, babb, babababab \dots\}$



Wrong strings:
bca
bbb
abah

TRANSITION TABLE

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

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

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

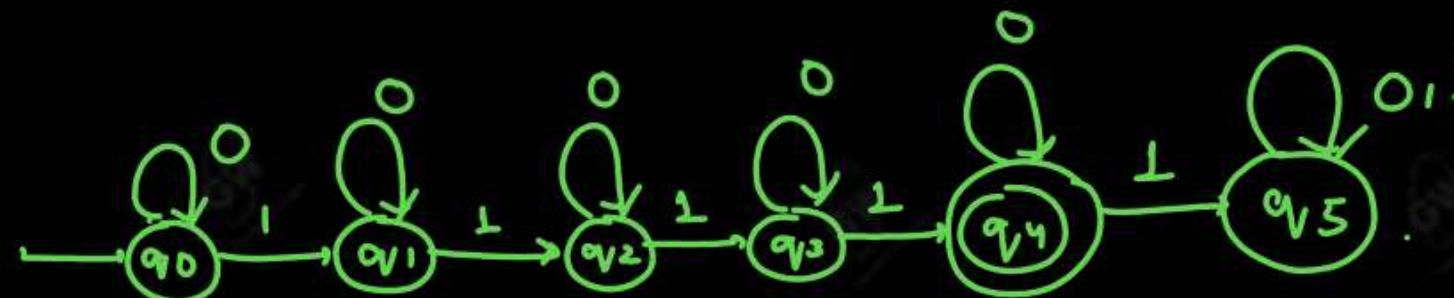
Note

Minimum state = minimum string length + 2

Design the finite automata which accept set of string containing exactly four 1's in every string
 $\Sigma = \{0, 1\}$

TRANSITION DIAGRAM

$$L = \{ \underline{1111}, 001111, 01010101, 110011, 11110000, 111010 \}$$



Wrong string

001101011

1111010

1010101110

TRANSITION TABLE

	0	1
$\rightarrow q_0$	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

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

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

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 $\underline{A^*}$
- 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 $A^* =$

$\{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$,

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 with the 10.

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

BASIC CONCEPT AND AUTOMATA THEORY

TODAY's TARGET

- DFA does/does not end with a string w.
- DFA having odd number of 1's
- DFA having even number of 1s'
- DFA in which string length exactly/ at least/at most 5
- DPP

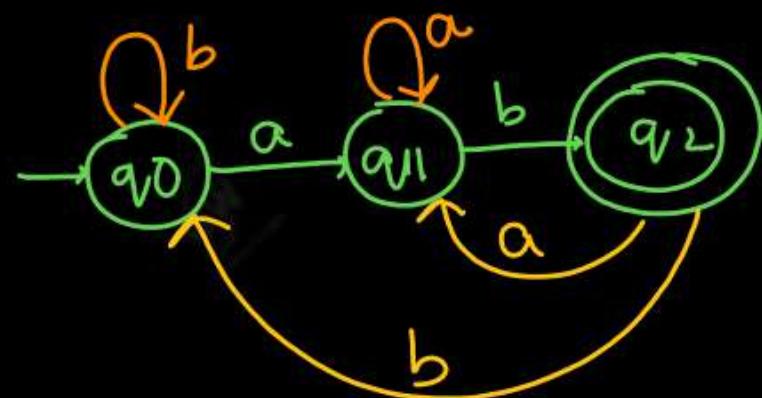
By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

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

TRANSITION DAIGRAM

$L = \{ab, bab, aaabbbaab, aaaaab, bbbbabaaab, \dots\}$



DFA

Wrong string

- $\times ababaaa$
- $\times bababb$
- $\times Abababbbb$

$w = ab$
 $|w| = 2$
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_1	q_0

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

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

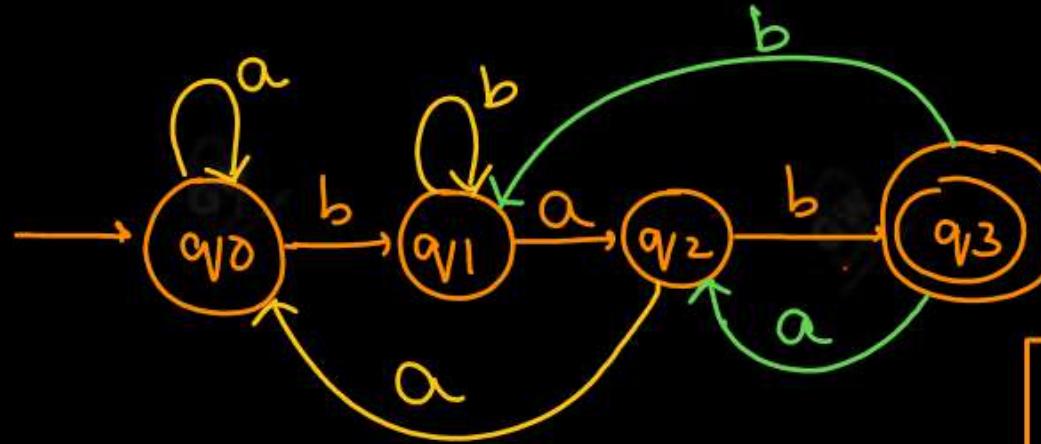
Note no dead state

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

$L = \{bab, aaaabab, aaabbabbab, bbabaabab, \dots\}$



Min length string = bab (w)

$$|w| \geq 3$$

$$\text{Minimum Status} = 3 + 1 = 4$$

Wrong string

- aaabaaa
- aababaa
- aaabbabbaba
- bbbbaba

TRANSITION TABLE

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

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

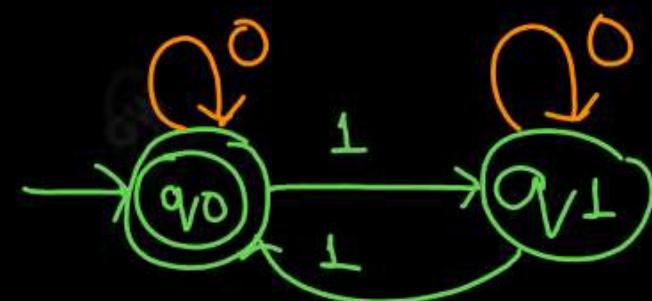
$\{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_3\}\}$

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

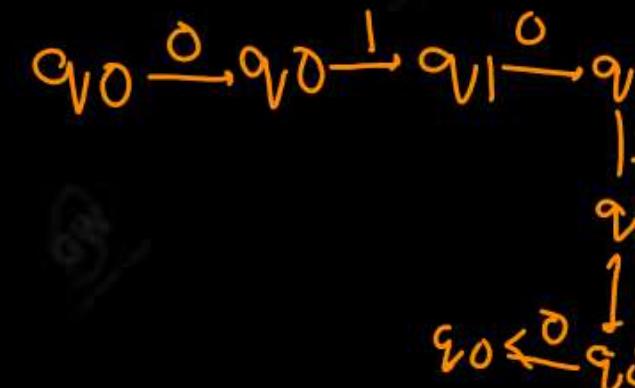
TRANSITION DIAGRAM

$L=\{0,00000,11,0101,110011,0101011, 1100110011 \dots\}$

\rightarrow



NOTE 010100



TRANSITION TABLE

	0	1
q_0	q_0	q_1
q_1	q_1	q_0

Wrong string

- ✓ 111000
- ✓ 000|010|
- ✓ ||||| | 000

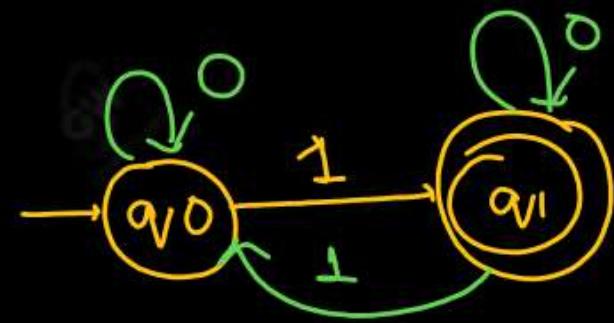
$\{Q, \Sigma, \delta, q_0, F\}$

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

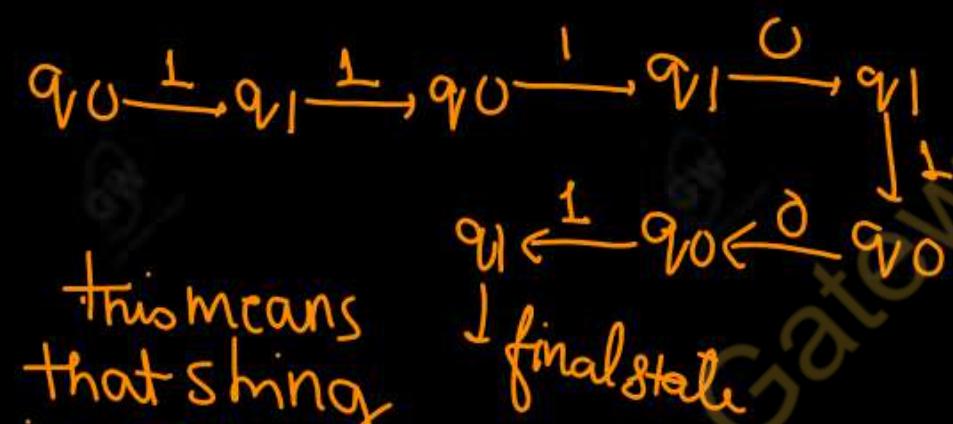
Design a DFA/MDFA $\Sigma=\{0,1\}$ such that accept the string which contain Odd number of 1s

TRANSITION DIAGRAM

$L=\{1, 10, 01, 1110, 10000, 1110000, 00001111, 000110100, 001110, \dots\}$



NOTE → 1110101



This means
that string
is accepted

TRANSITION TABLE

	0	1
q0	q0	q1
q1	* q1	q0

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

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

Non accepting string

111100 ✓

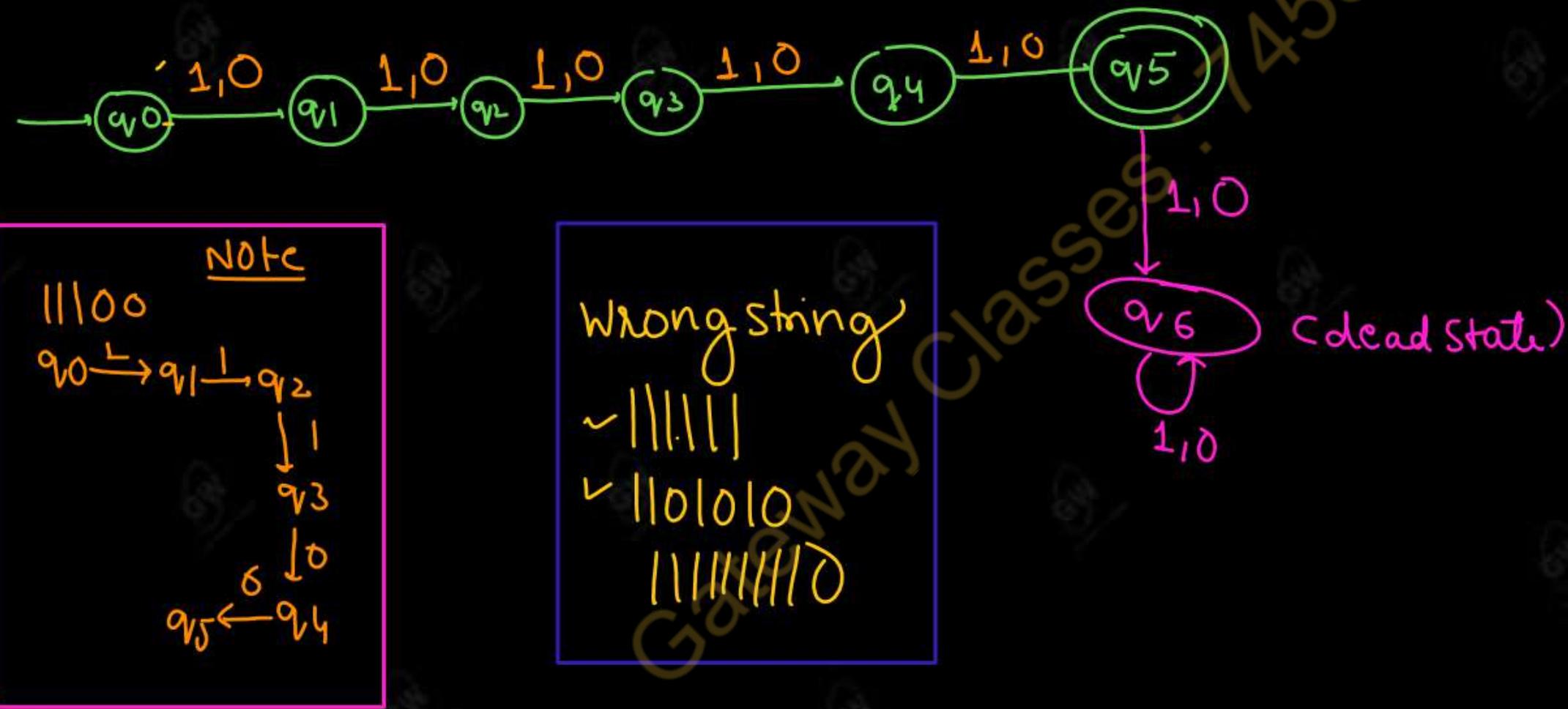
0001111

1010101

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

TRANSITION DAIGRAM

$L = \{00000, 11111, 10101, 01011, 00011, 11000, 10101, \dots\}$



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_6	q_6
q_6	q_6	q_6

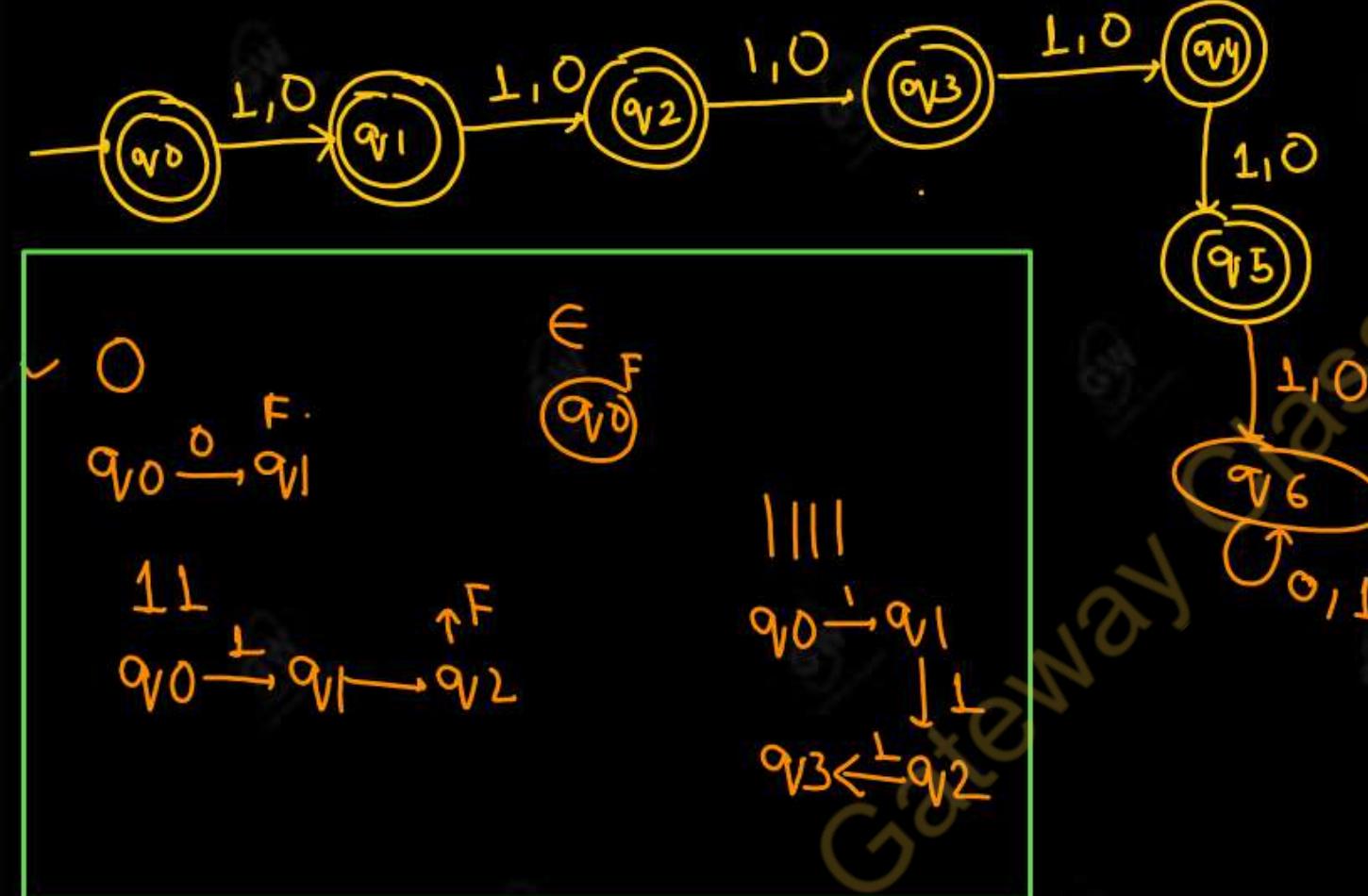
$\{Q, \Sigma, \delta, q_0, F\}$

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

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

TRANSITION DAIGRAM

$$L = \{\epsilon, 1, 0, 11, 01, 10, 111, 0111, 01010, 10101, \dots\}$$



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_6	q_6
q_6	q_6	q_6

Wrong string

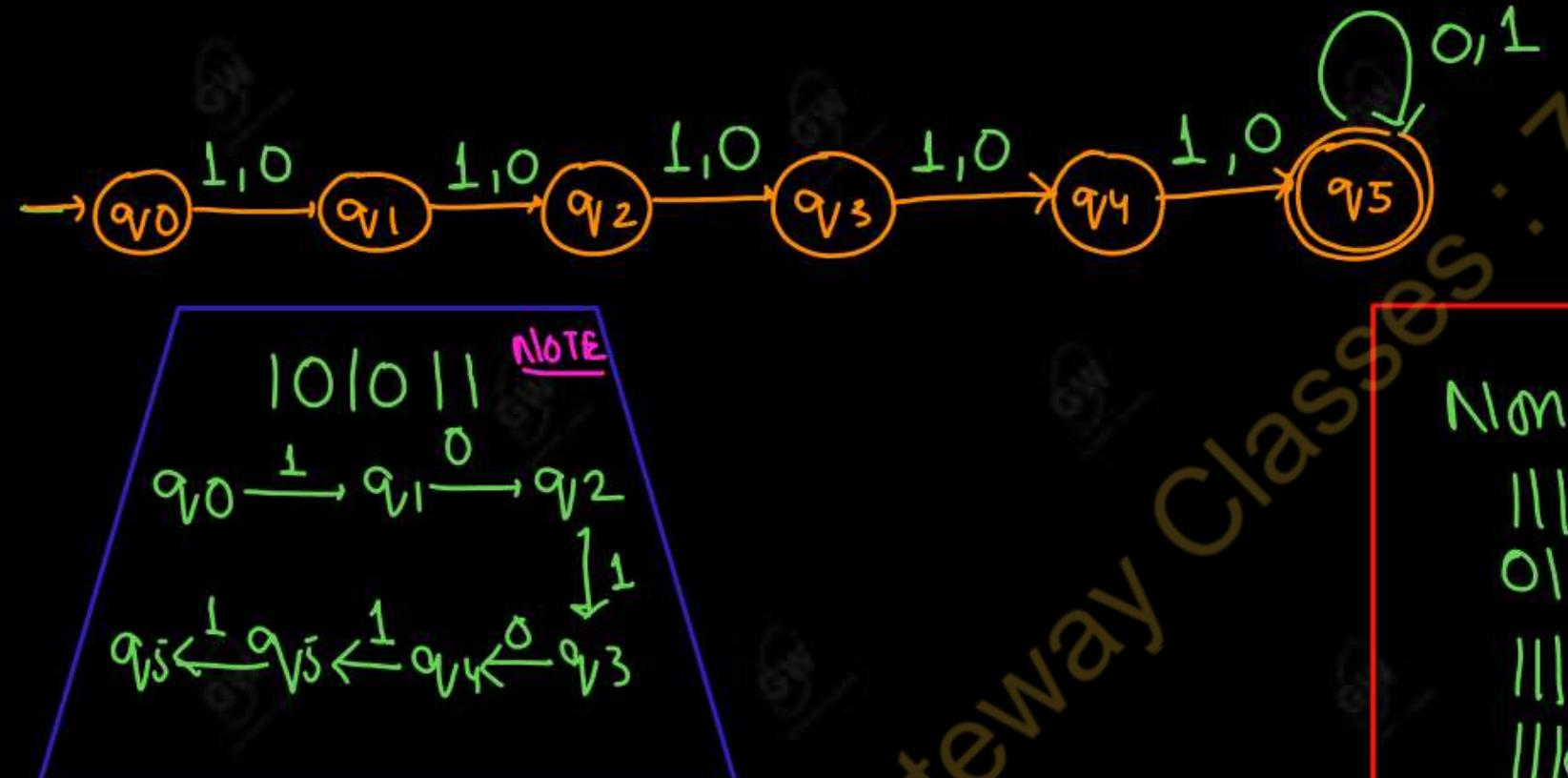
111111
0000000
101010101

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

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

TRANSITION DAIGRAM

$L = \{00000, 11111, 11111111, 1010100011, 010111, 0001000, 1100000, 101010 \dots\}$



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

Non-acceptable string

|||
010
|||
|||0

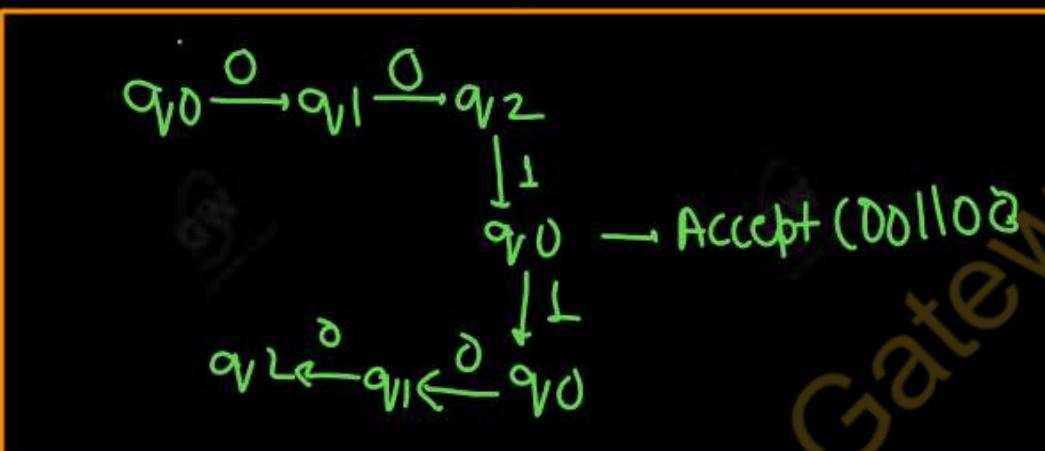
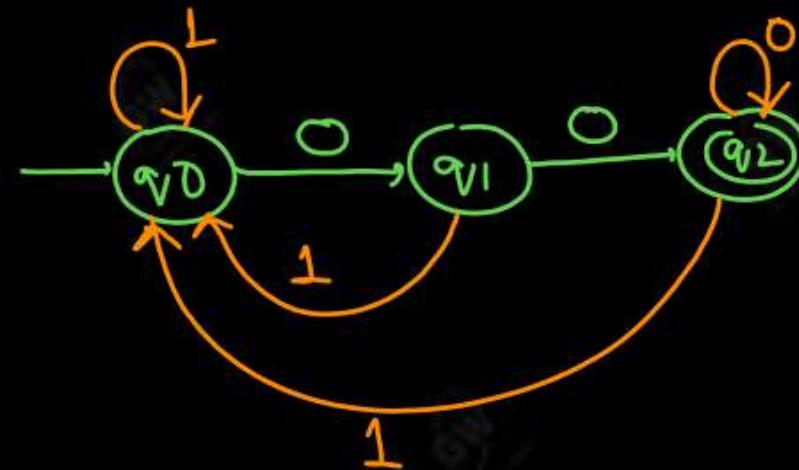
$\{Q, \Sigma, \delta, q_o, F\}$

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

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

TRANSITION DAIGRAM

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



TRANSITION TABLE

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

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

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

Non accepting string.

1111

1110

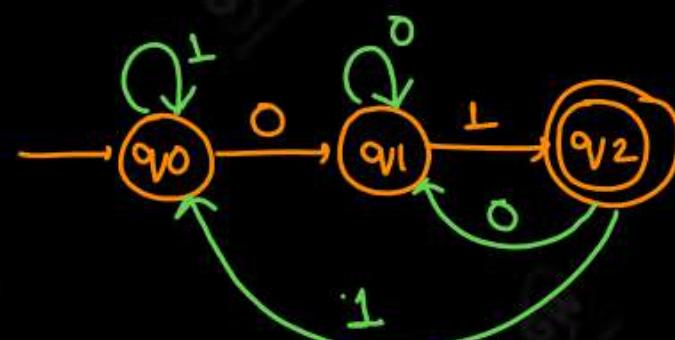
0010

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

TRANSITION DIAGRAM

$$L = \{ \epsilon, 0, 1, 1111, 000000, 0101111, \dots \}$$

① end with 01



→ Convert final state to non-final

→ Convert all non-final State to final

DFA that will accept $(0^+ 1)^*$



wrong string

- ~ 000001
- ~ 111101
- 11100001
- 01

TRANSITION TABLE

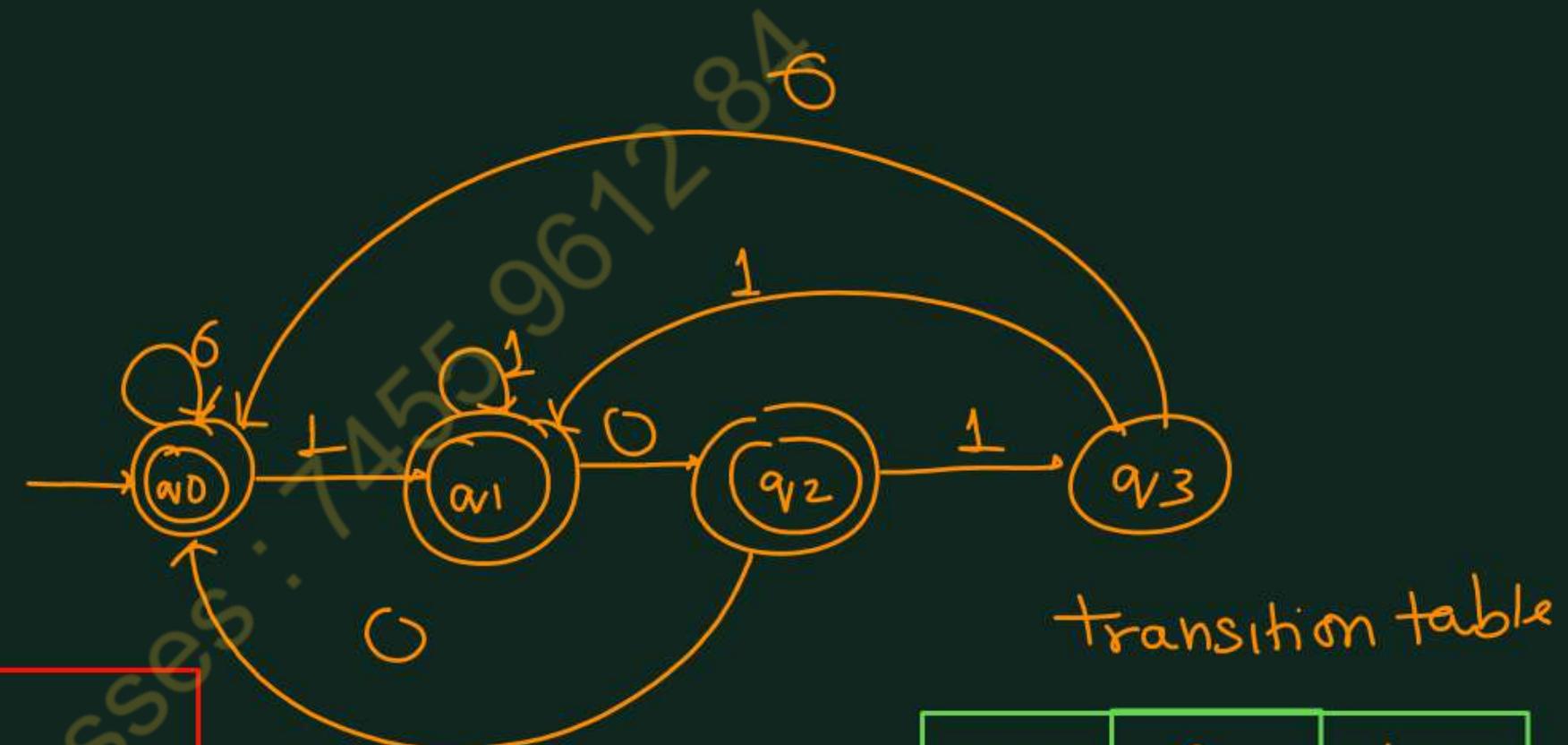
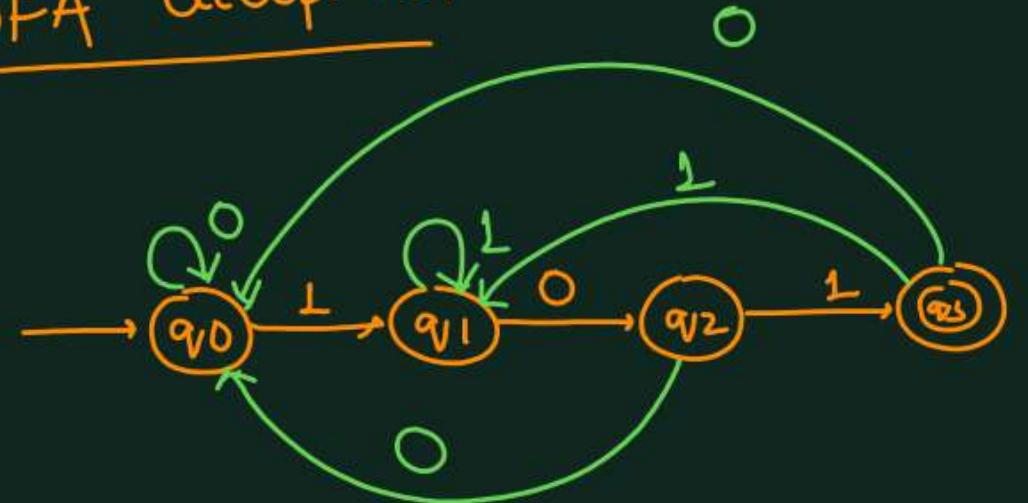
	0	1
$\rightarrow *q_0$	q_1	q_0
$*q_1$	q_1	q_2
q_2 (Non-final state)	q_1	q_0

$\sim \epsilon$ (Right String)
 q_1
 00
 111
 11
 10
 111
 010

$\{Q, \Sigma, \delta, q_0, F\}$
 $\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0,$
 $\{q_0, q_1\}\}$

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

DFA Accept 101



transition table

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

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

Q1	Construct a DFA that accept all the string of 1's and 0's where length of string is (1) at most 3 (2) At least 3 (3) Exactly 3
Q2	Design a DFA/MDFA/FA $\Sigma=\{a, b\}$ such that every string accepted (1) must end with a string $w=bb$ <input checked="" type="checkbox"/> (2) must end with a string $W=abb$ <input checked="" type="checkbox"/> (3) does not end with a string $W=abba$ <input checked="" type="checkbox"/> (4) does not end with a string $W=aba$
Q3	Design a DFA/MDFA/FA $\Sigma=\{0, 1\}$ such that every string accepted (1) does not end with a string $w=11$ (2) does not end with a string $w=01$ (3) does end with a string $w=0101$
Q4	Construct a DFA that accept all the string of a and b where length of string is (1) at most 4 (2) At least 4 (3) Exactly 4
Q5	Design a DFA/MDFA $\Sigma=\{0,1\}$ such that accept the string (1) which contain even number of 0s (2) which contain odd number of 0s

BASIC CONCEPT AND AUTOMATA THEORY

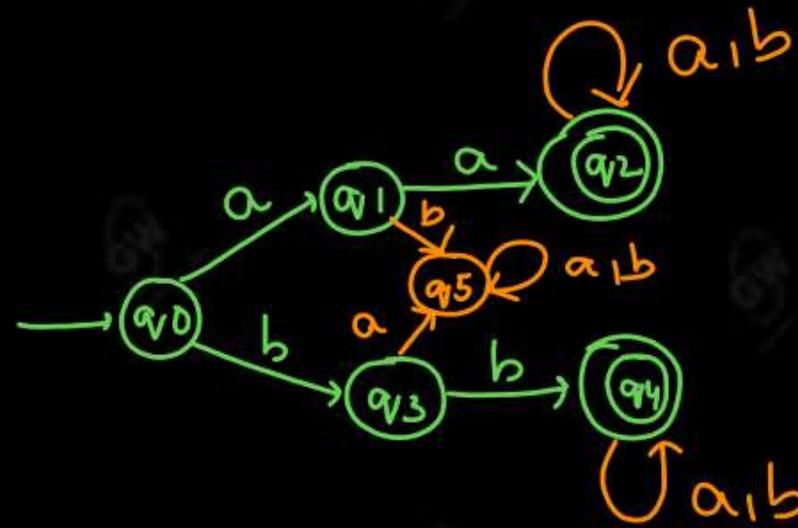
TODAY's TARGET

- DFA that start/end with aa or bb
- DFA that contain the substring aa or bb
- DFA of $a^n | n \geq 0$ / $a^n b^m | n, m \geq 0$ / $a^n b^m | n, m \geq 1$
- DFA with modulus
- DFA in which a is divisible by 2(even) and b is divisible by 3
- DPP

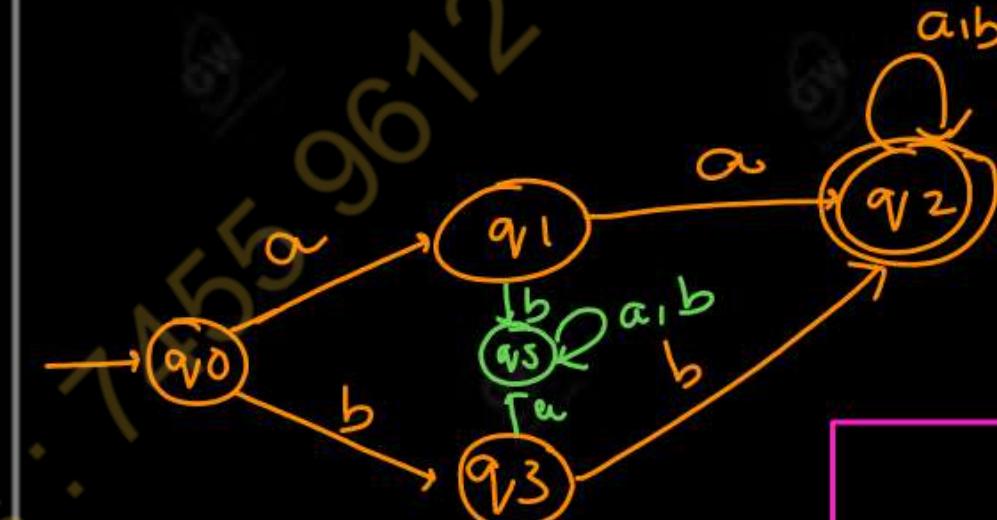
Design a DFA/MDFA/FA $\Sigma=\{a, b\}$ such that every string accepted that start with either aa or bb

TRANSITION DIAGRAM

$L=\{aa, bb, aabab, bbbaba, aabbba.....\}$



Minimize



Wrong string
✓ abab
✗ baba
✗ b

(Right String)
aab
aababc
bbaa

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

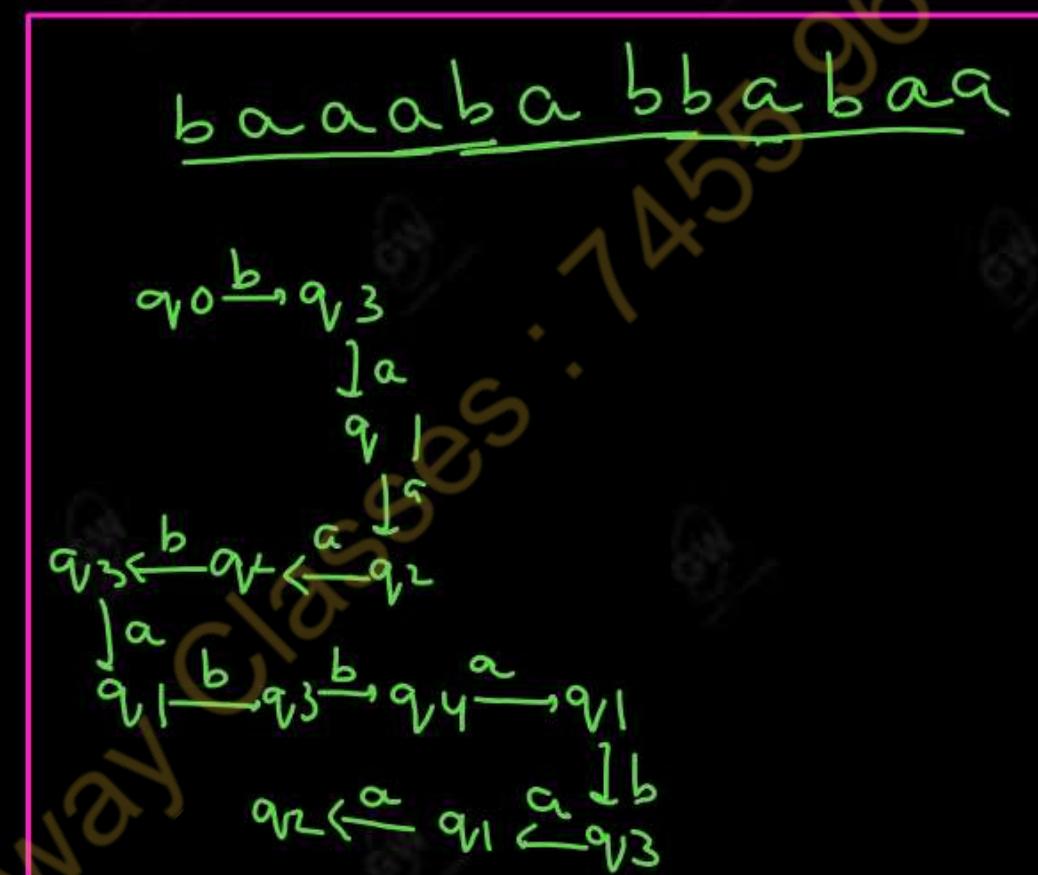
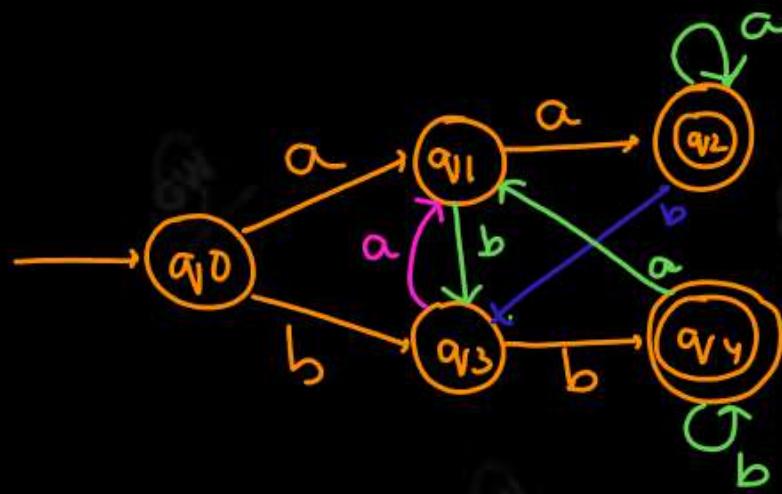
TRANSITION TABLE

	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

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

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

TRANSITION DIAGRAM

 $L=\{aa, bb, baaababbabaa, ababbaaa, babababb, ababbabb, abababbbbaa, \dots\}$ 

Wrong String

 $\rightarrow abab$
 $\rightarrow bubba$

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

TRANSITION TABLE

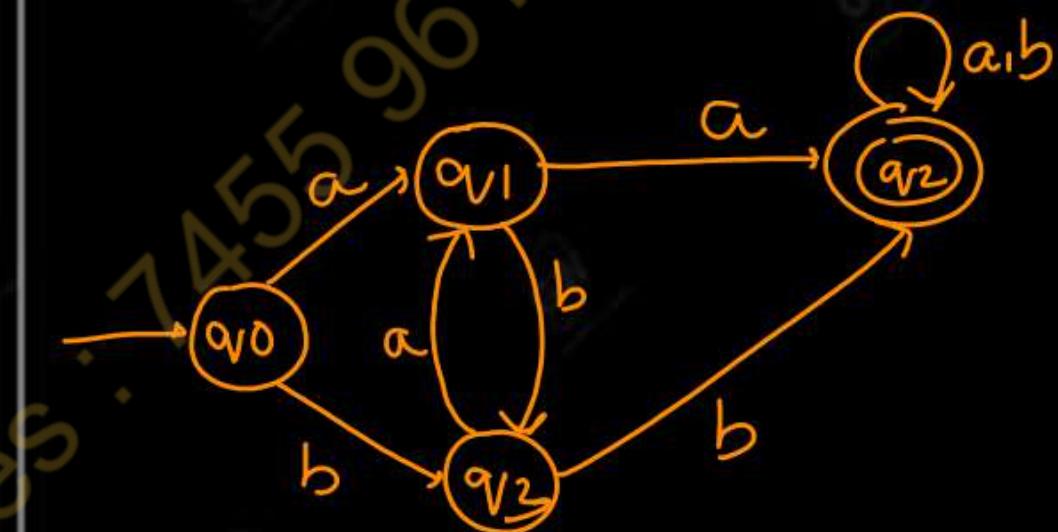
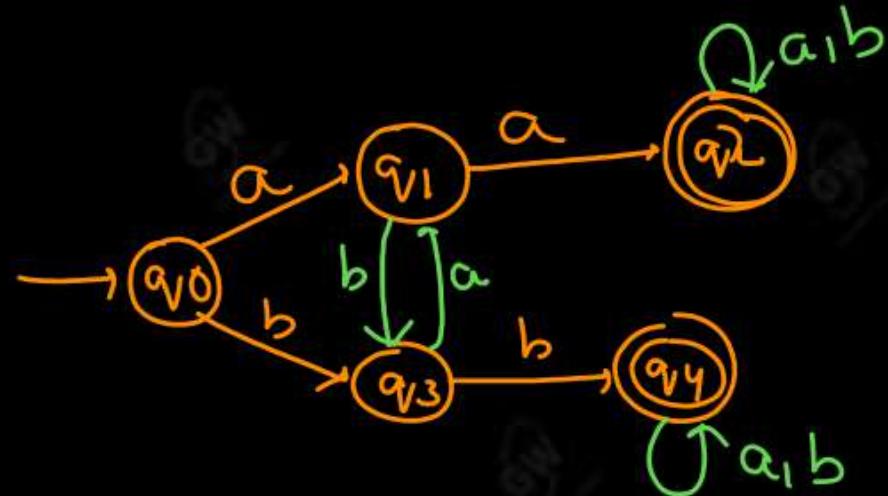
	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_3
$*q_2$	q_2	q_3
q_3	q_1	q_4
$*q_4$	q_1	q_5

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

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

TRANSITION DAIGRAM

$L=\{aa, bb, aababab, bbbbabab, bababaabbab, bbbabaaabb, abababbbbabab\}$



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

TRANSITION TABLE

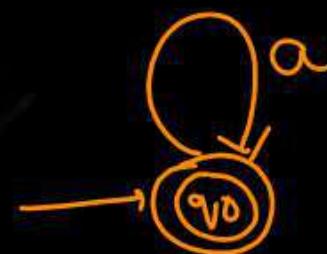
	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_3
$*q_2$	q_2	q_2
q_3	q_1	q_2

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

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

TRANSITION DAIGRAM

$L = \{\epsilon, a, aa, aaaa, aaaaa, \dots\}$



TRANSITION TABLE

	a
$\rightarrow *q_0$	q_0

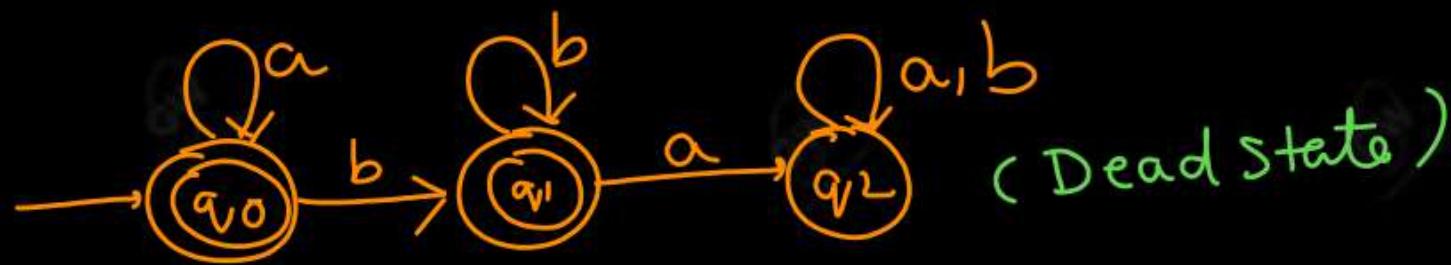
$\{Q, \Sigma, \delta, q_0, F\}$

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

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

TRANSITION DIAGRAM

$L=\{\epsilon, a, aaaa, b, bbbb, aabbb \dots\}$



<u>aba</u>	ϵ (Acceptable String)
abaaaa	<u>aaaaaa</u>
babbba	<u>b</u>
(Non-acceptable String)	bbbbbb
	aab
	aabb

TRANSITION TABLE

	a	b
ϵ	q_0	q_1
a	q_0	q_1
b	q_2	q_2
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_2

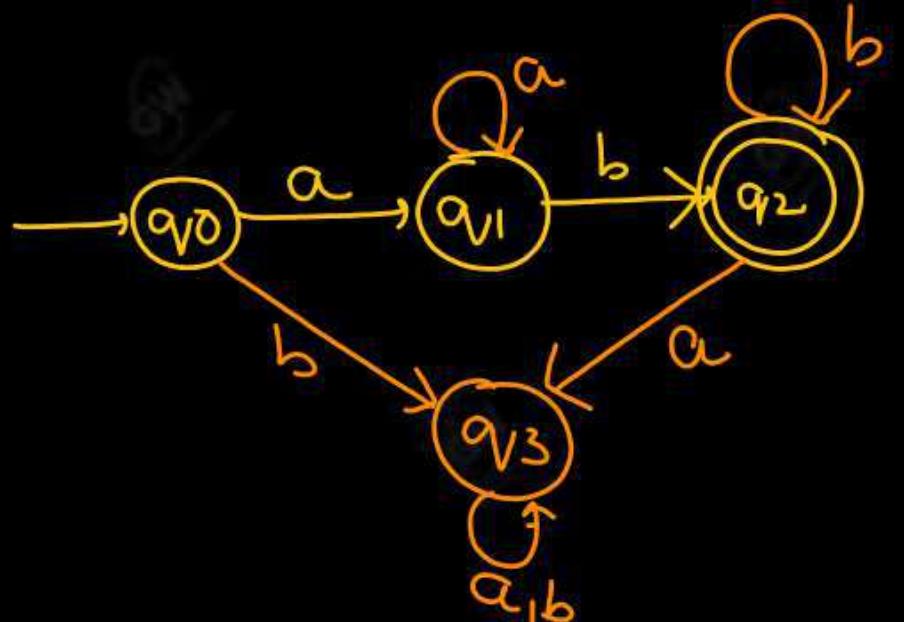
$\{Q, \Sigma, \delta, q_0, F\}$

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

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

TRANSITION DIAGRAM

$L=\{ab, aaabbffff, abbbb, aaaaab\}$



Wrong strings

- ✓ aabsa
- ✓ 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

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

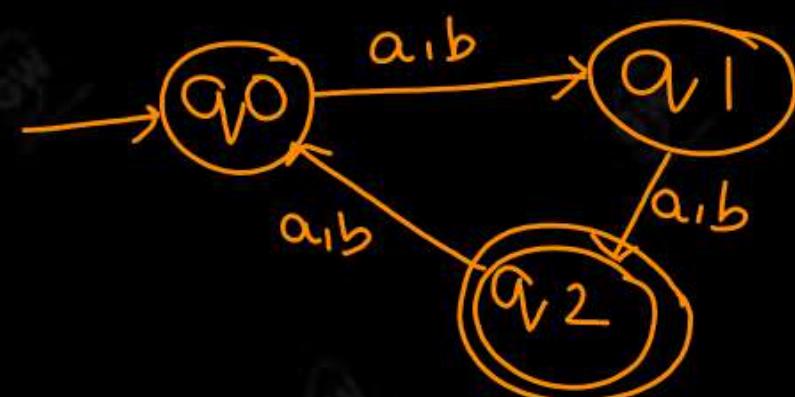
$\{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_2\}\}$

Construct a DFA over { a, b } such that every string accepted must $|w| \equiv 2 \pmod{3}$ / { w : | w | mod 3 > 1 }

TRANSITION DIAGRAM

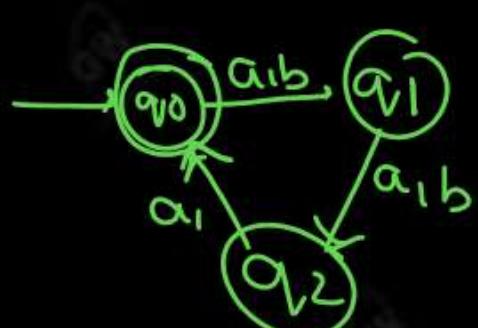
$L = \{ aa, bb, aaabb, aabbb, aabbaabb, bbbbaaaa, \dots \}$

$$|w| \equiv 2 \pmod{3}$$



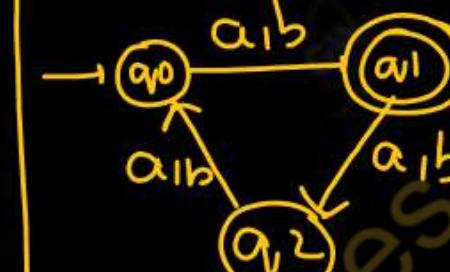
(Multiple of 3), divisible by 3

$$|w| \equiv 0 \pmod{3}$$



	a	b
*q0	q1	q1
*q1	q2	q2
q2	q0	q0
q0	q0	q0

$$|w| \equiv 1 \pmod{3}$$



	a	b
*q0	q1	q1
*q1	q2	q2
q2	q0	q0
q0	q0	q0

TRANSITION TABLE $|w| \equiv 2 \pmod{3}$

	a	b
*q0	q1	q1
q1	q2	q2
*q2	q0	q0
q0	q0	q0

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

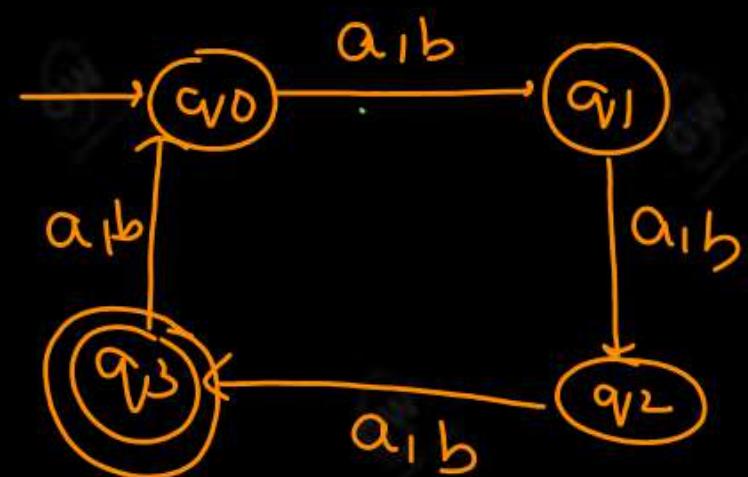
$\{ \{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, q_2 \}$

NOTE

$w = aa \quad |w| = 2 \text{ (length)}$

TRANSITION DAIGRAM

$L = \{ \text{aaa, bbb, abababa, bababab, ababababbbb....} \}$
 $|w|: 3 \quad 3 \quad 7 \quad 7 \quad 11$

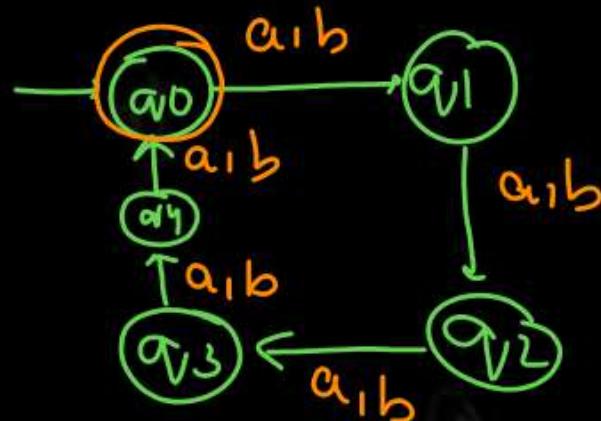
NOTE

	Initial State	Final state
$0 \pmod{4}$ / Multiple of 4	q_0	q_0
$1 \pmod{4}$	q_0	q_1
$2 \pmod{4}$	q_0	q_2

TRANSITION TABLE

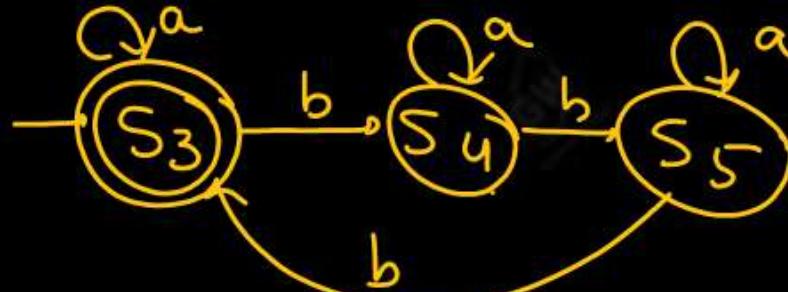
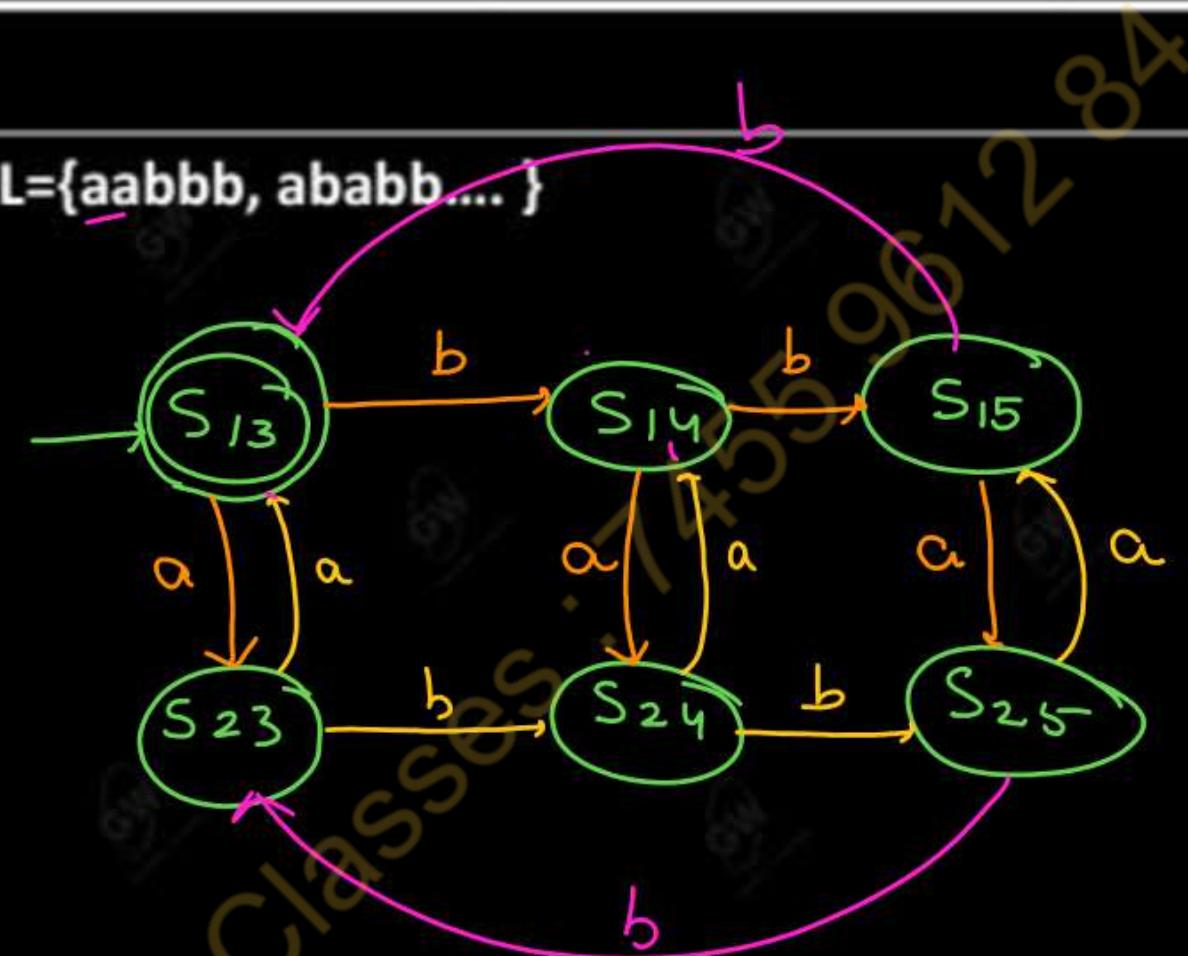
	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
$*q_3$	q_0	q_0

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

TRANSITION DAIGRAM $L = \{\epsilon, aaaaa, bbbbb, ababababab, bababababa, \dots\}$ **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

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

*a is multiple of 2**a is divisible 2 | $|w|_a \equiv 0 \pmod{2}$* *a is even**b is divisible | Multiple of 3 $|w|_b = 0 \pmod{3}$*  $L = \{aabbbb, ababb\ldots\}$  $|w|_a \pmod{2} = 0$ and
 $|w|_b \pmod{3} = 0$ 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

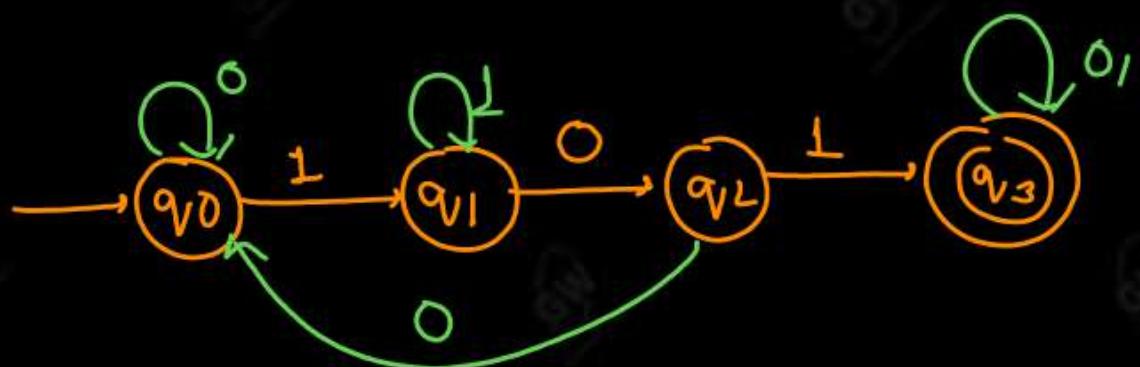
	a	b
$\rightarrow^* S_{13}$	S_{23}	S_{14}
S_{14}	S_{24}	S_{15}
S_{15}	S_{25}	S_{13}
S_{23}	S_{13}	S_{24}
S_{24}	S_{14}	S_{25}
S_{25}	S_{15}	S_{23}

$$\{Q, \Sigma, \delta, q_0, F\} = \{\{S_{13}, S_{14}, S_{15}, S_{23}, S_{24}, S_{25}\}, \{a, b\}, \delta, S_{13}, S_{13}\}$$

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

DFA accept 101 as substring

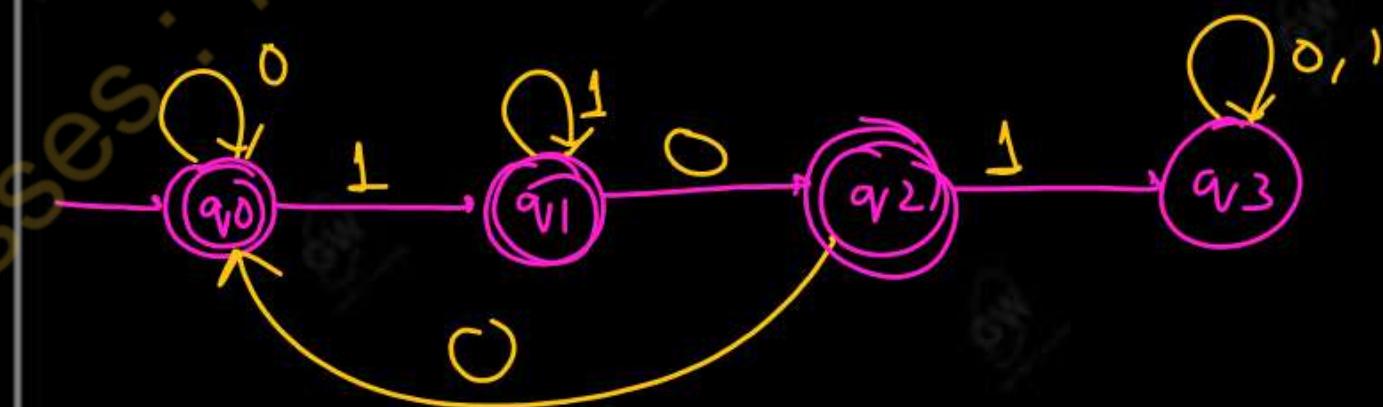
$$L = \{101, 10111, 10100101, 1111010100 \dots\}$$



complement of $L(\bar{L})$

$$\bar{L} = \{ \epsilon, 0, 1, 11, 1000, 001, 1111, 1001 \dots \}$$

(this will not accept 101 as substring)



Wrong string
 $\times 101$
 000101
 11011

Construct a DFA that accept all the string over $\{0,1\}$ 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

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

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

BASIC CONCEPT AND AUTOMATA THEORY

TODAY's TARGET

- DFA for Particular symbol at particular position from left/ right
- DFA for decimal number , binary number divisible by 2, 3 etc

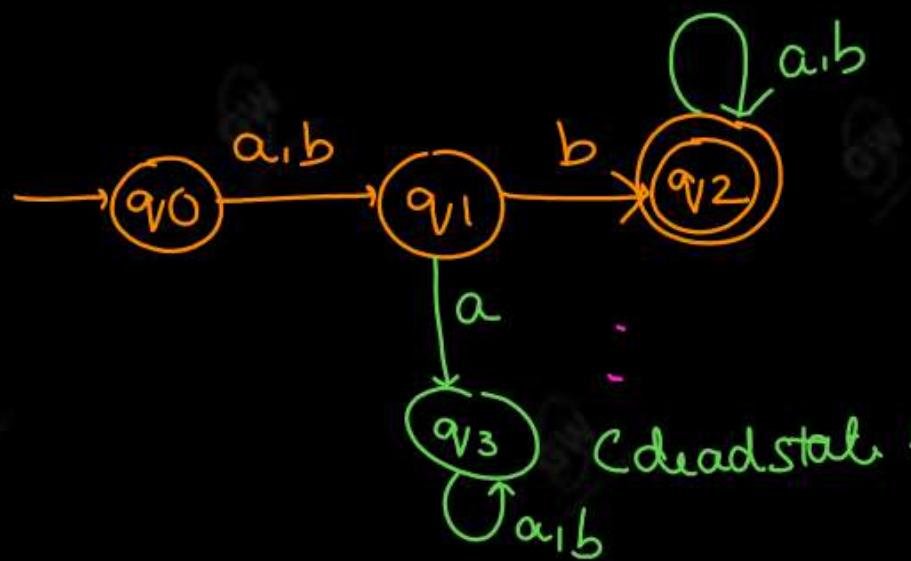
By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

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

TRANSITION DAIGRAM

$L = \{ab, bb, abbbbab, bbababab, \dots\}$



Wrong String
aaba
baba

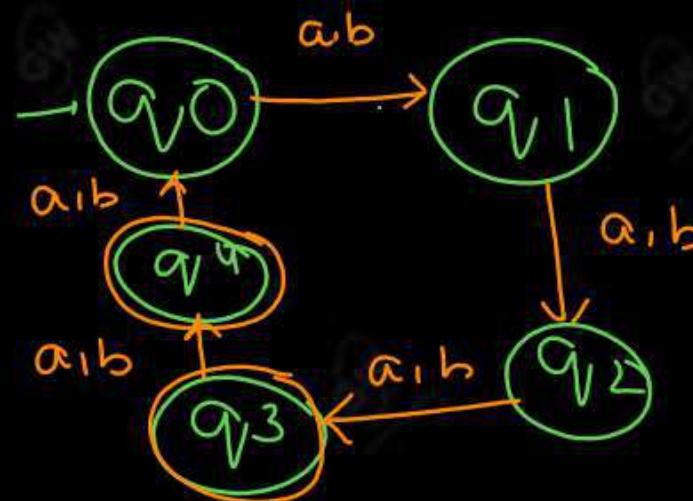
TRANSITION TABLE

	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

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

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

TRANSITION DIAGRAM

 $L = \{aaa, baba, babababa, \dots\}$ 

$w = ababb$
 $|w| = 7$

$w = ababa$
 $|w| = 5$

$w = aba$
 $|w| = 3$

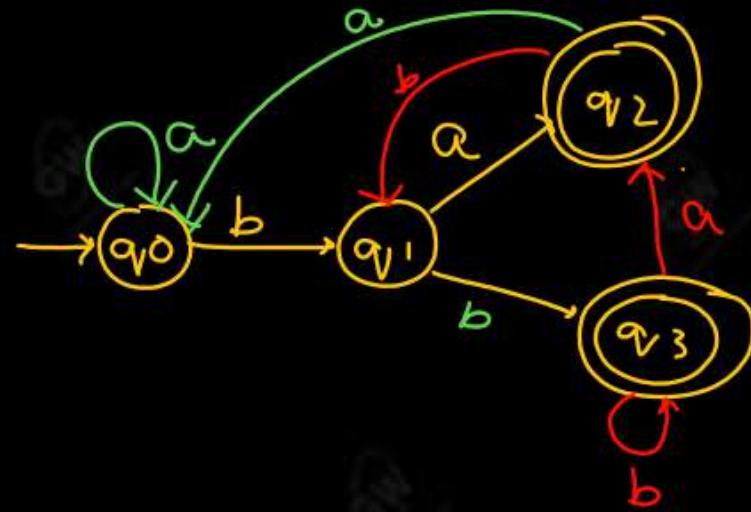
TRANSITION TABLE

	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

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

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

TRANSITION DAIGRAM

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

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)

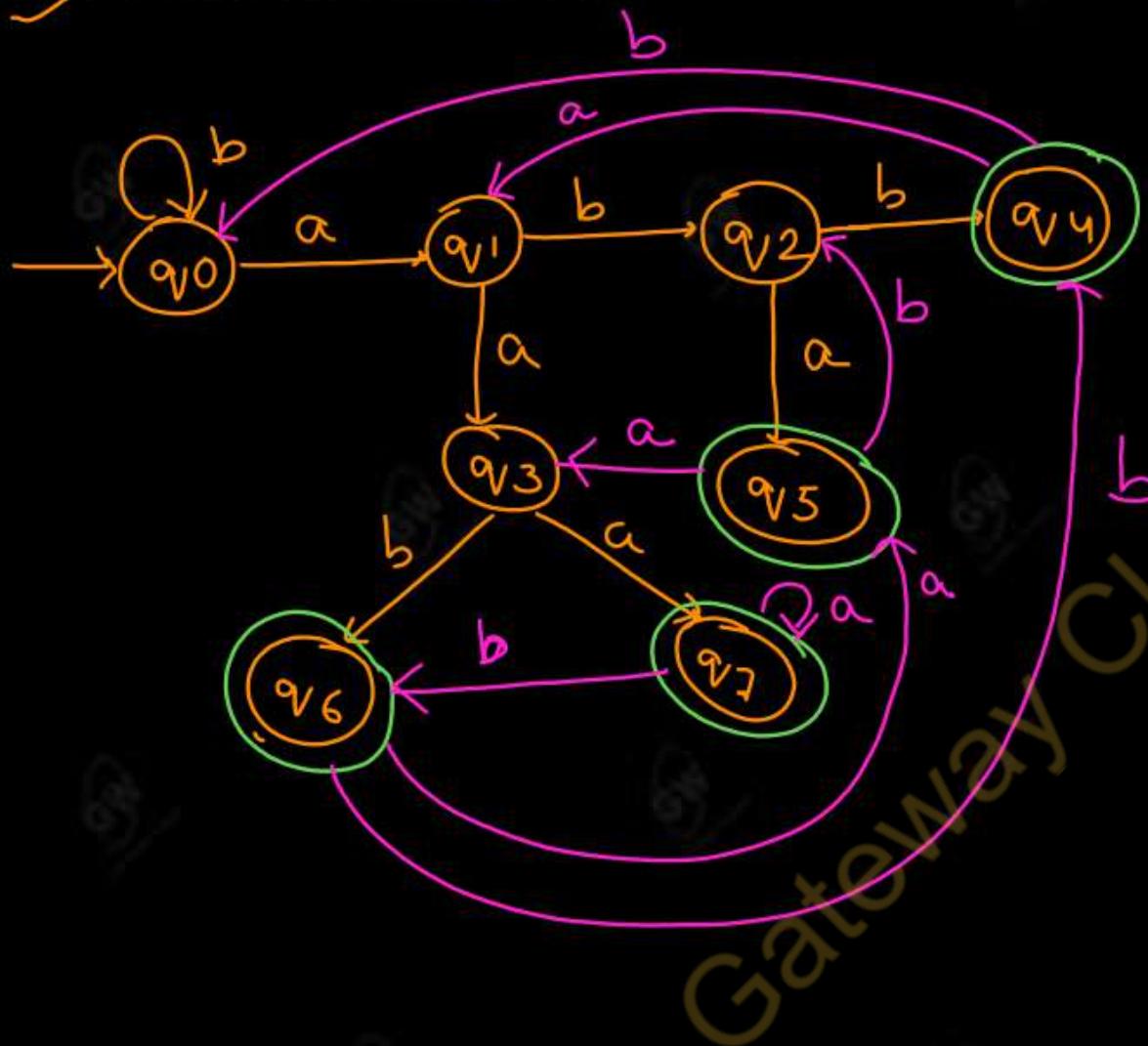
$\{Q, \Sigma, \delta, q_0, F\}$

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

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

$L = \{aaa, baaa, bbbaba, \dots\}$



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

$\emptyset, \epsilon, \delta, q_0, F$

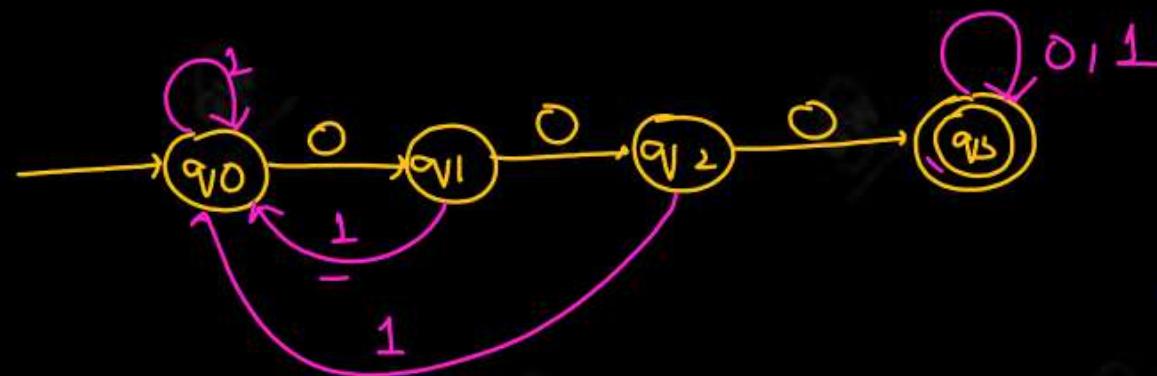
$\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}, \{\{a, b\}\}, \delta, q_0$

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

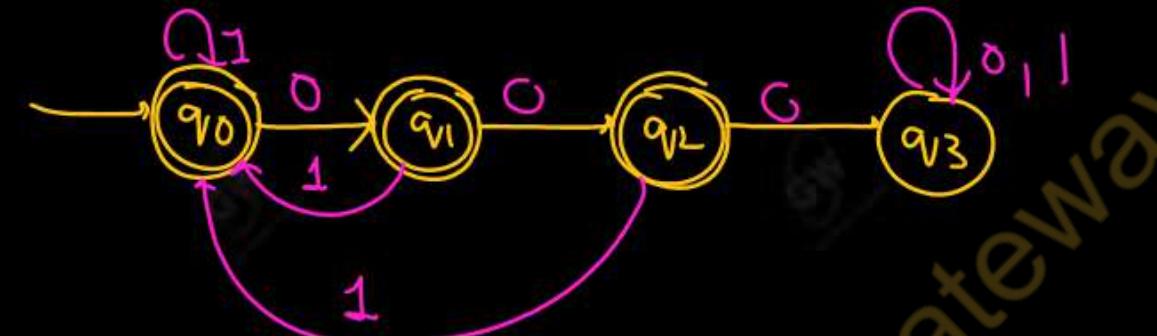
Design a DFA that accept all string over the $\Sigma = \{0,1\}$ such that every accepted all string which has three consecutive ~~three~~ zeros

TRANSITION DIAGRAM

$$L = \{000, 11010000, 000111, 101000111, \dots\}$$



Do not contain consecutive three zeros



$$\overline{L} = \{0, 00, 11, 10, 01, 11010001, \dots\}$$

Complement of L

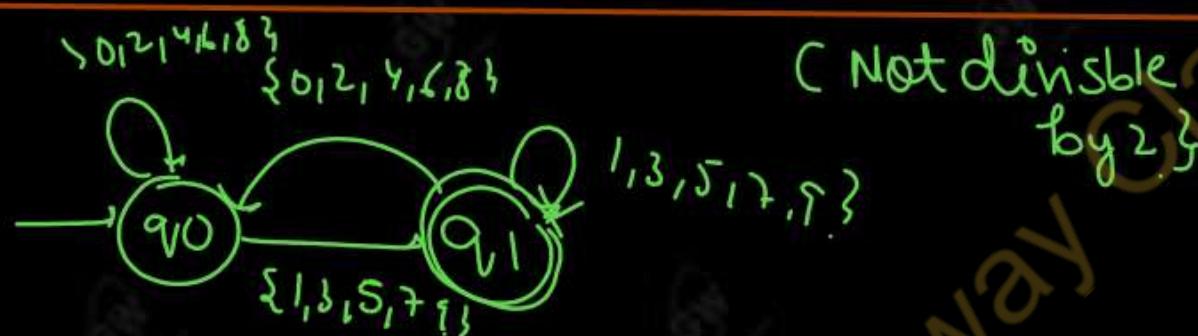
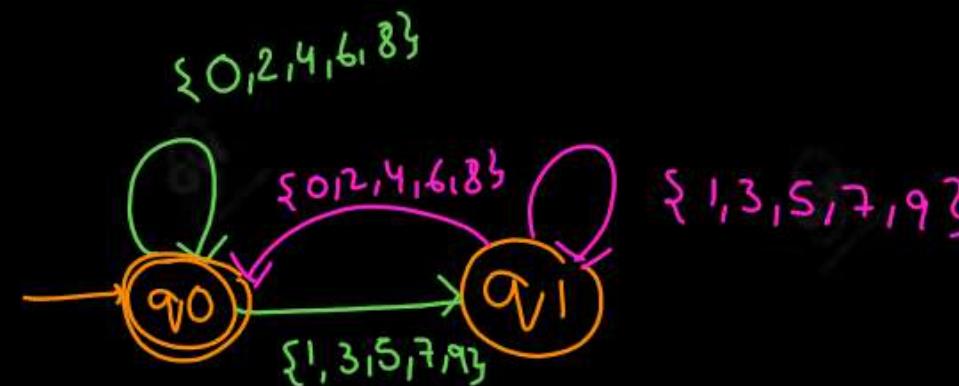
TRANSITION TABLE

	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

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

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

TRANSITION DIAGRAM

 $L = \{2, 4, 8, 10, 12, 14, 20, 22, 24, \dots\}$


TRANSITION TABLE

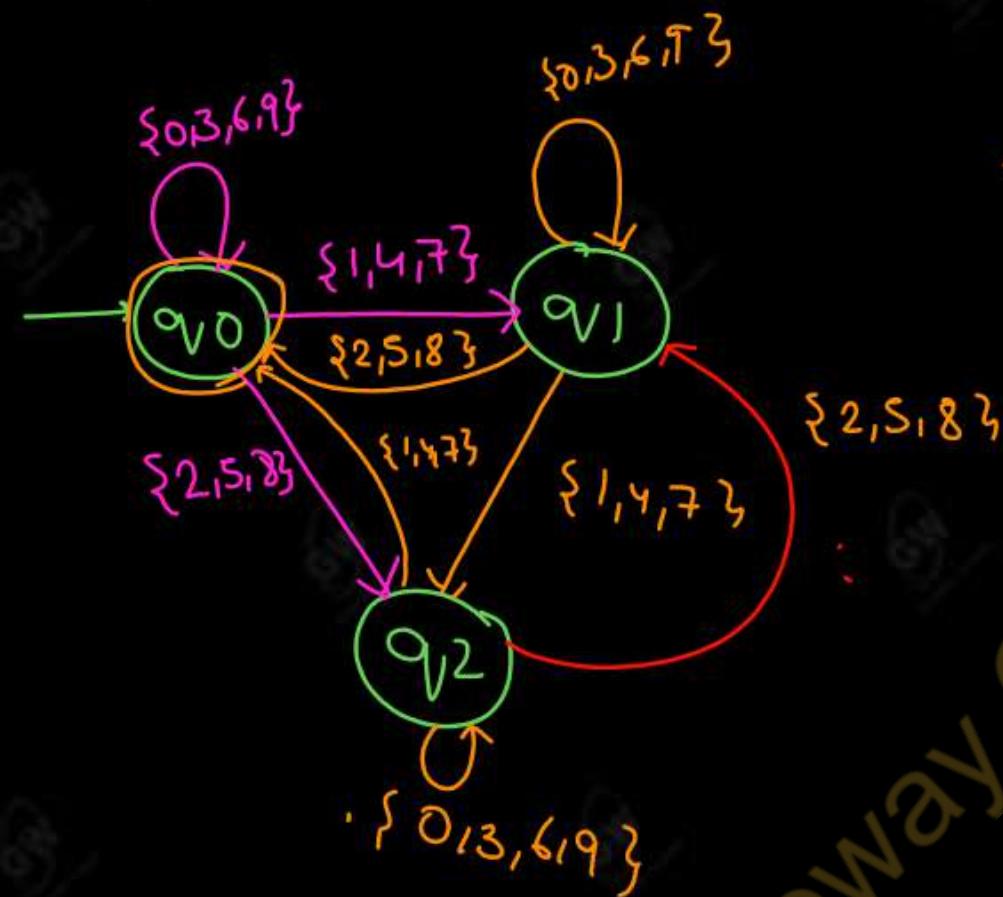
	$\{0,2,4,6,8\}$	$\{1,3,5,7,9\}$
$\rightarrow^* q_0$	q_0	q_1
q_1	q_0	q_1

 $\{Q, \Sigma, \delta, q_0, F\}$
 $\{q_0, q_1\}$
 $\delta, q_0, \{q_0\}$
 $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

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

TRANSITION DIAGRAM

$$L = \{0, 3, 6, 9, \dots\}$$



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

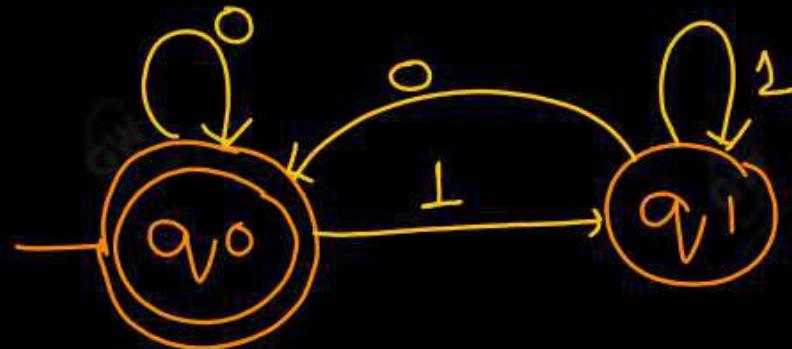
$$\{Q, \Sigma, \delta, q_0, F\}$$

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

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

$$L = \{0, 00, 10, 000, 010, 100, \dots\}$$



TRANSITION TABLE

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

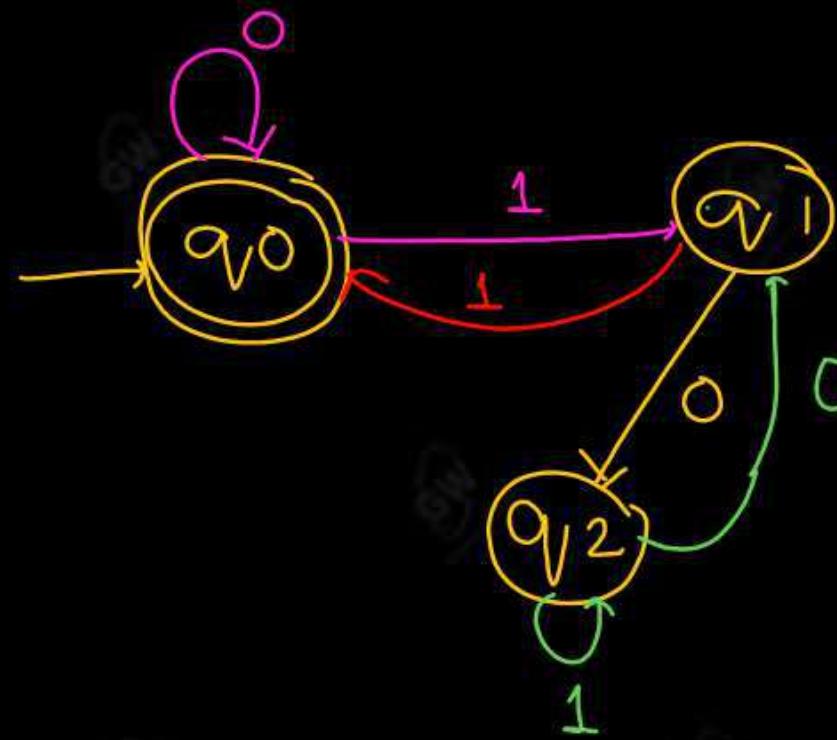
$$\{Q, \Sigma, \delta, q_0, F\}$$

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

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

TRANSITION DIAGRAM

$$L = \{0, 11, 110, \dots\}$$



TRANSITION TABLE

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

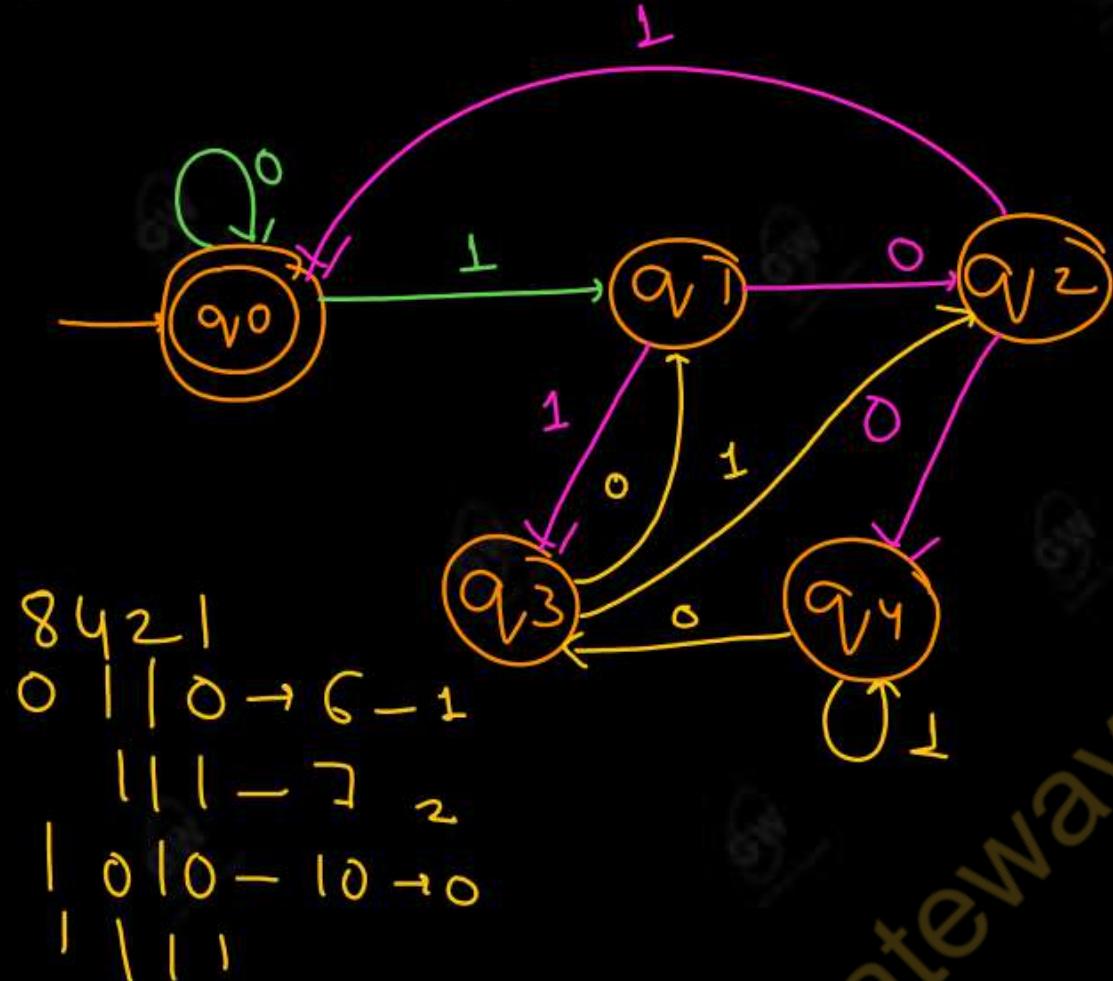
$$\{Q, \Sigma, \delta, q_0, F\}$$

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

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

TRANSITION DIAGRAM

$$L = \{0, 101, 1010, \dots\}$$



TRANSITION TABLE

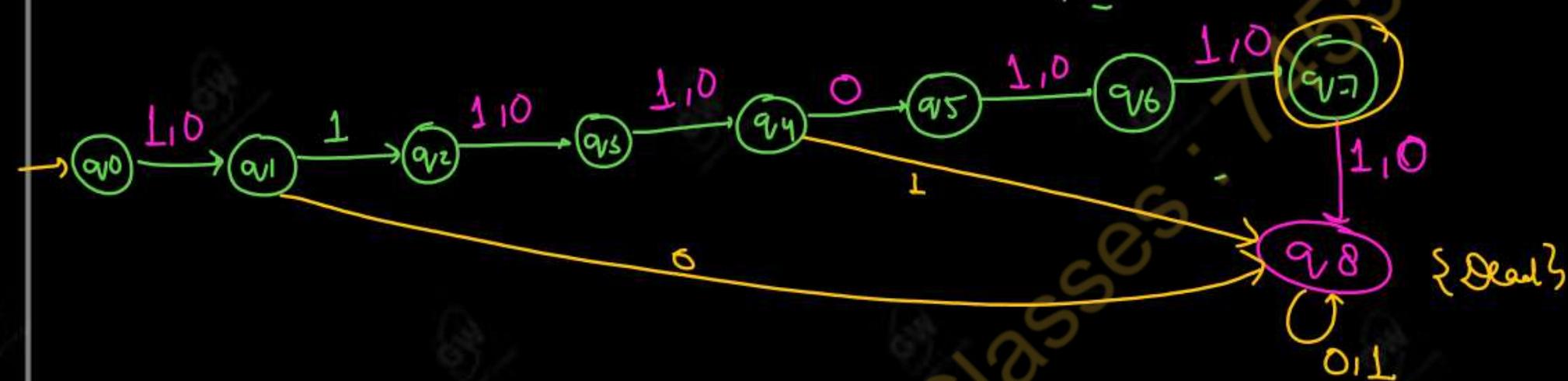
	0	1
$\xrightarrow{*} q_0$	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

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

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

Construct a DFA for which accept the string of all 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

BASIC CONCEPT AND AUTOMATA THEORY

TODAY's TARGET

- Minimization of DFA
- Practice questions

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

➤ 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 state but logically always one dead state is sufficient to complete the functionality

➤ **UNREACHABLE STATE**

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

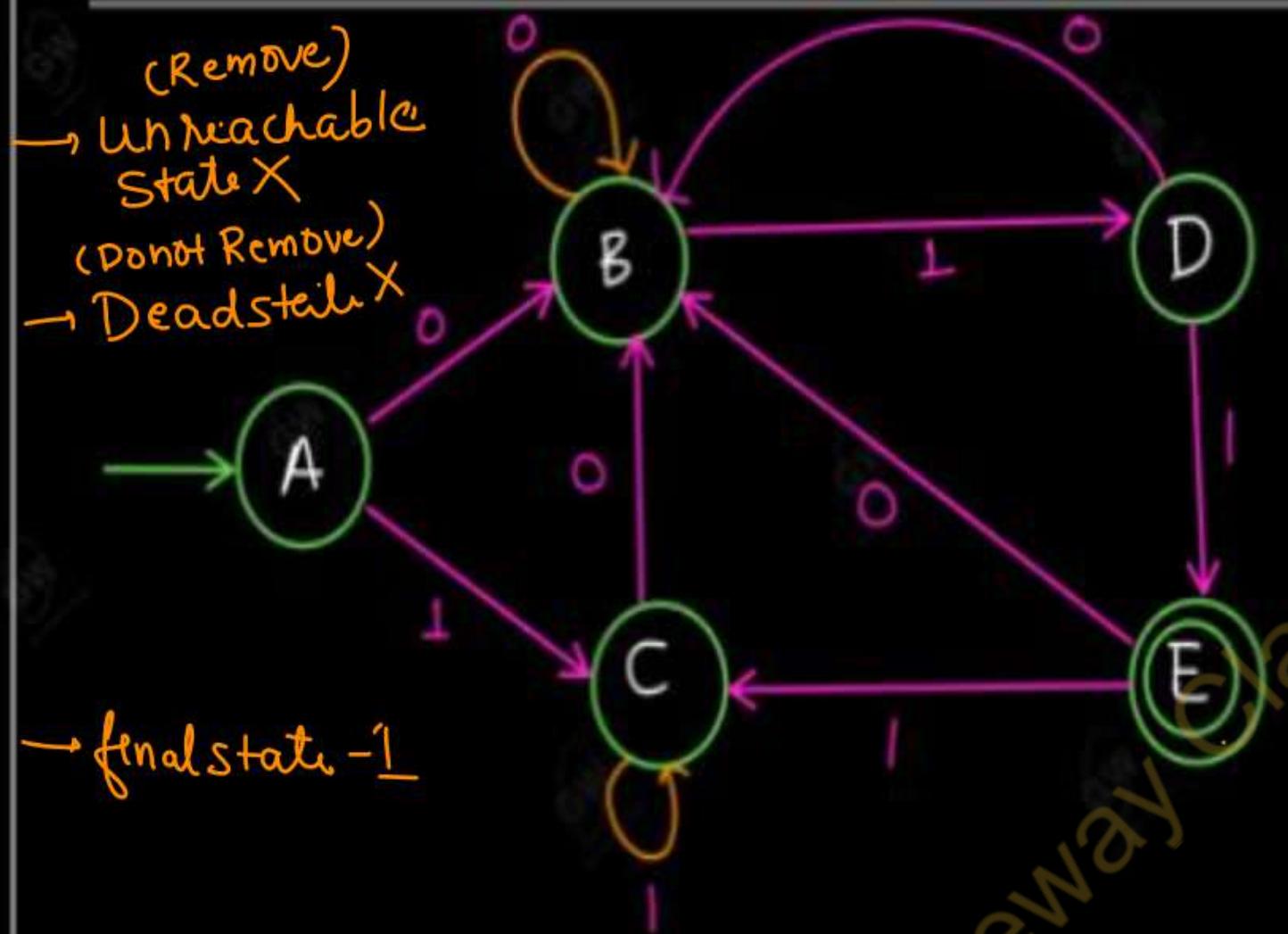
➤ **EQUIVALENT STATE**

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 equivalence 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



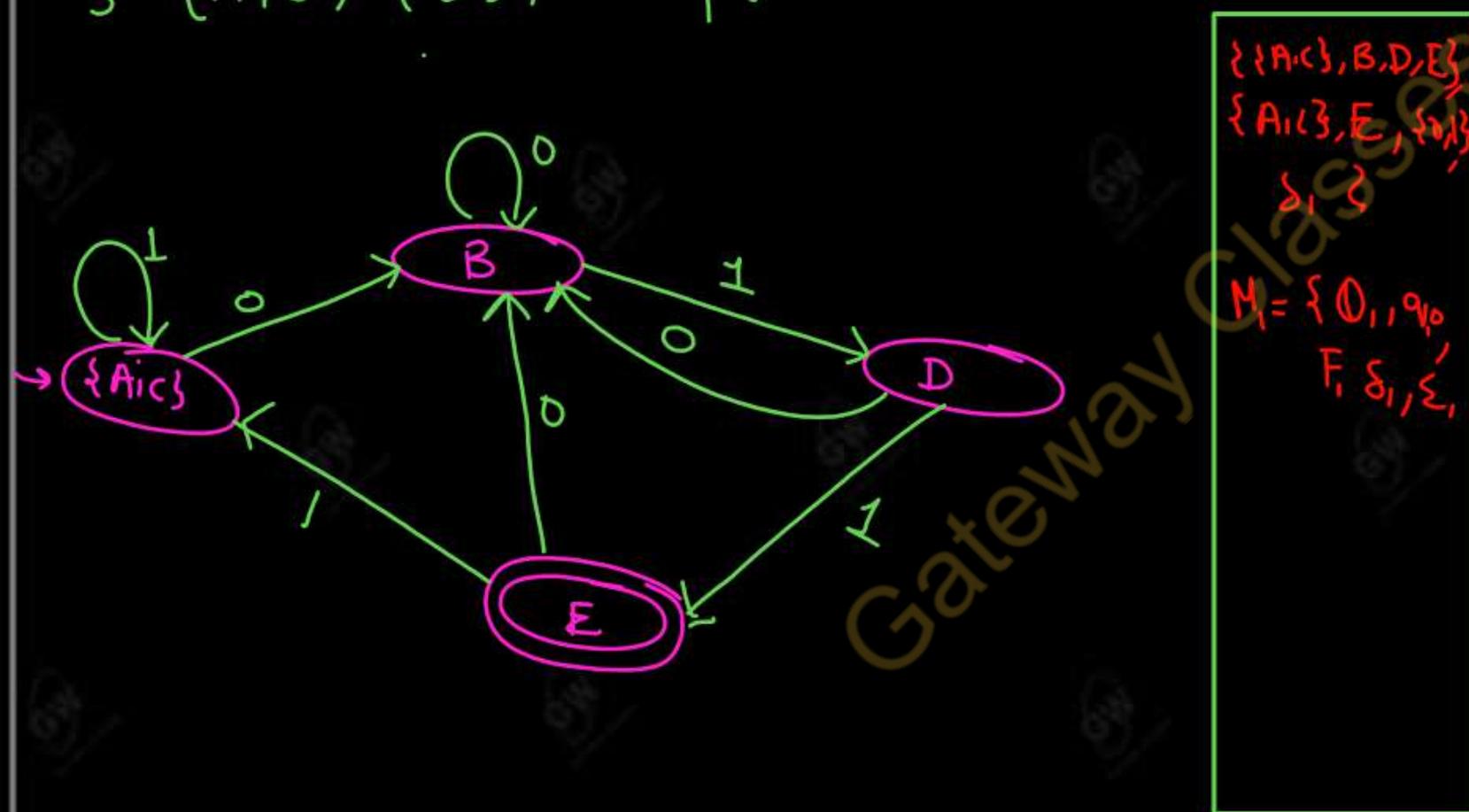
TRANSITION TABLE

	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-equivalence class
 $\pi_2 = \{A, C\}$ {B} {D} {E} - 2-equivalence class
 $\pi_3 = \{A, C\}$ {B} {D}, {E} 3-equivalence



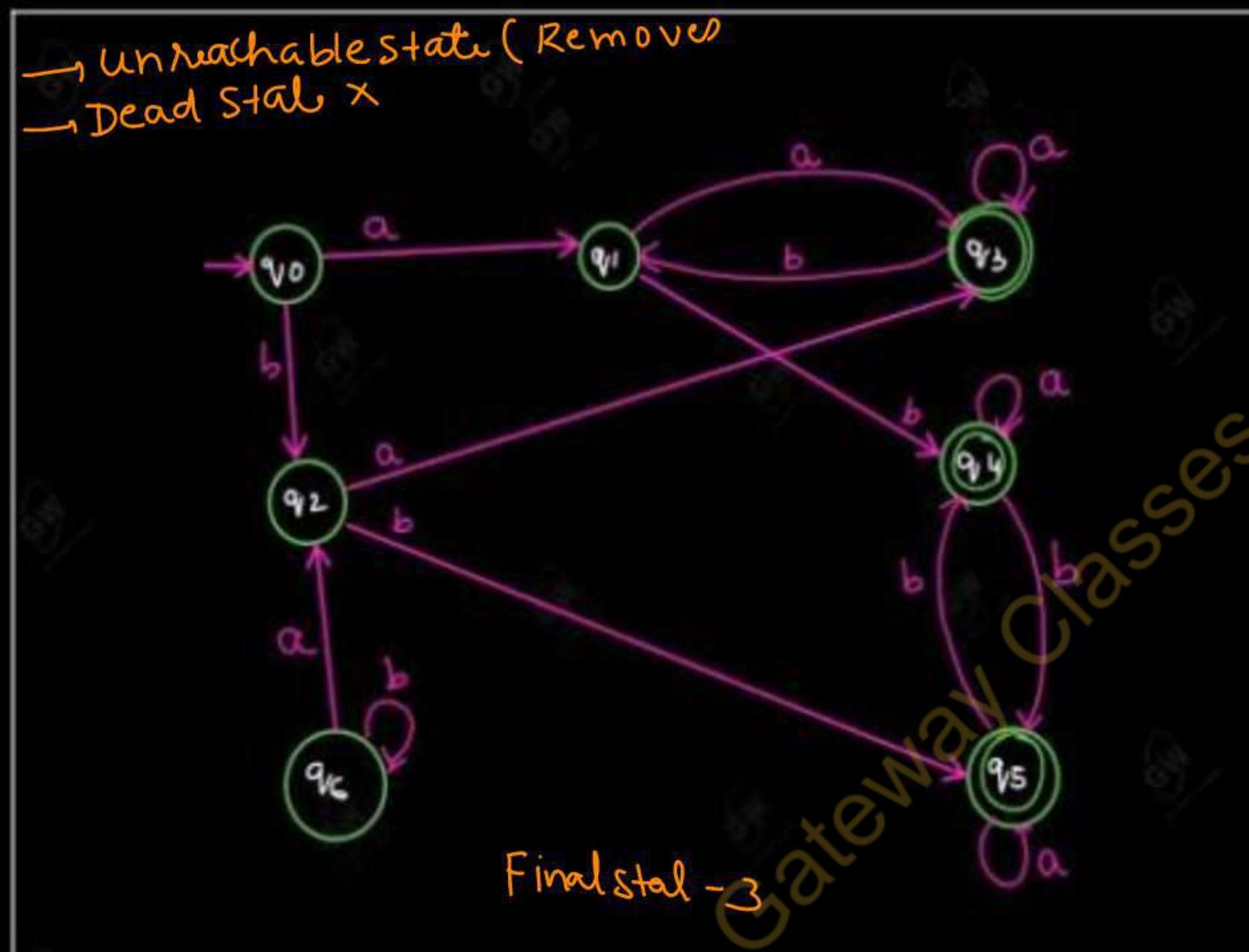
TRANSITION TABLE

	0	1
$\rightarrow A$	B	C
B	B	D
C	B	C
D	B	E
*E	B	C

MINIMIZED TRANSITION TABLE

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

2. MINIMIZATION OF DFA



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
q_6	q_2	q_6

q_6 unreachable state

2. MINIMIZATION OF DFA

EQUIVALENCE CLASS

$$\pi_0 = \{q_0, q_1, q_2\} \quad \{q_3, q_4, q_5\} \quad 0\text{-equivalence class}$$

$$\pi_1 = \{q_0\}, \{q_1, q_2\}, \{q_3\}, \{q_4, q_5\}$$

$$\pi_2 = \{q_0\}, \{q_1, q_2\}, \{q_3\}, \{q_4, q_5\} \quad 1\text{-equivalence class}$$

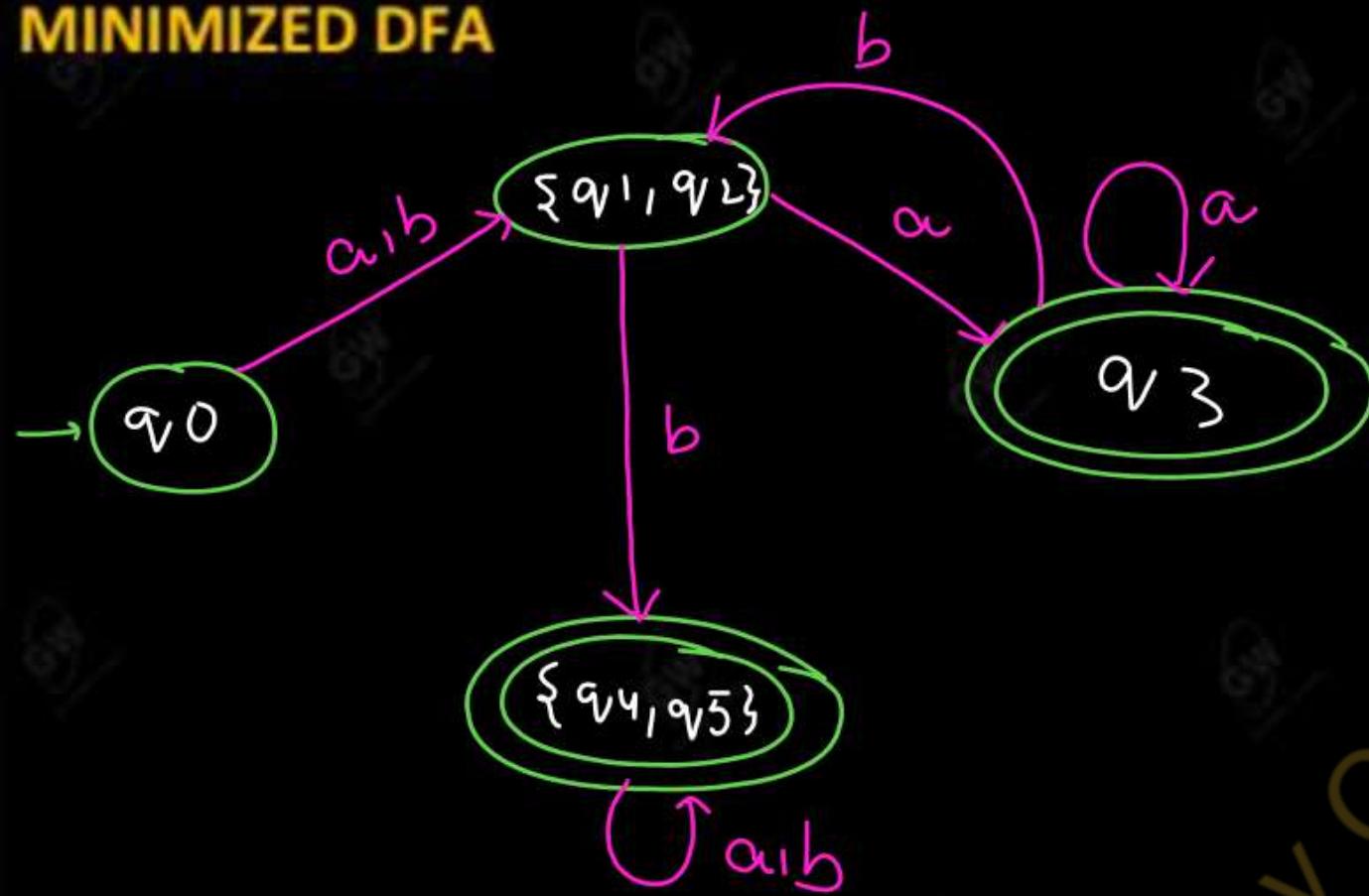
(2 equivalence classes)

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

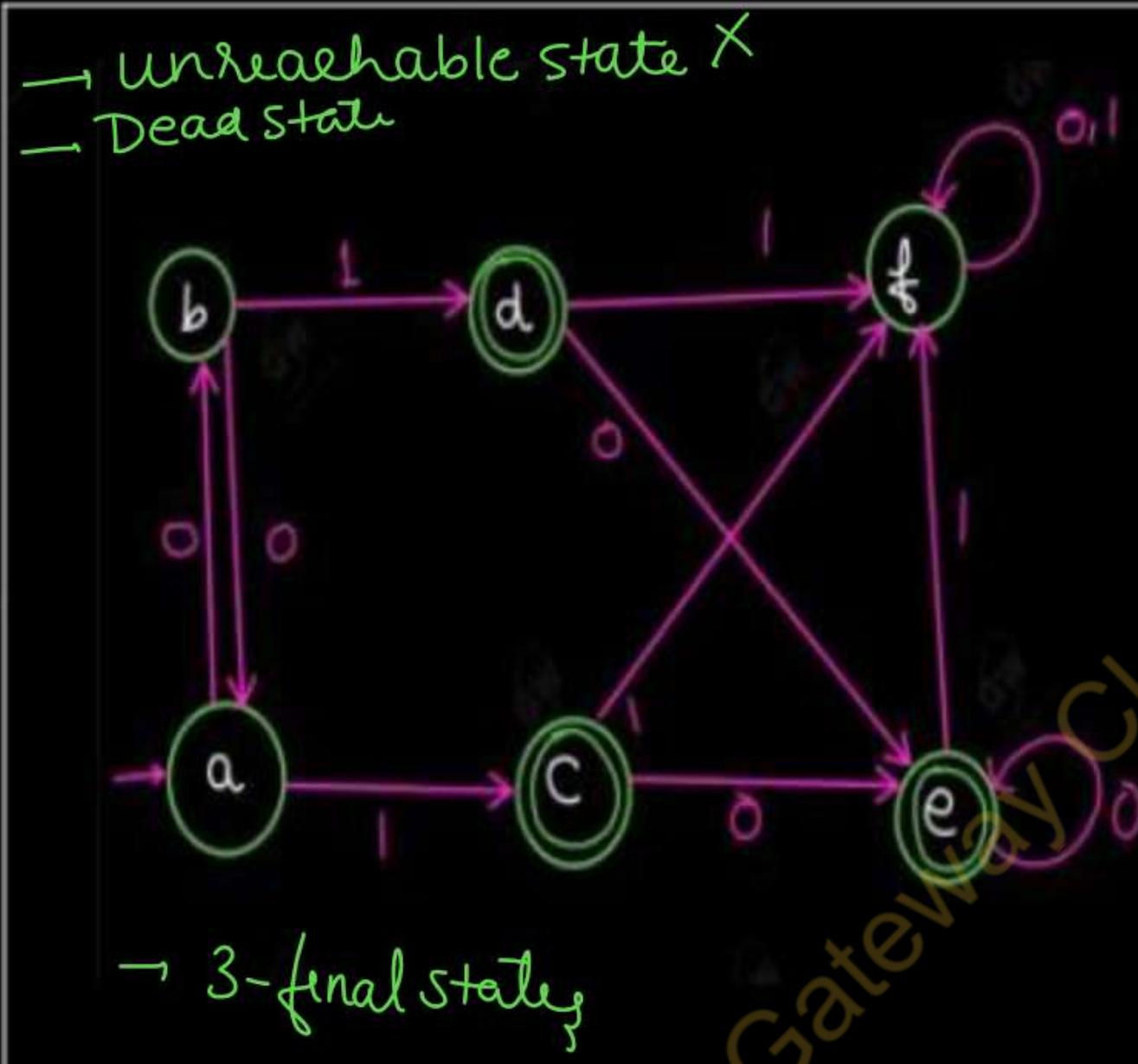
MINIMIZED DFA



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



TRANSITION TABLE

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

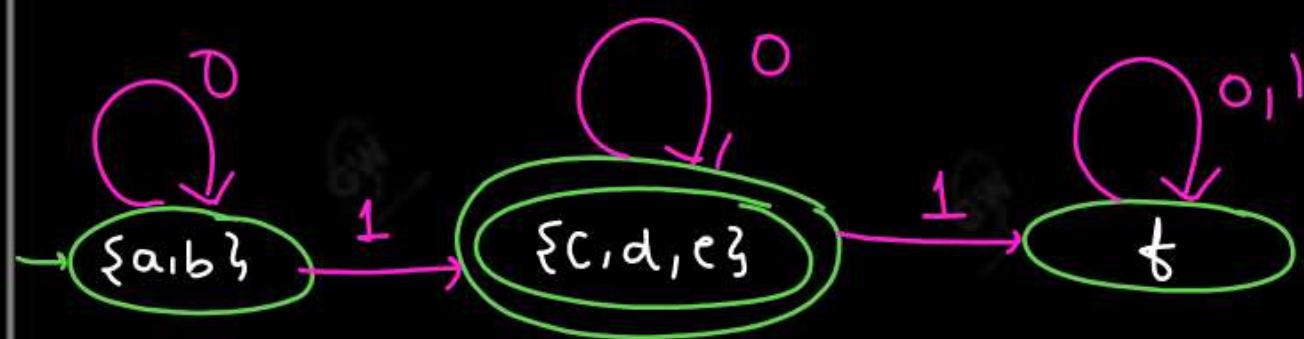
f dead state

3. MINIMIZATION OF DFA

EQUIVALENCE CLASS $\pi_0 = \{a, b, f\} \quad \{c, d, e\}$ 0-equivalence class $\pi_1 = \{a, b\} \{f\} \quad \{c, d, e\}$ 1-equivalence class $\pi_2 = \{a, b\}, \{f\} \quad \{c, d, e\}$ 2-equivalence class**TRANSITION TABLE**

	0	1
$\rightarrow a$	b	c
b	a	d
*c	e	f
*d	e	f
*e	e	f
f	f	f

MINIMIZED DFA



TRANSITION TABLE

	0	1
$\rightarrow\{a,b\}$	{a,b}	{c,d,e}
$*\{c,d,e\}$	{c,d,e}	f
f	f	f

BASIC CONCEPT AND AUTOMATA THEORY

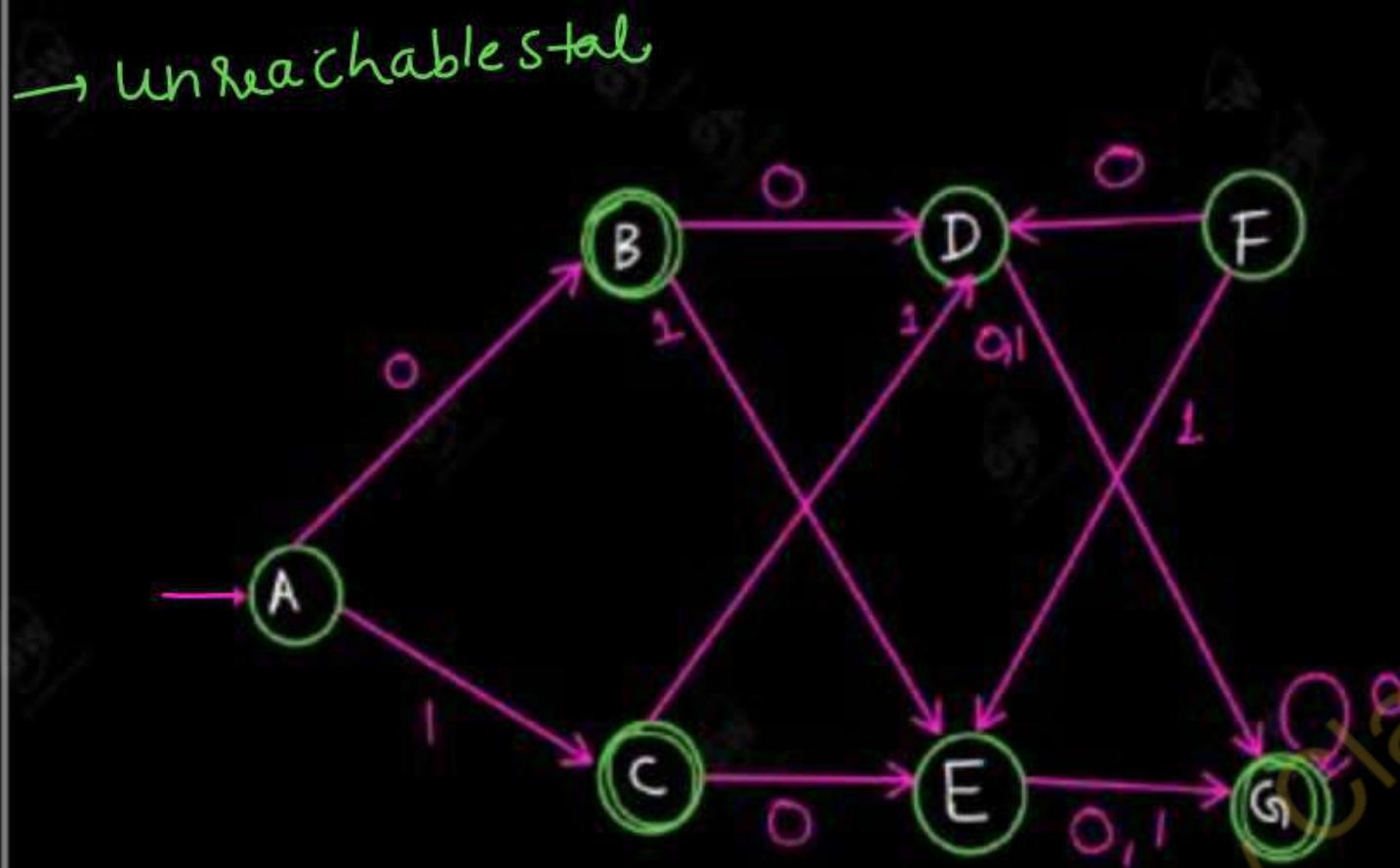
TODAY s' TARGET

- Minimization of DFA practice questions of different types

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E)

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

EQUIVALENCE CLASS $\pi_0 = \{A, D, E\} \quad \{B, C, G\}$ 0-equivalence class $\pi_1 = \{A, D, E\} \quad \{B, C\} \quad \{G\}$ 1-equivalence class $\pi_2 = \{A\}, \{D, E\}, \{B, C\}, \{G\}$

2-equivalence class

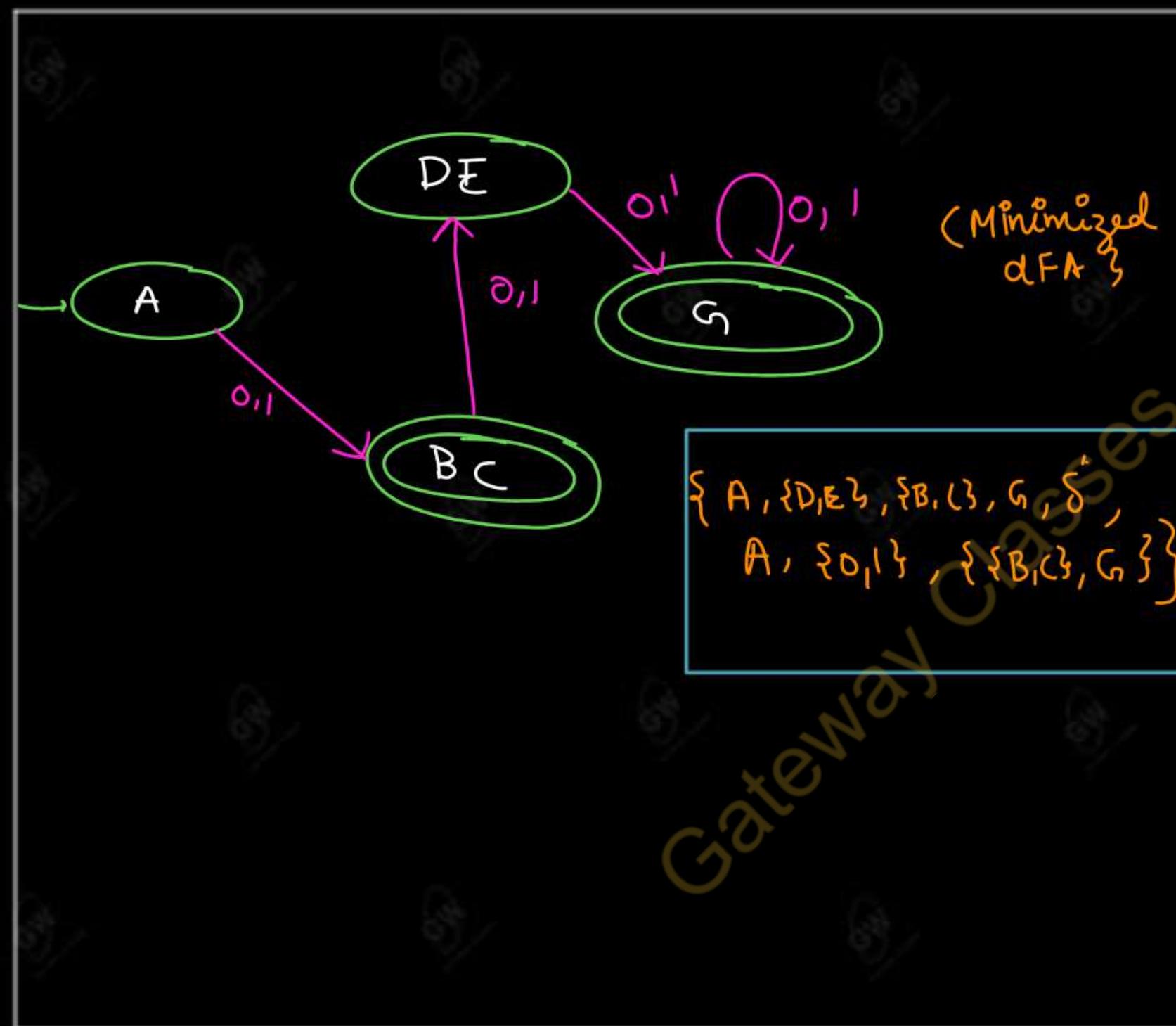
 $\pi_3 = \{A\}, \{D, E\}, \{B, C\}, \{G\}$

3-equivalence class

TRANSITION TABLE

	0	1
$\rightarrow A$	B	C
*B	D	E
*C	E	D
D	G	G
E	G	G
*G	G	G

4 MINIMIZATION OF DFA

(Minimized)
TRANSITION TABLE

	0	1
$\rightarrow A$	{B, C}	{B, C}
$\{D, E\}$	G	G
$*\{B, C\}$	{D, E}	{D, E}
$*G$	G	G



TRANSITION TABLE

	0	1
$\rightarrow q_0$	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

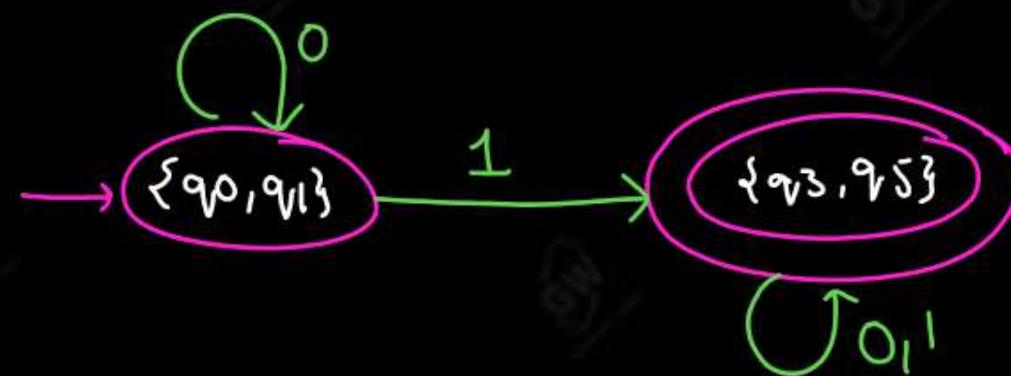
Note

 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



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

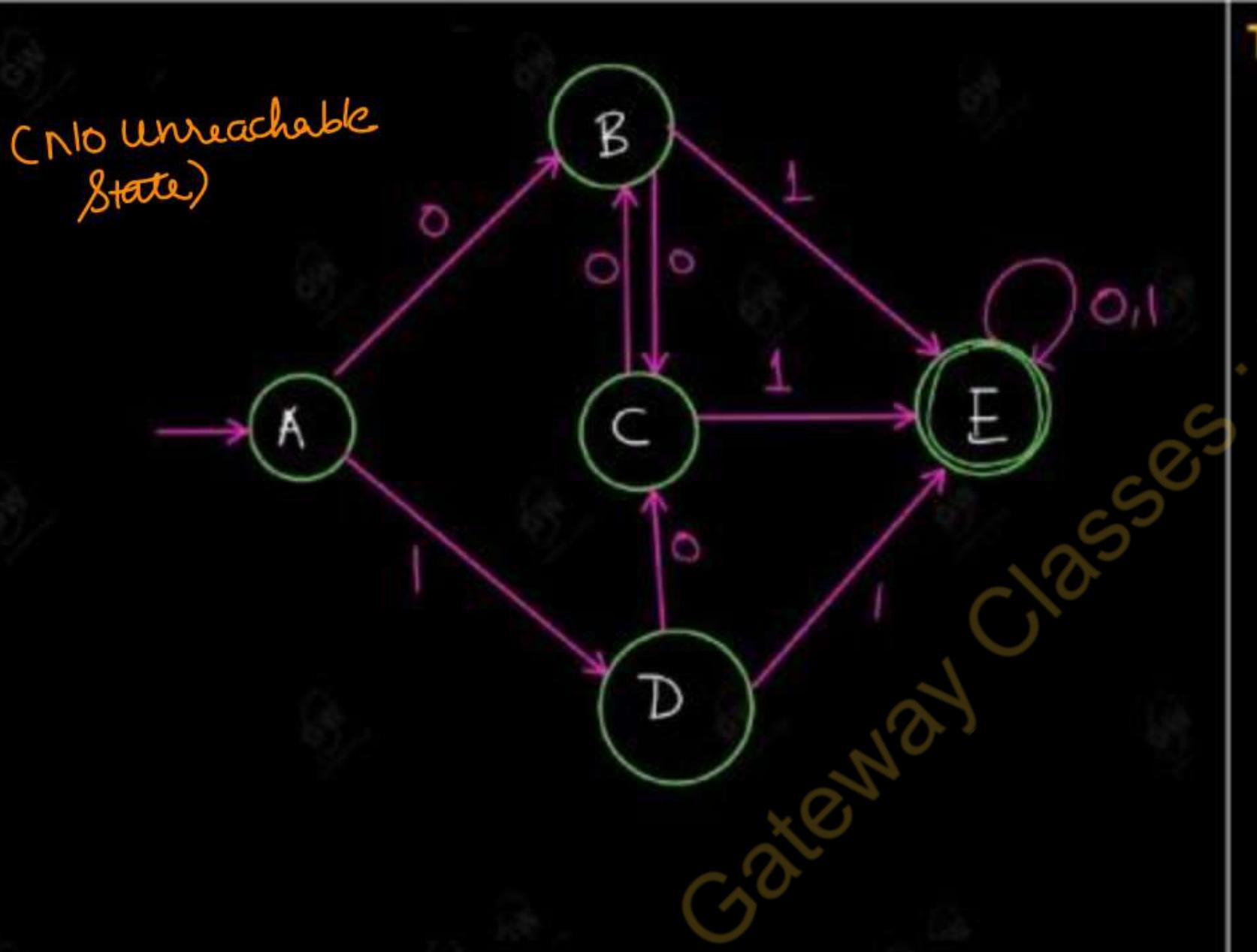
TRANSITION TABLE

	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

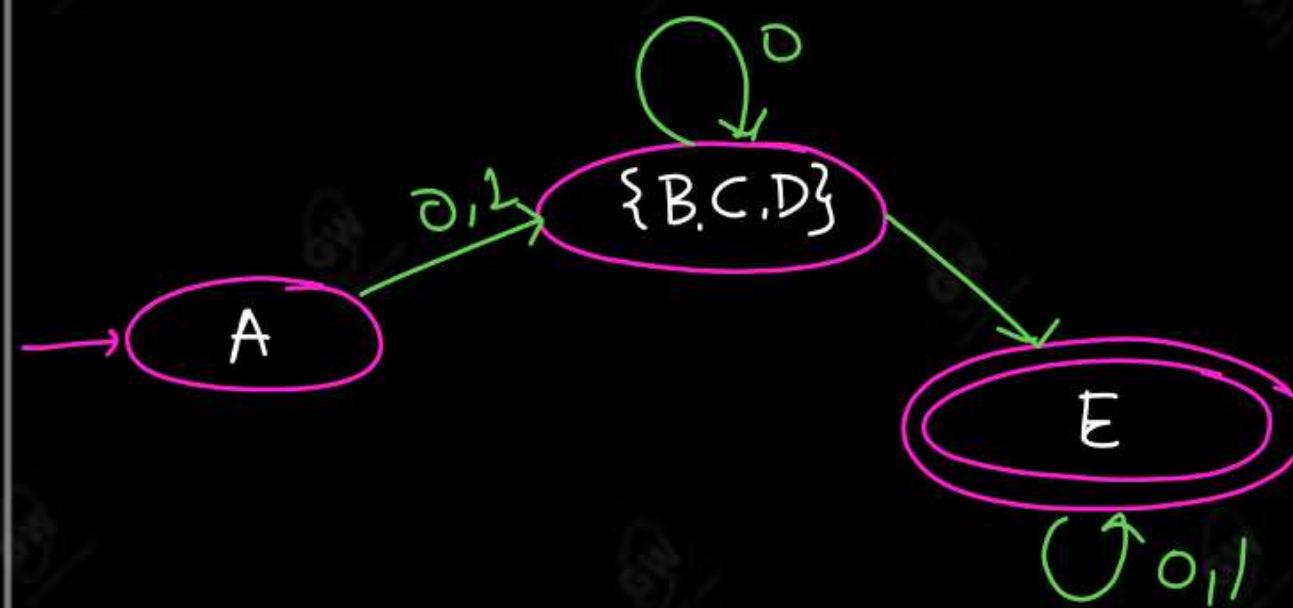
$\pi_2 = \{A\}, \{B, C, D\}, \{E\}$ 2-equivalence Class

TRANSITION TABLE

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

6 MINIMIZATION OF DFA

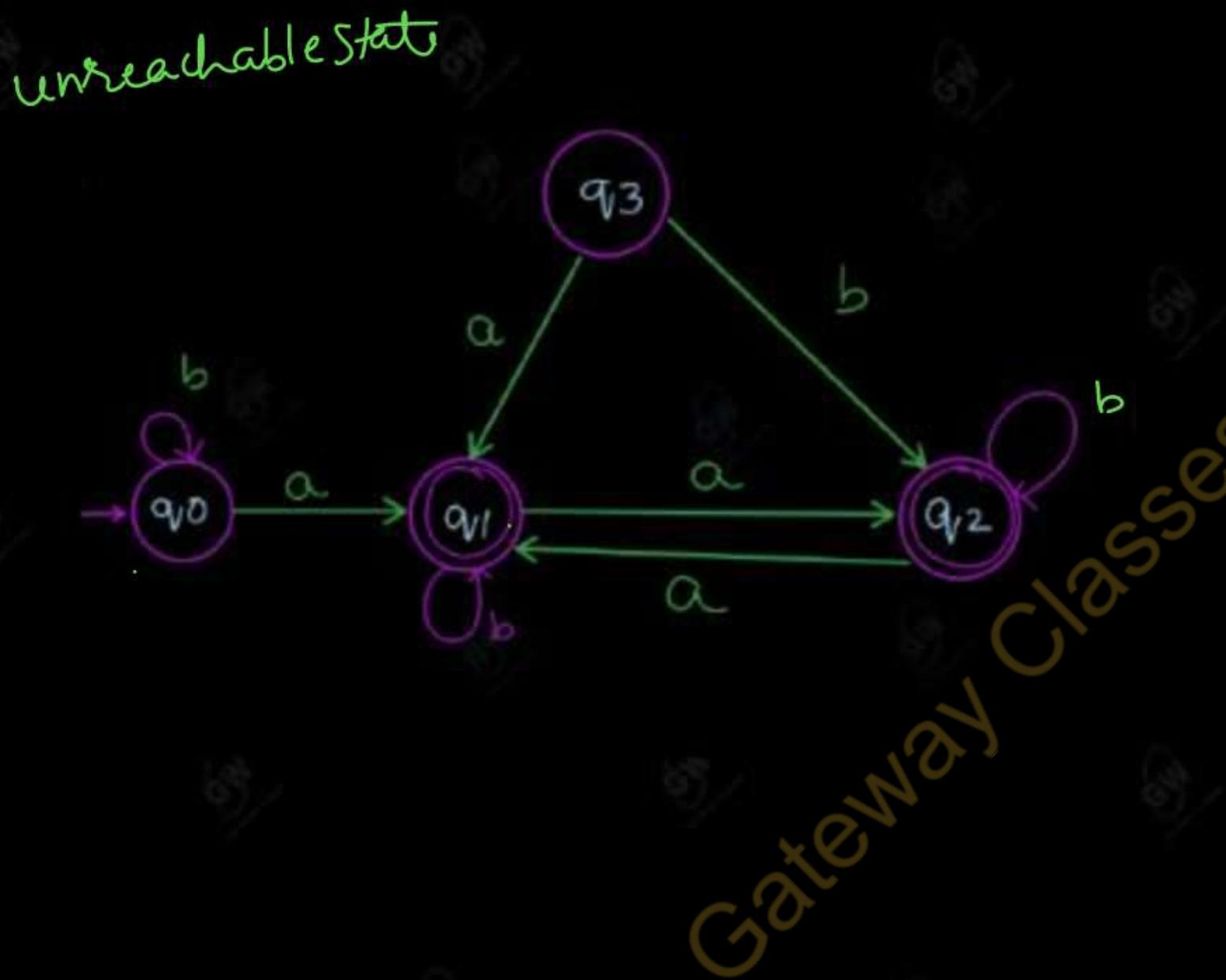
MINIMIZED DFA



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



TRANSITION TABLE

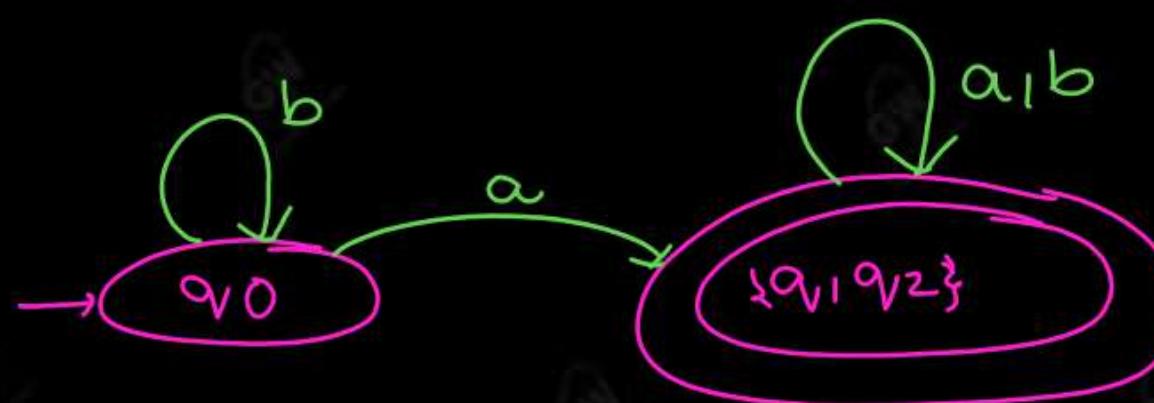
	a	b
$\rightarrow q_0$	q_1	q_0
$*q_1$	q_2	q_1
$*q_2$	q_1	q_2
q_3	q_1	q_2

q_3 is unreachable state

7 MINIMIZATION OF DFA

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

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



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
$*\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_1, q_2\}$

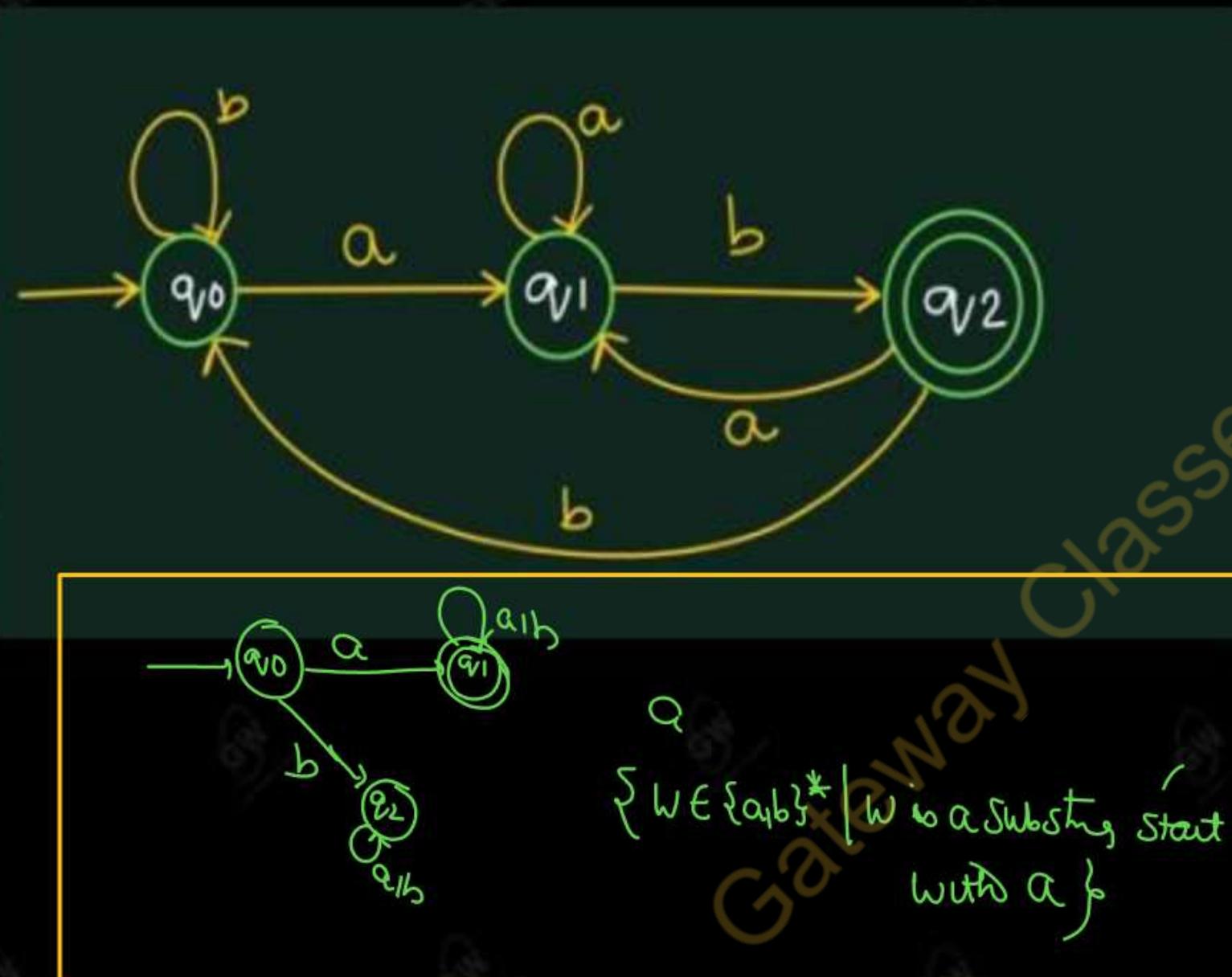
BASIC CONCEPT AND AUTOMATA THEORY

TODAY's TARGET

- HOW TO FIND ANY LANGUAGE IF DFA IS GIVEN
- HOW FIND THE COMPLEMENT OF DFA

By PRAGYA RAJVANSHI
B.Tech, M.Tech(C.S.E)

What language accepted by this DFA



Explanation

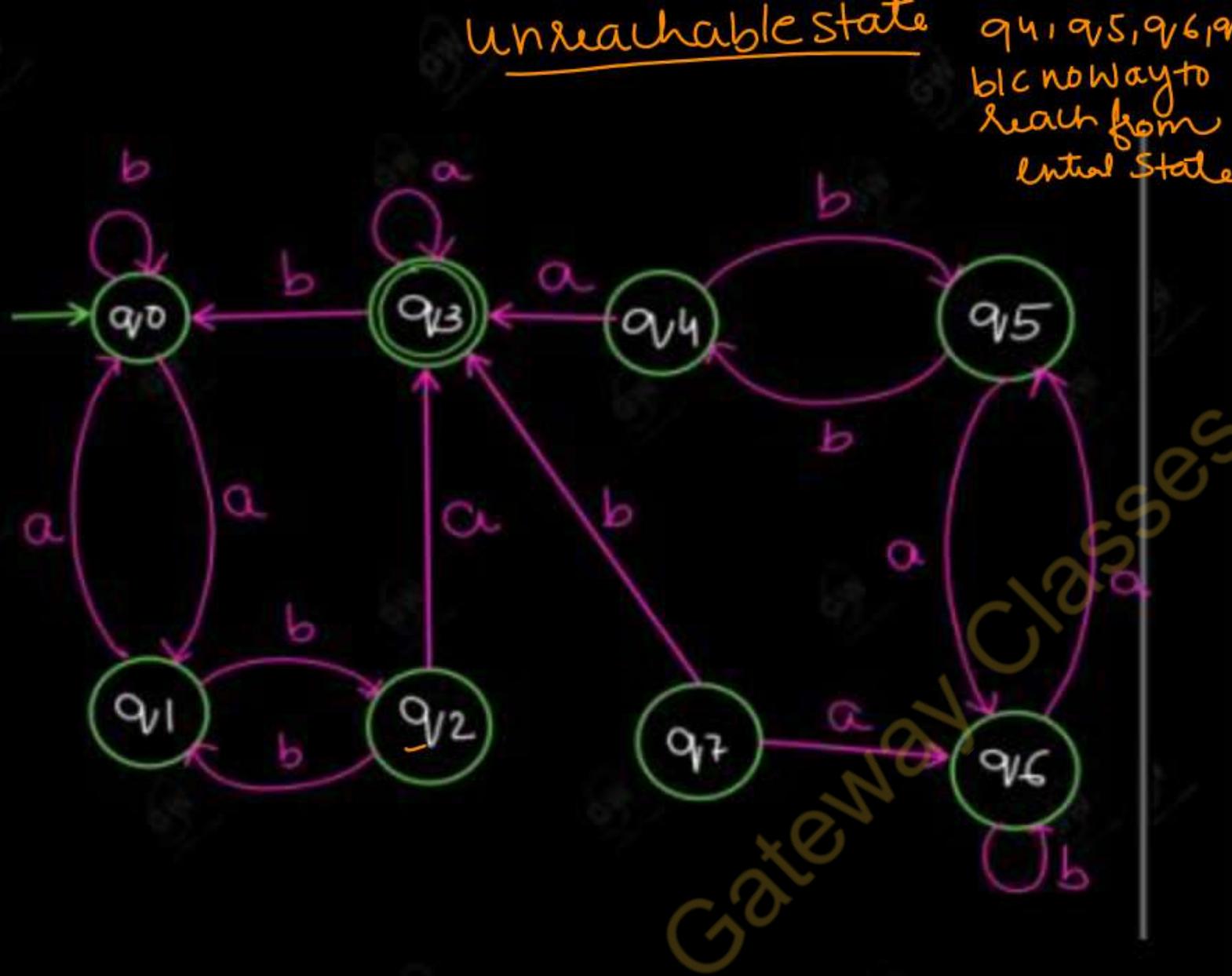
- \in belongs to
- $w \rightarrow ab$ $w \rightarrow abab$ $w \rightarrow abaabb$
- dead state \rightarrow NO \rightarrow Starting with ab X
- ✓ Ending with ab
- Substring ab \rightarrow X q_2 loop a, b

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

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

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

8. MINIMIZATION OF DFA



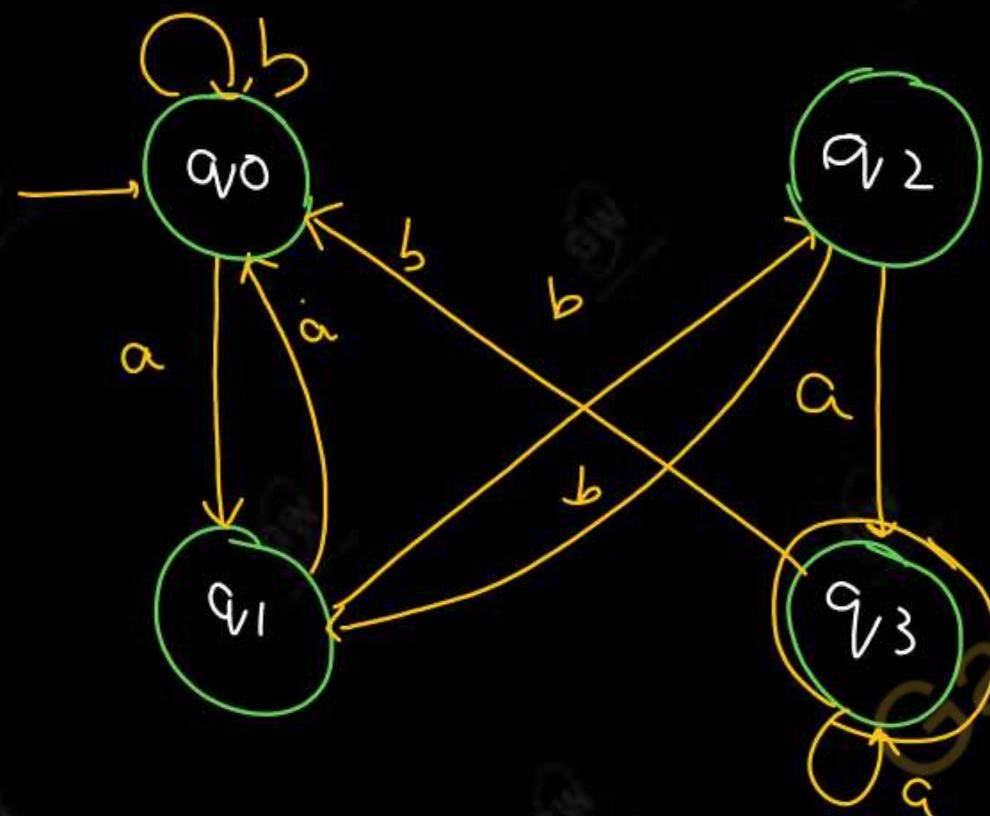
TRANSITION TABLE

	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

- $\pi_0 = \{q_0, q_1, q_2\} \quad \{q_3\}$ 0-equivalence class
 $\pi_1 = \{q_0, q_1\} \quad \{q_2\}, \{q_3\}$ 1-equivalence class
 $\pi_2 = \{q_0\} \{q_1\} \quad \{q_2\}, \{q_3\}$ 2-equivalence class
 $\pi_3 = \{q_0\}, \{q_1\}, \{q_2\}, \{q_3\}$ 3-equivalence class



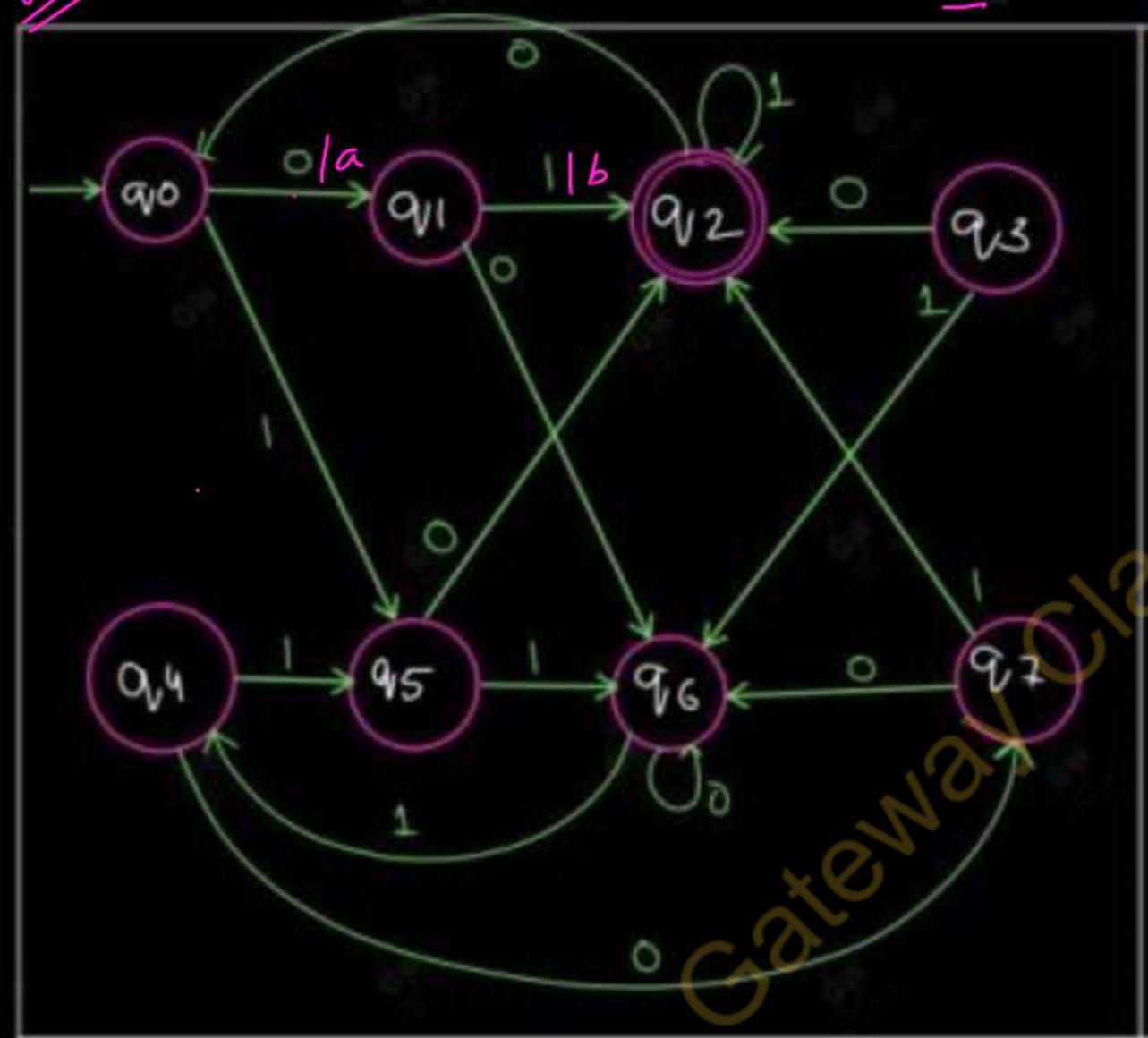
TRANSITION TABLE

	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

DONElet $a = 0$
 $b = 1$ 

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_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

$$\bar{\pi}_0 = \{q_0, q_1, q_4, q_5, q_6, q_7\} \quad \{q_2\}$$

$$\bar{\pi}_1 = \{q_0, q_4, q_6\} \quad \{q_1, q_7\} \quad \{q_5\} \quad \{q_2\}$$

$$\bar{\pi}_2 = \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_5\}, \{q_2\}$$

$$\bar{\pi}_3 = \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_5\} \quad \{q_2\}$$

NOTE - If DFA is Given first make DFA

from table Why?

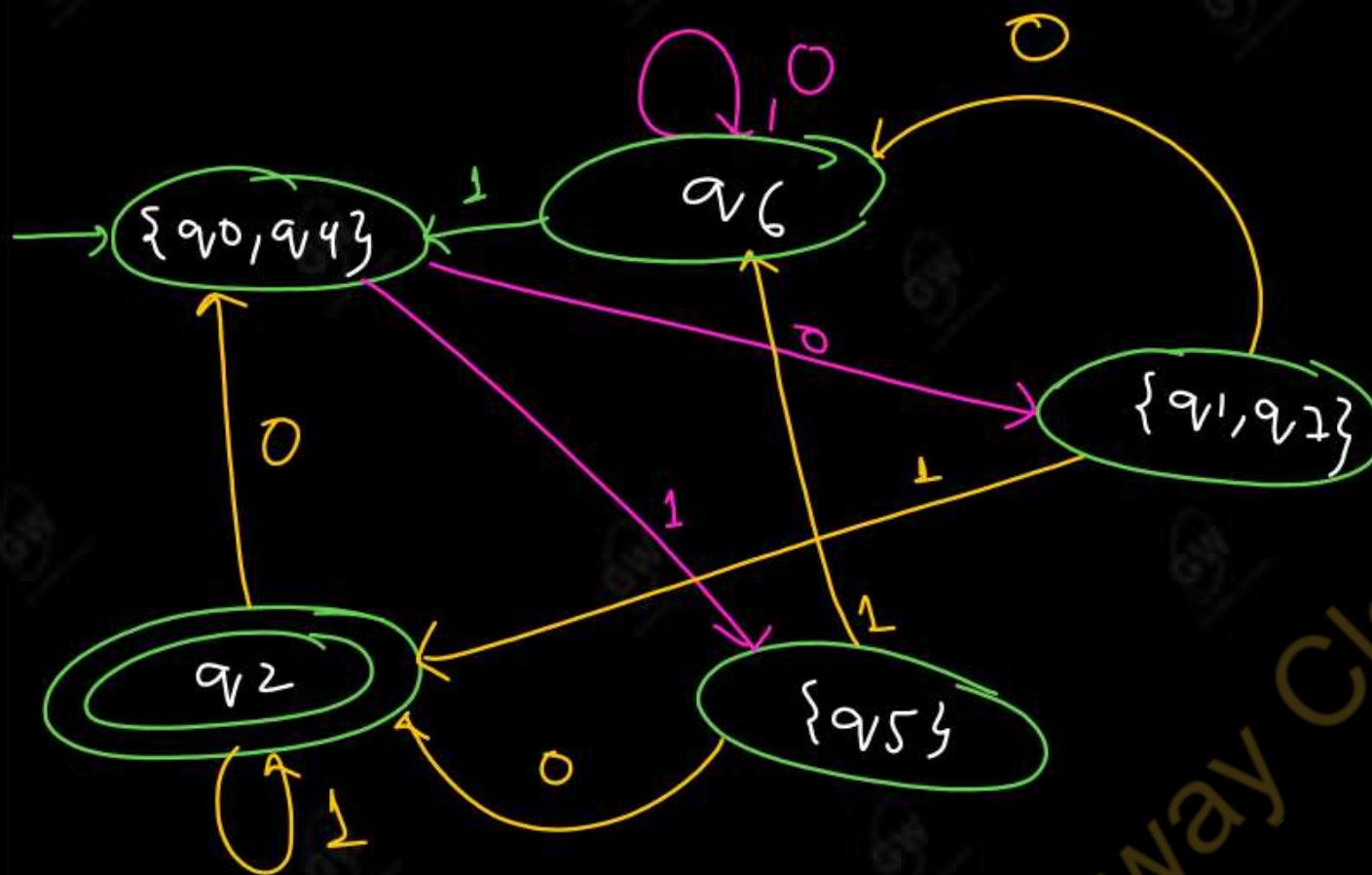
So that you can find unreachable states

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 Diagram



MINIMIZED TRANSITION TABLE

	0 a	1 b
$\rightarrow \{q_0, q_4\}$	$\{q_1, q_7\}$	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

BASIC CONCEPT AND AUTOMATA THEORY

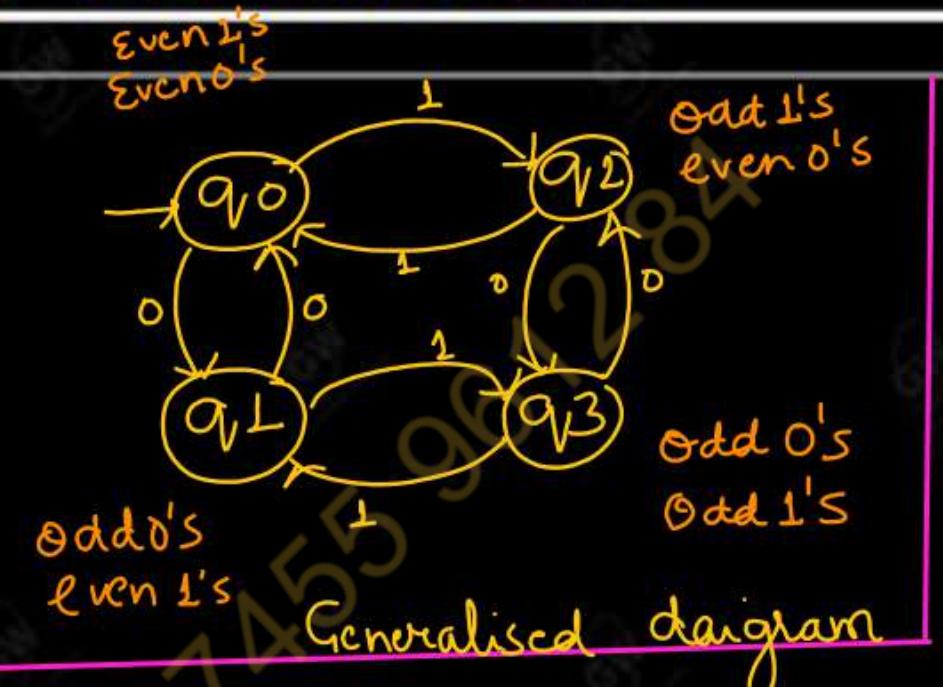
TODAY's TARGET

- DFA FOR TERNARY NUMBER
- DFA FOR EVEN AND ODD COMBINATION OF A AND B AND MANY MORE

By PRAGYA RAJVANSI

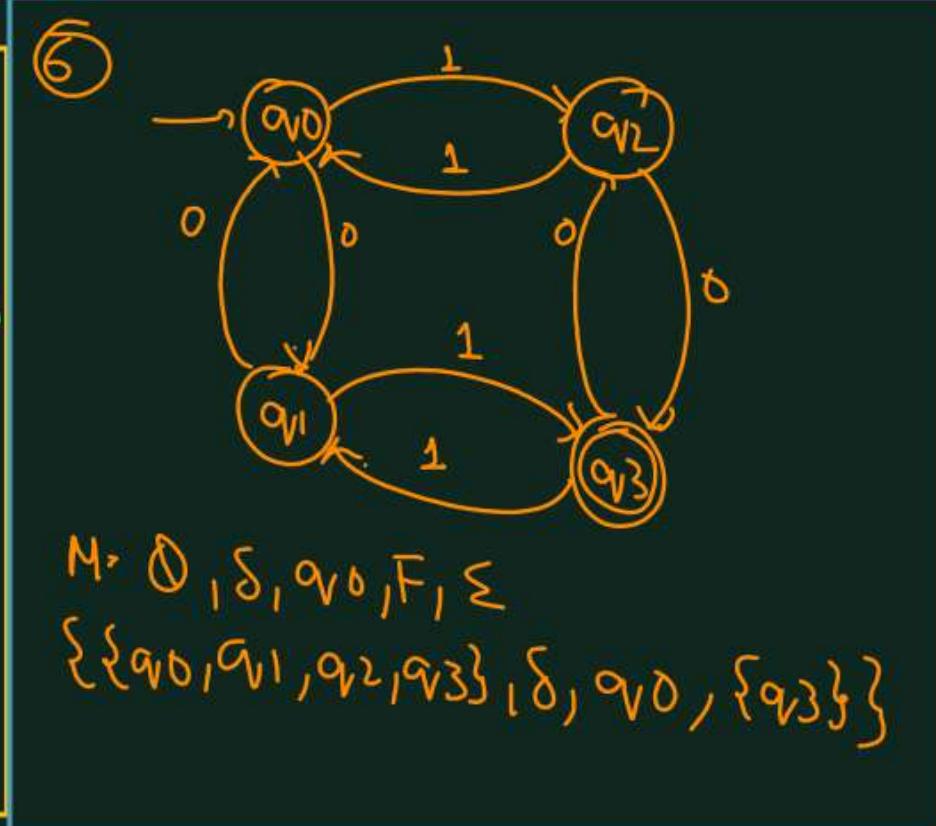
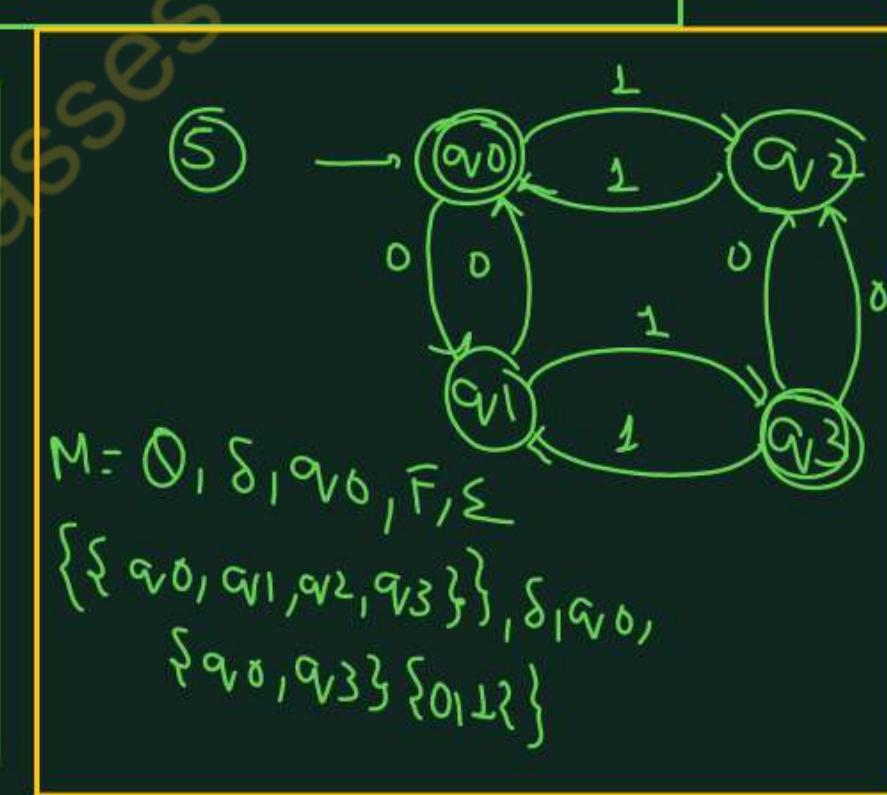
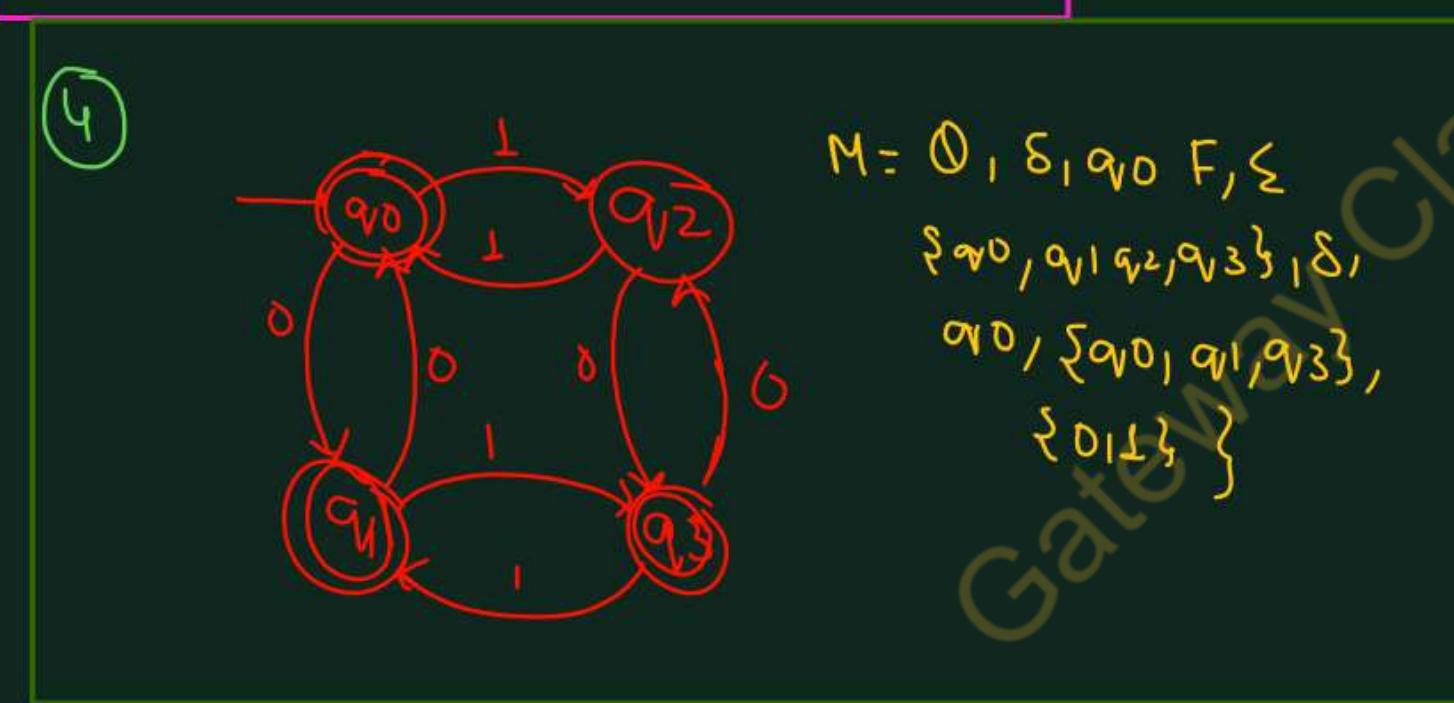
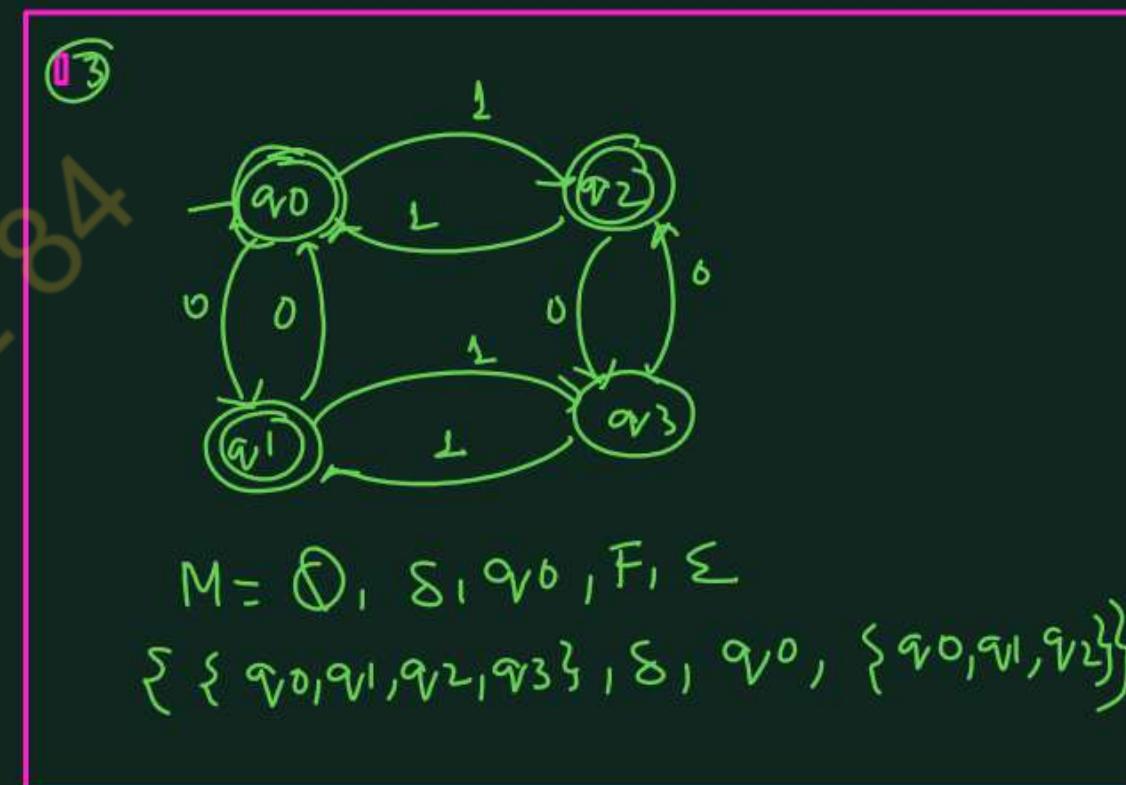
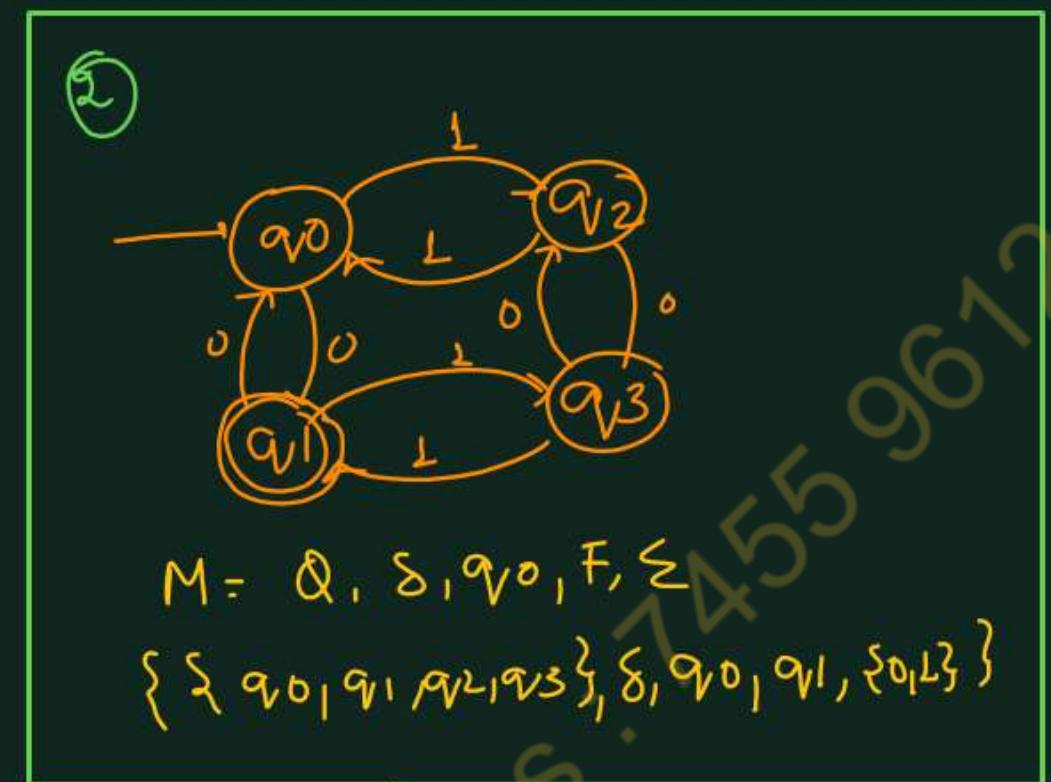
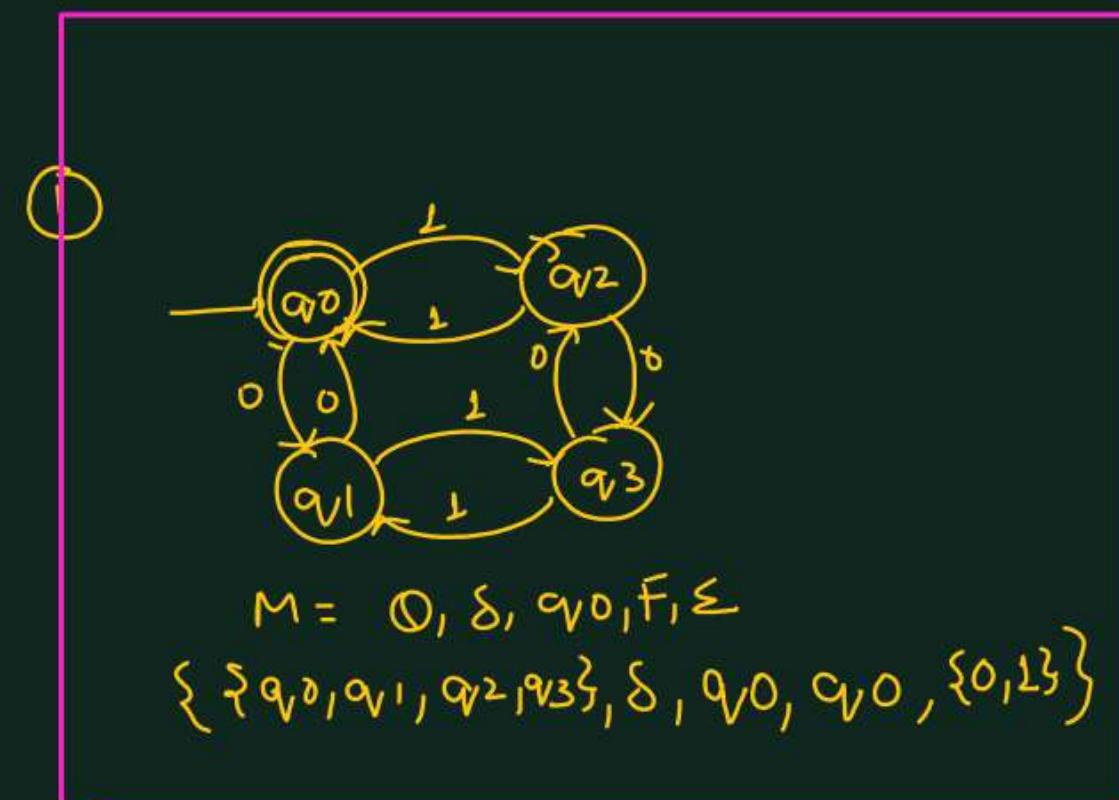
B.Tech, M.Tech(C.S.E)

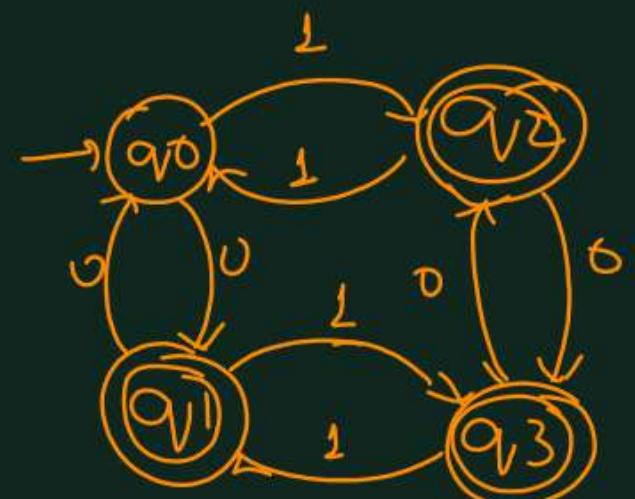
- Final*
- Even number of 0 and even number of 1 $\{q_0\}$
 - Odd number of 0 and even number of 1 $\{q_1\}$
 - Even number of 0 or even number of 1 $\{q_0, q_1, q_2\}$
 - Odd number of 0 or even number of 1 $\{q_0, q_1, q_3\}$
 - either odd number of 0 or even number of 1
But not both together $\{q_0, q_3\}$
 - odd number of 0 and odd number of 1 $\{q_3\}$
 - Odd number of 0 or odd number of 1 $\{q_1, q_2, q_3\}$



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



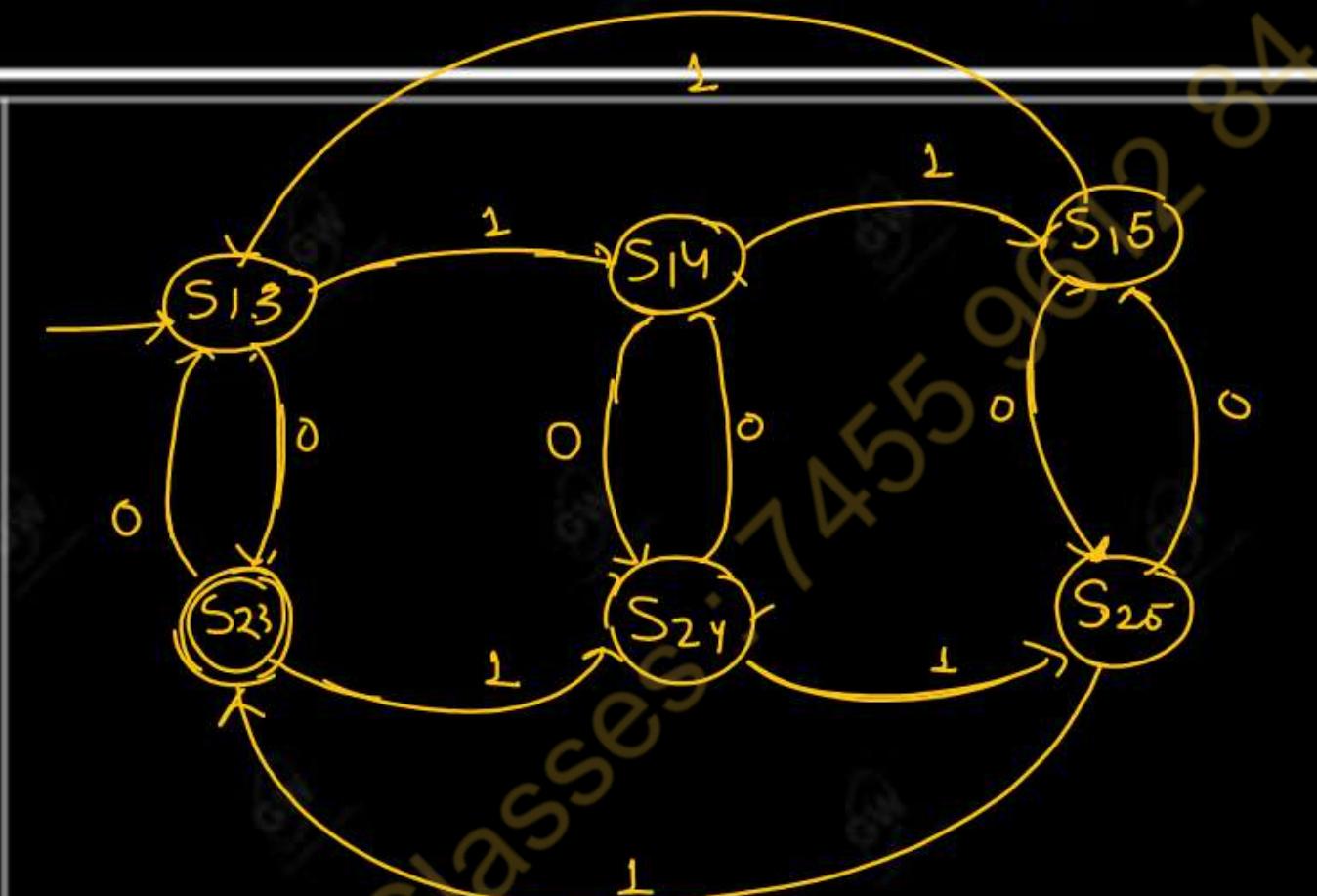
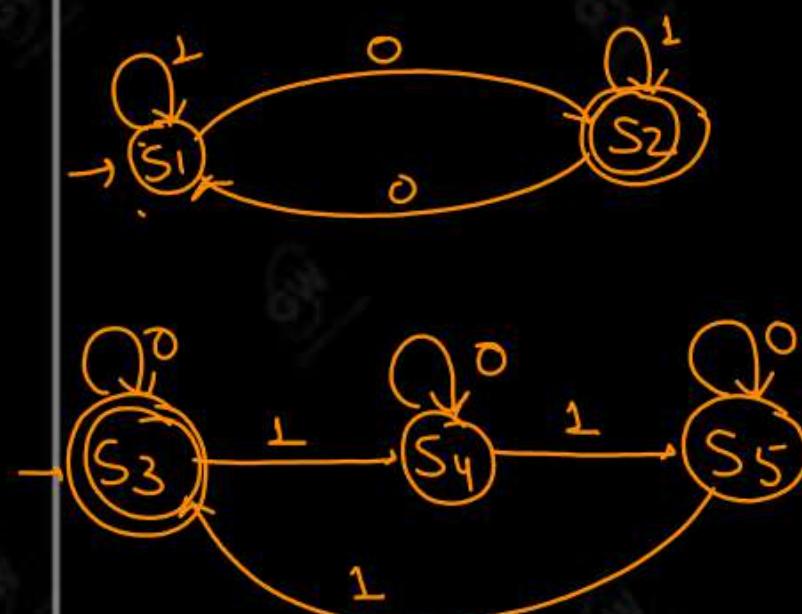


$M = \{Q, \Sigma, q_0, F, \delta\}$

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

Gateway Classes : 7455 9612 84

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}

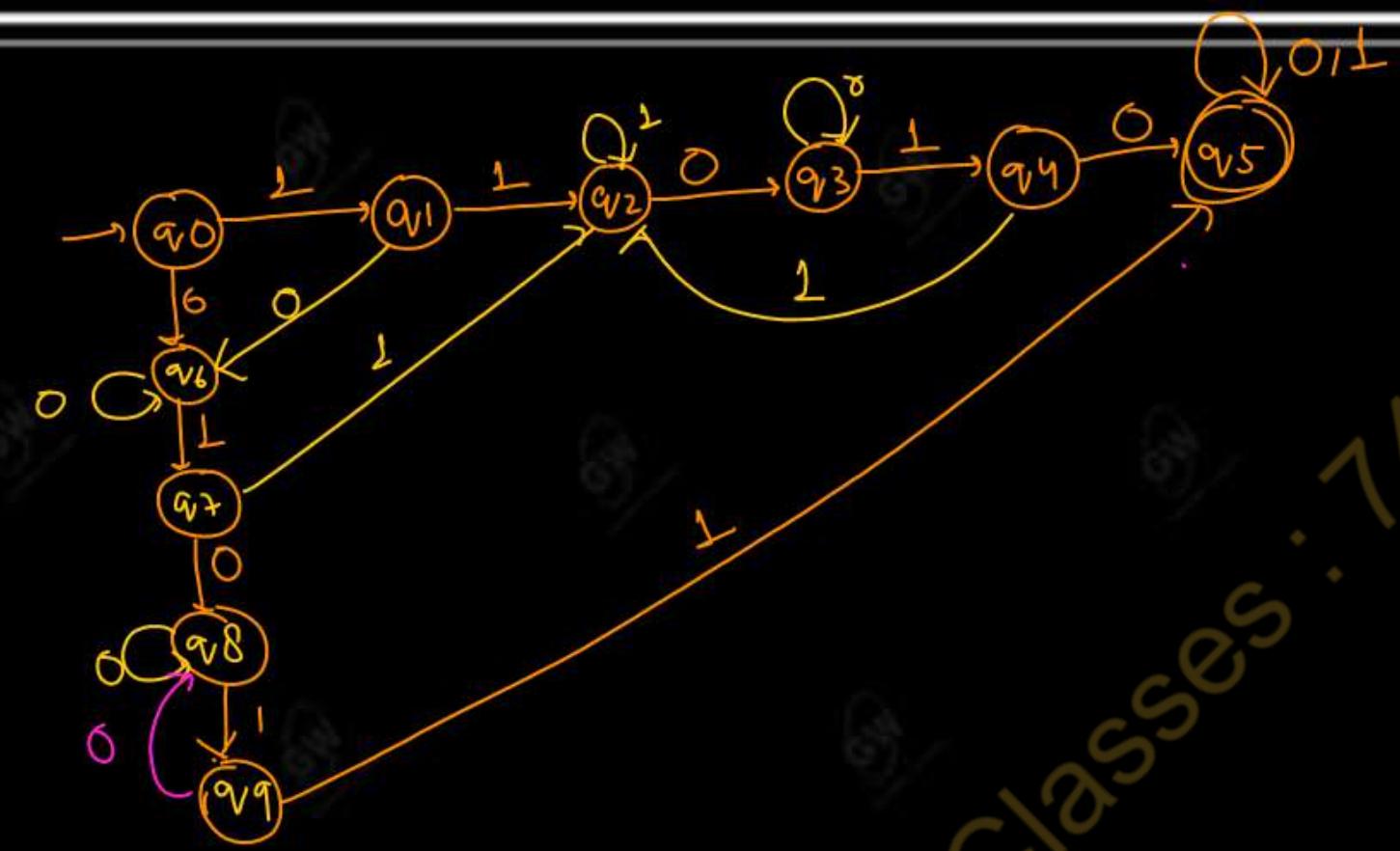
	0	1
$\rightarrow S_{13}$	S_{23} ✓	S_{14} ✓
S_{14}	S_{24} ✓	S_{15} ✓
S_{15}	S_{25} ✓	S_{13}
$*S_{23}$	S_{13} ✓	S_{24} ✓
S_{24}	S_{14} ✓	S_{25} ✓
S_{25}	S_{15} ✓	S_{23} ✓

{Q, Σ, δ, q₀, F}

{S₁₃, S₁₄, S₁₃, S₂₃, S₂₄, S₂₅}, {0,1}, δ, S₁₃, {S₂₃}

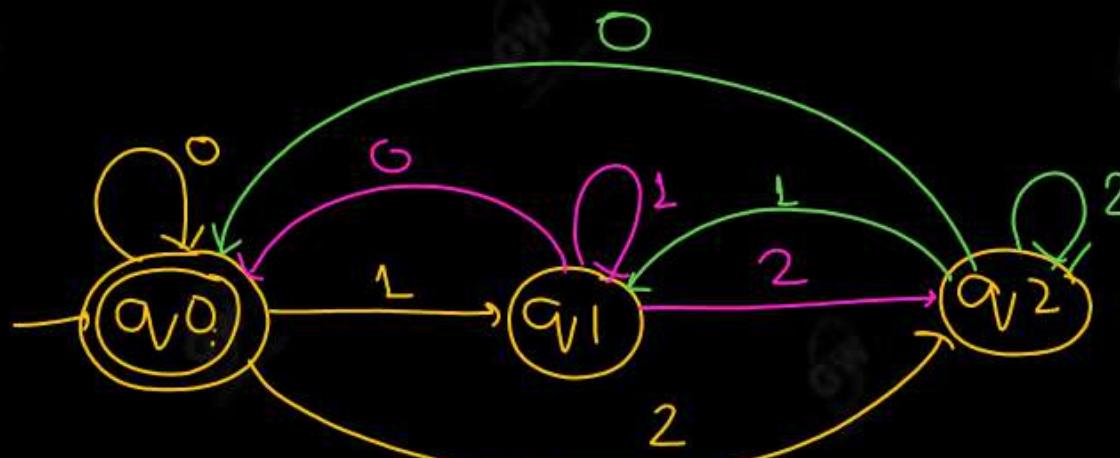
Construct the finite automata that accept all the string containing 11 and 010 as substring

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



Construct a DFA for ternary number divisible by 3

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



TRANSITION TABLE

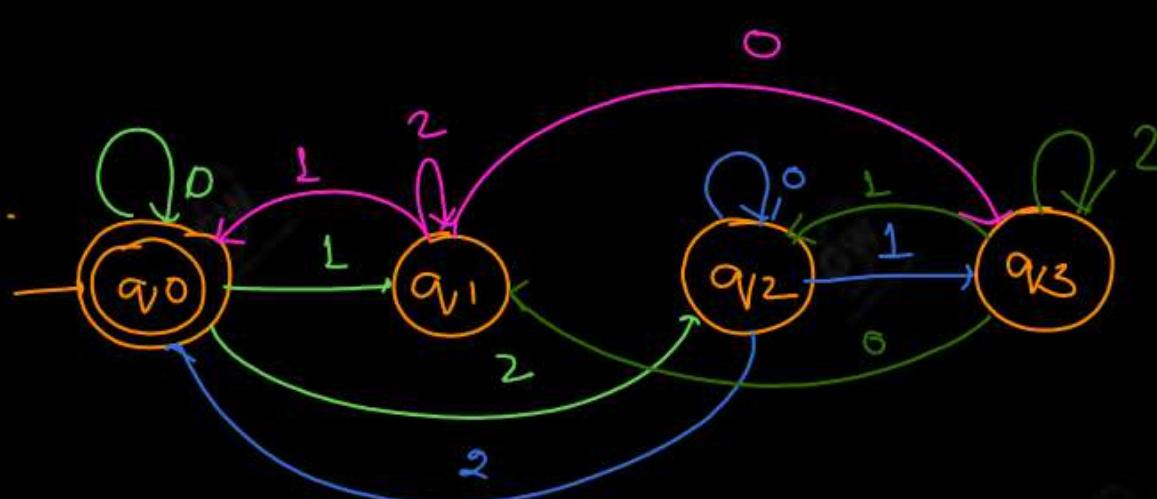
	0	1	2
$\rightarrow^* q_0$	q_0	q_1	q_2
q_1	q_0	q_1	q_2
q_2	q_0	q_1	q_2

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

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

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

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

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

Construct a DFA for ternary number divisible by 5

STATE TRANSITION DIAGRAM

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

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

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

BASIC CONCEPTS AND AUTOMATA THEORY

TODAY's TARGET

- NFA | NDFA
- Construct some NFA

By PRAGYA RAJVANSI
B.Tech, M.Tech(C.S.E)

- It is defined as a 5 tuple, $M=(Q, \Sigma, \delta, q_0, F)$

Q: Finite set of states

Σ : Finite set of the input symbol

q_0 : Initial state

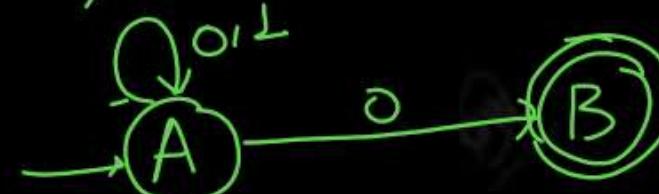
F: Final state $F \subseteq Q$

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 {0,1} accept all set of string end with zero



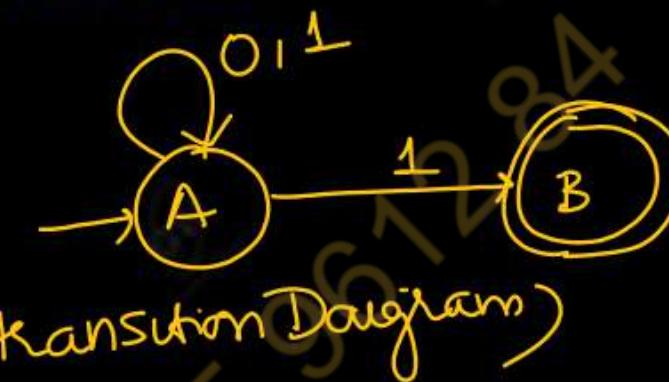
(transition diagram)

transition table

	0	1
$\rightarrow A$	{A,B}	A
$*B$	--	--

$$M = (Q, \Sigma, \delta, q_0, F) = \{\{A, B\}, \{0, 1\}, \delta, A, B\}$$

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



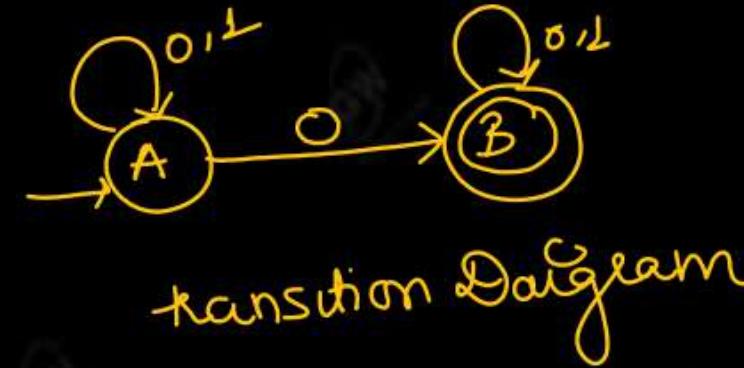
(transition Diagram)

transition table

	0	1
$\rightarrow A$	A	{A,B}
$*B$	\circledast	--

$$M = (Q, \Sigma, \delta, q_0, F) = \{\{A, B\}, \{0, 1\}, \delta, A, B\}$$

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



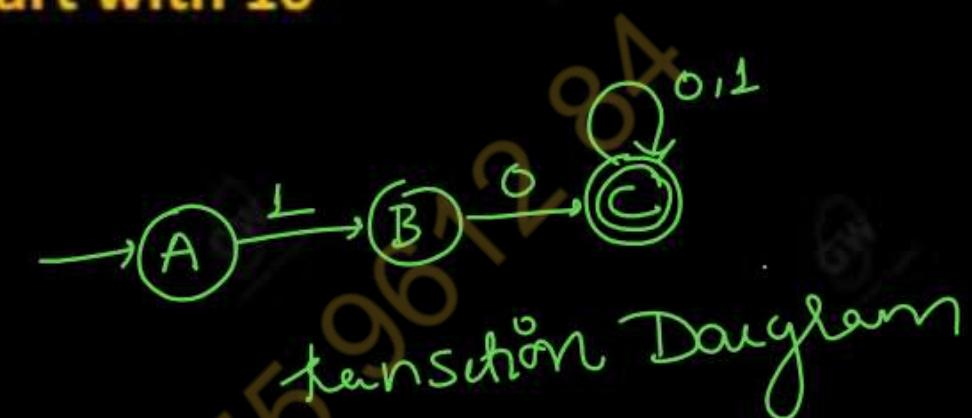
transition table

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

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

$$\{\{A, B\}, \{0, 1\}, \delta, A, B\}$$

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

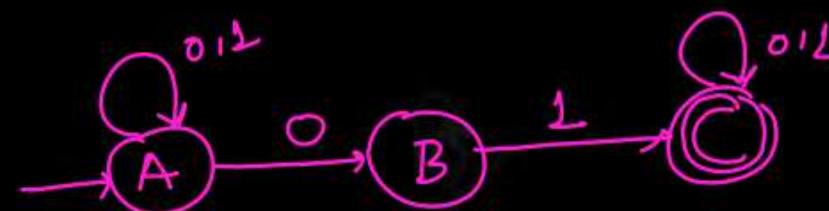


transition table

	0	1
$\rightarrow A$	--	B
B	C	--
*C	C	C

$$M = (Q, \Sigma, \delta, q_0, F) = \{\{A, B, C\}, \{0, 1\}, \delta, A, C\}$$

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



	0	1
$\rightarrow A$	{A,B}	A
B	--	C
*C	C	C

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

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

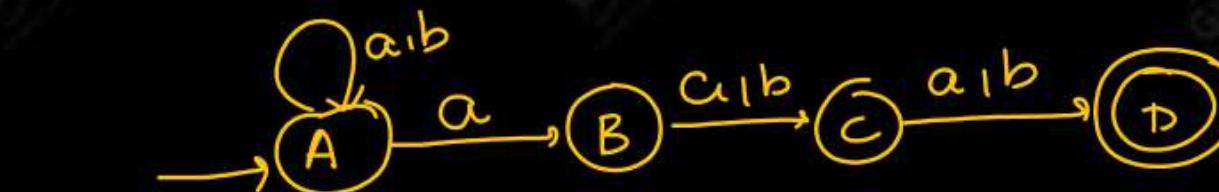
Construct the NFA over $\{0,1\}$ accept all set of string end with 11



	0	1
$\rightarrow A$	{A}	{A,B}
B	--	C
*C	--	--

$$M = (Q, \Sigma, \delta, q_0, F) = \{\{A,B,C\}, \{0,1\}, \delta, A, C\}$$

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 $\rightarrow 90-95\%$

③ Minimization of DFA $\rightarrow 90\%$

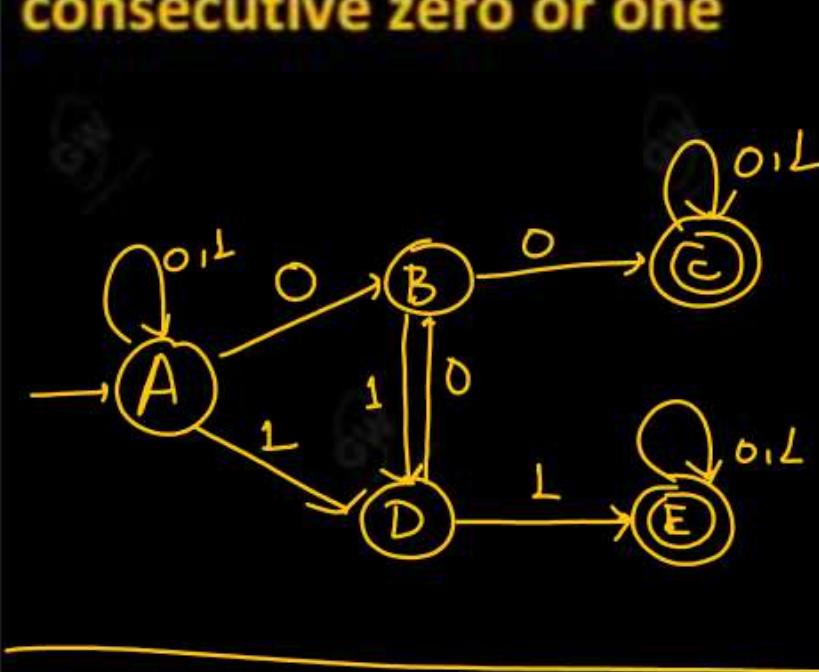
Construct NFA \rightarrow DFA (Minimization)
20-30%

TRANSITION TABLE

	a	b
a	{A,B}	A
b	C	C
c	D	D
*d	-	-

$$M = (Q, \Sigma, \delta, q_0, F) = \{\{A, B, C, D\}, \{0, 1\}, \delta, A, D\}$$

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



TRANSITION TABLE

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

$$M = (Q, \Sigma, \delta, q_0, F) = \{\{A, B, C, D, E\}, \{0, 1\}, \delta, A, \{E, C\}\}$$

BASIC CONCEPTS AND AUTOMATA THEORY

TODAY's TARGET

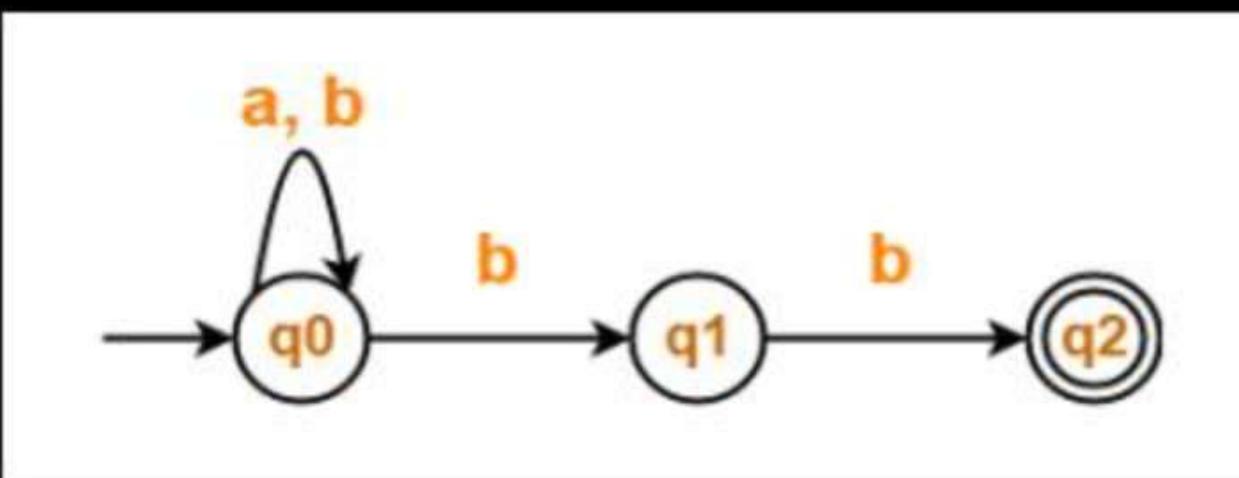
- NFA TO DFA CONVERSION PRACTICE PROBLEM

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E)

Gateway Classes . 745596284
PR

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



(NFA)

Step 1

TRANSITION TABLE OF NFA

State / Alphabet	a	b
$\rightarrow q_0$	q_0	q_0, q_1
q_1	-	$*q_2$
$*q_2$	-	-

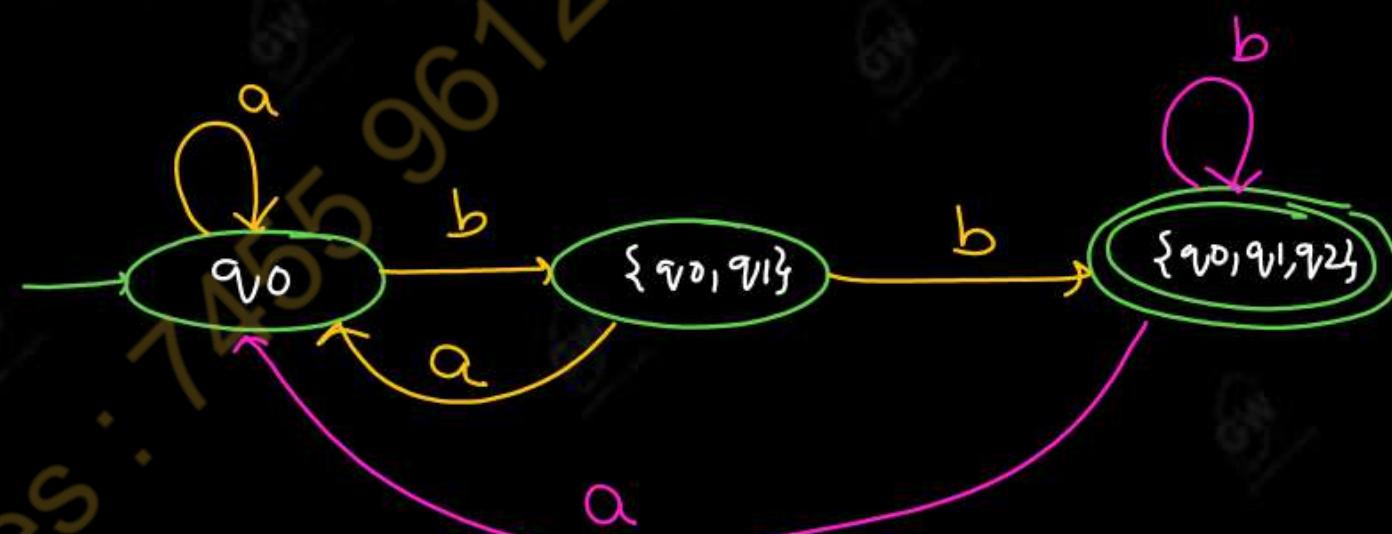
TRANSITION TABLE FOR DFA

State / Alphabet	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	$\{q_0, q_1, q_2\}$

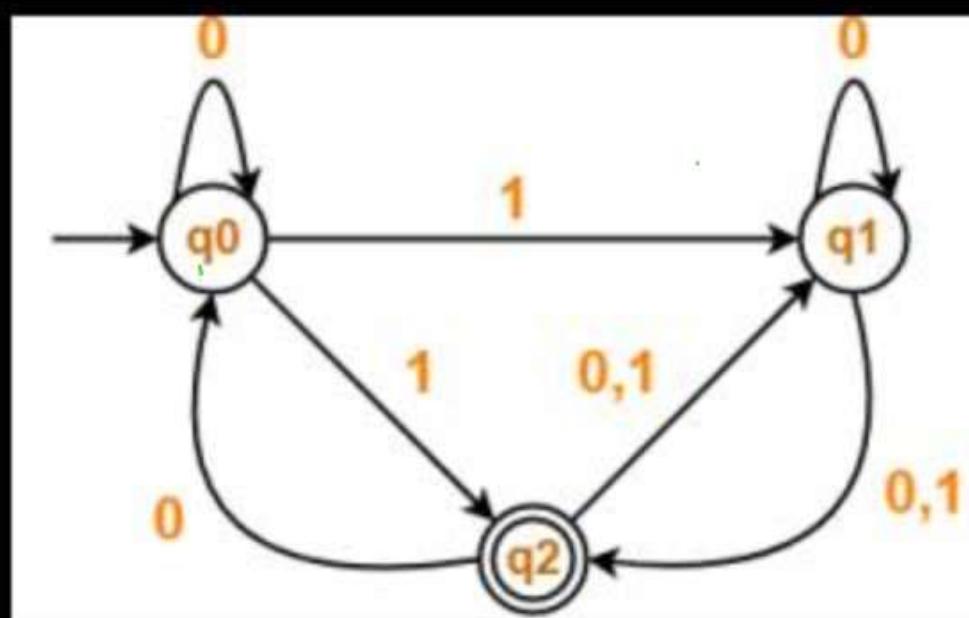
TRANSITION TABLE FOR DFA

State / Alphabet	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	$\{q_0, q_1, q_2\}$

TRANSITION DIAGRAM FOR DFA



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



(NFA)

TRANSITION TABLE OF NFA

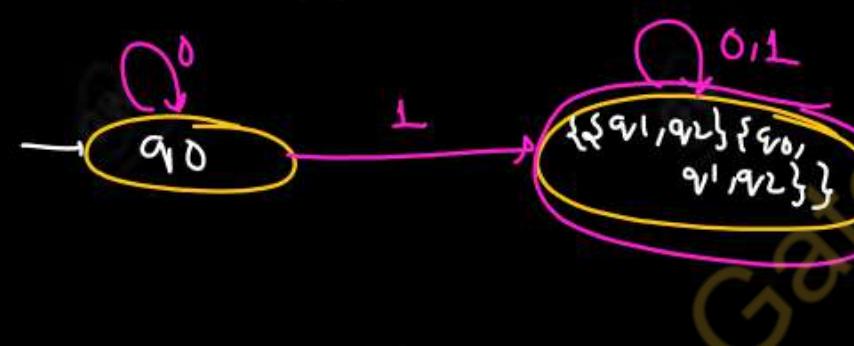
State / Alphabet	0	1
$\rightarrow q_0$	q_0	q_1, q_2
q_1	q_1, q_2	q_2
$*q_2$	q_0, q_1	q_1

TRANSITION TABLE FOR DFA

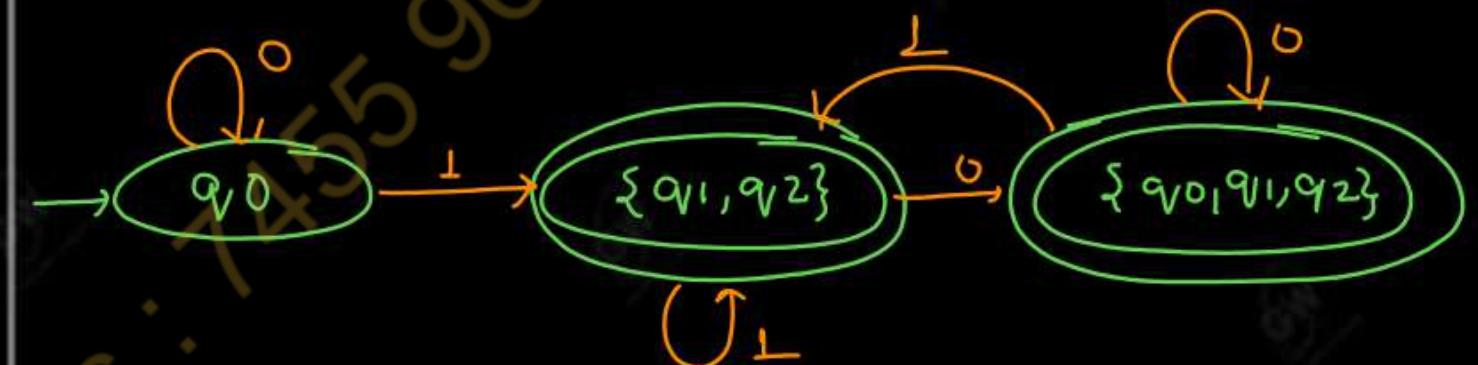
State / Alphabet	0	1
$\rightarrow q_0$	q_0	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$

TRANSITION TABLE FOR DFA

State / Alphabet	0	1
$\rightarrow q_0$	q_0	$\{q_1, q_2\}$
$*\{q_1, q_2\}$	$*\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$



TRANSITION DIAGRAM FOR DFA

Minimization DFA ($NFA \rightarrow M_{DFA}$)

$$\bar{\pi}_0 = \{q_0\} \cup \{q_1, q_2\}, \{q_0, q_1, q_2\}$$

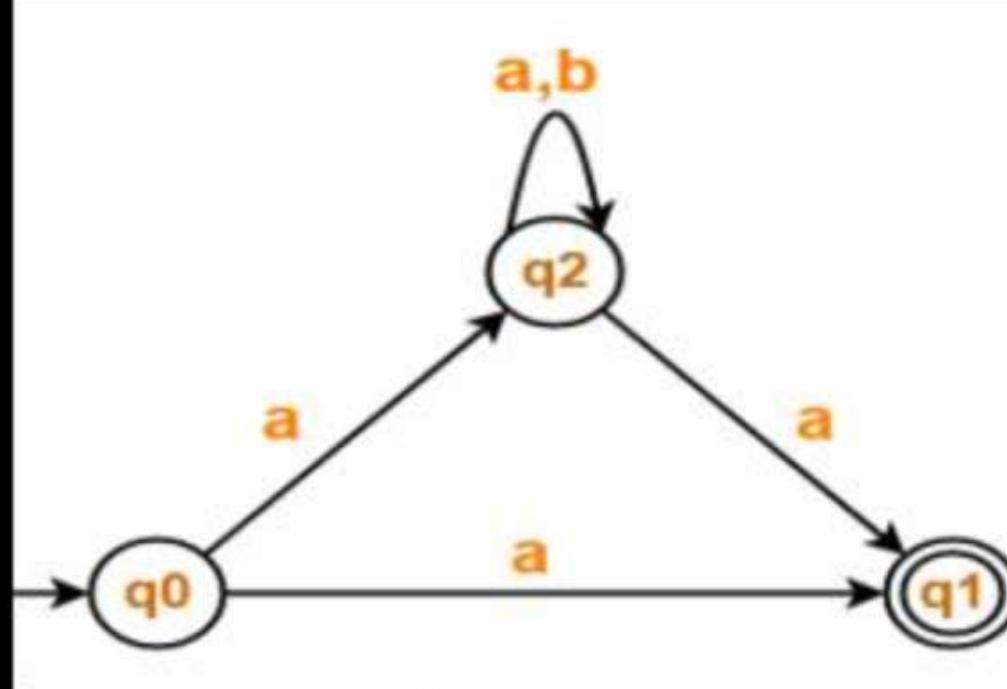
$$\bar{\pi}_1 = \{q_0\} \cup \{q_1\}, \{q_0, q_1, q_2\}$$

- ① $NFA \rightarrow DFA$
 $DFA \rightarrow M_{DFA}$

(MDFA)

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

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



(NFA)

TRANSITION TABLE OF NFA

State / Alphabet	a	b
$\rightarrow q_0$	q_1, q_2	-
$*q_1$	-	-
q_2	q_1, q_2	q_2

TRANSITION TABLE FOR DFA

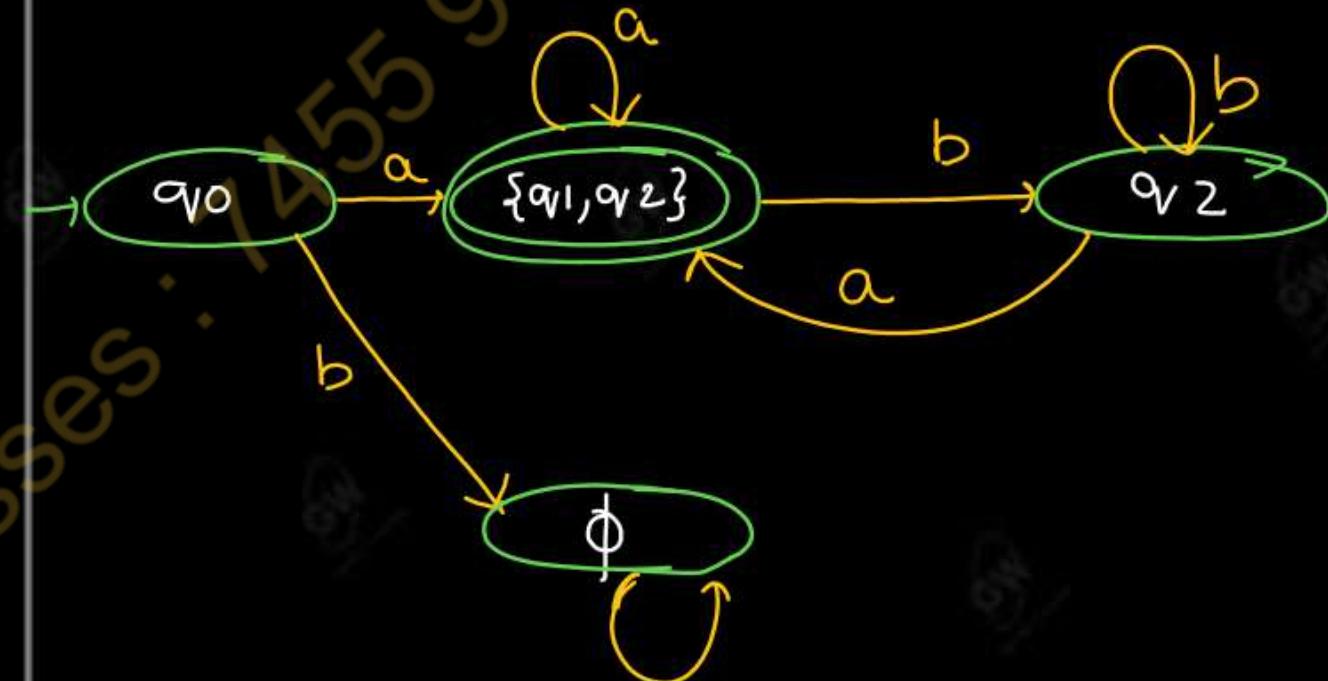
State / Alphabet	a	b
$\rightarrow q_0$	$\{q_1, q_2\}$	\emptyset
$\{q_1, q_2\}$	$\{q_1, q_2\}$	q_2
q_2	$\{q_1, q_2\}$	q_2
\emptyset	\emptyset	\emptyset

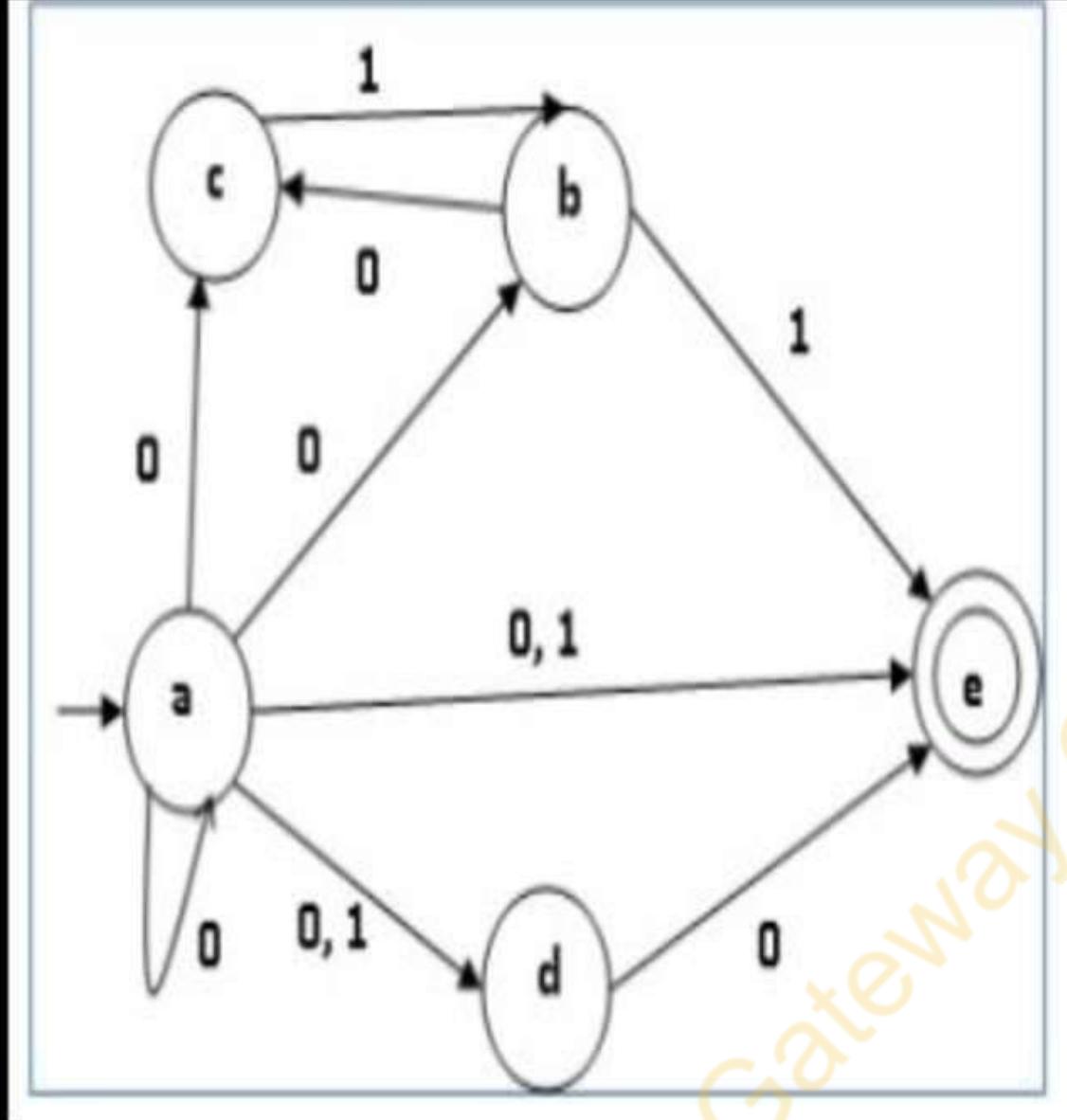
\emptyset (Dead State)

TRANSITION TABLE FOR DFA

State / Alphabet	a	b
$\rightarrow q_0$	$\{q_1, q_2\}$	\emptyset
$*\{q_1, q_2\}$	$\{q_1, q_2\}$	q_2
q_2	$\{q_1, q_2\}$	q_2
\emptyset	\emptyset	\emptyset

TRANSITION DIAGRAM FOR DFA





(NFA)

TRANSITION TABLE FOR NDFA

State / Alphabet	0	1
$\rightarrow a$	{a,b,c,d,e}	{d,e}
b	{c}	{e}
c	\emptyset	{b}
d	{e}	\emptyset
*e	\emptyset	\emptyset

Transition table for DFA

State / Alphabet	0	1
$\rightarrow a$	$\{a, b, c, d, e\}$	$\{\bar{d}, \bar{e}\}$
* $\{a, b, c, d, e\}$	$\{a, b, c, d, e\}$	$\{b, \bar{d}, \bar{e}\}$
* $\{\bar{d}, \bar{e}\}$	$\{\bar{e}\}$	\emptyset
* $\{b, \bar{d}, \bar{e}\}$	$\{c, \bar{e}\}$	$\{\bar{e}\}$
* $\{\bar{e}\}$	\emptyset	$\{\bar{e}\}$
* $\{c, \bar{e}\}$	\emptyset	$\{\bar{b}\}$
$\{\bar{b}\}$	$\{c\}$	$\{\bar{e}\}$
$\{c\}$	\emptyset	$\{\bar{b}\}$
\emptyset	\emptyset	\emptyset

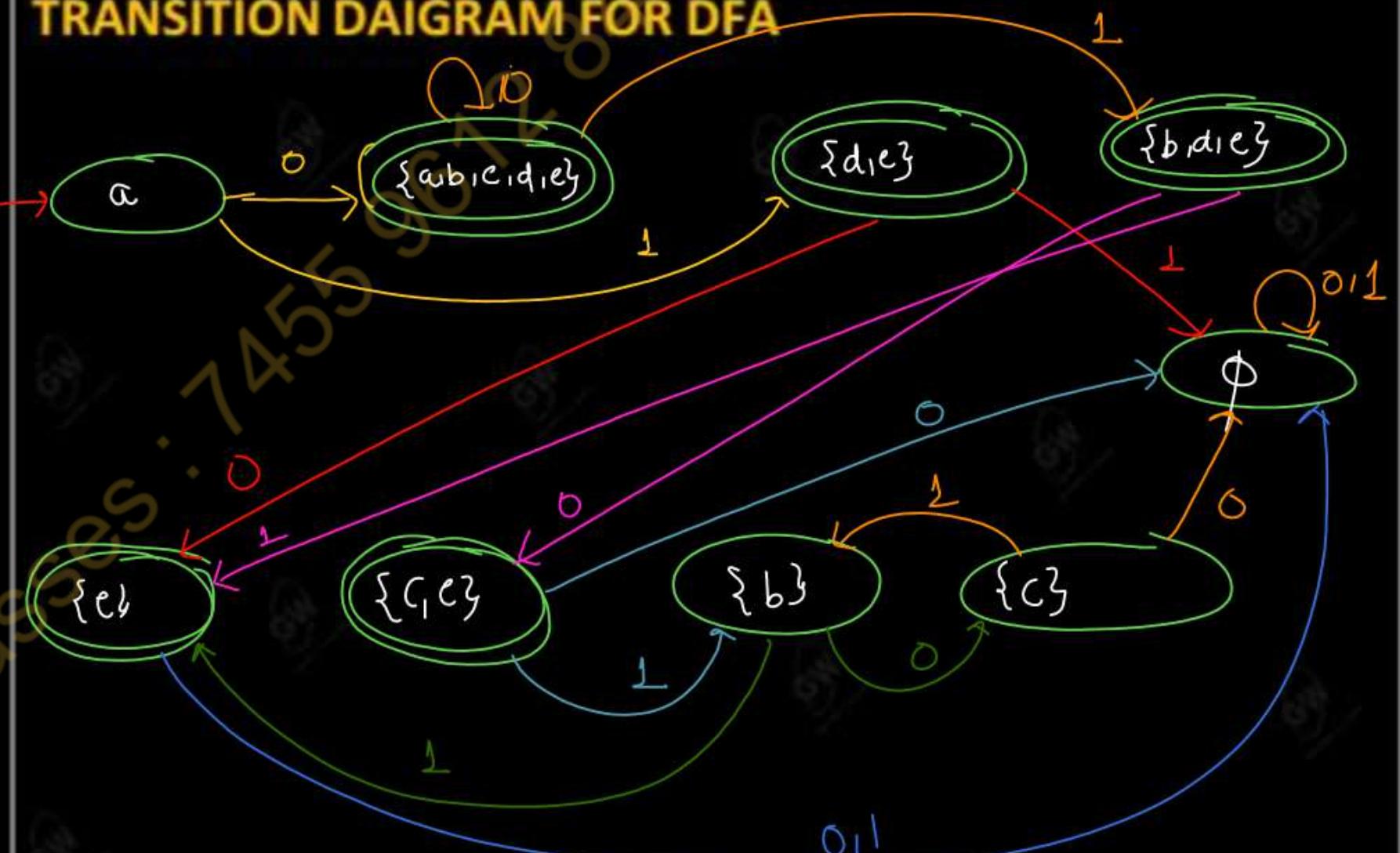
TRANSITION TABLE FOR NDFA

State / Alphabet	0	1
$\rightarrow a$	$\{a, b, c, d, e\}$	$\{d, e\}$
b	$\{c\}$	$\{e\}$
c	\emptyset	$\{b\}$
d	$\{e\}$	\emptyset
*e	\emptyset	\emptyset

TRANSITION TABLE FOR DFA

	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}	\emptyset
*{b,d,e}	{c,e}	{e}
*{e}	\emptyset	\emptyset
*{c, e}	\emptyset	{b}
{b}	{c}	{e}
{c}	\emptyset	{b}
\emptyset	\emptyset	\emptyset

NFA $\xrightarrow{\text{NFA to DFA}}$
 DFA $\xrightarrow{\text{DFA to MDFA}}$
 MDFA $\xrightarrow{\text{MDFA to DFA}}$

TRANSITION DIAGRAM FOR DFA

BASIC CONCEPTS AND AUTOMATA THEORY

TODAY's TARGET

ϵ -NFA TO NFA CONVERSION PRACTICE PROBLEM

By PRAGYA RAJVANSI

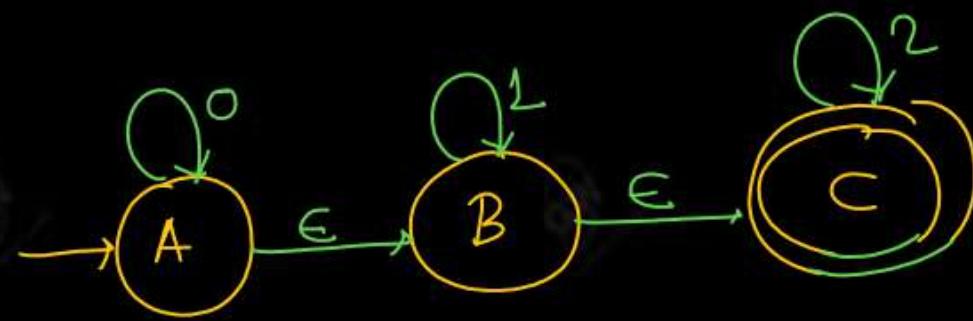
B.Tech, M.Tech(C.S.E)

Gateway Classes 745596284

- Topic*
- ① DFA
 - ② Minimization of DFA
 - ③ NFA to DFA
 - ④ ϵ -NFA to NFA

- Algorithms*
- 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
 - ⑦ ϵ -NFA to Minimized DFA
 - ⑧ ϵ -NFA to NFA
 - NFA to DFA
 - DFA to Minimized DFA

Q1 EPSILON NFA TO NFA CONVERSION



$$\begin{aligned} \text{E-closure}(\delta(\text{E-closure}(A), 0)) \\ \text{E-closure}(\delta((A, B, C), 0)) \\ (\delta(A, 0) \cup \delta(B, 0) \cup \delta(C, 0)) \\ \text{E-closure}(A \cup \emptyset \cup \emptyset) \end{aligned}$$

TRANSITION TABLE OF ϵ NFA

State / Alphab et	0	1	2	ϵ
$\rightarrow A$	A	-	-	B
B	-	B	-	C
*C	-	-	C	-

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

Step 1

- epsilon closure A = {A, B, C}
- $\delta(A, 0) = A$
- $\delta(B, 0) = \emptyset$
- $\delta(C, 1) = \emptyset$

	ϵ	0	ϵ
ϵ closure	A	A	A, B, C
A	A	-	-
B	B	-	-
C	C	-	-

$A \cup \emptyset \cup \emptyset = \{A\}$

epsilon closure(A) = {A, B, C}

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

Q1 EPSILON NFA TO NFA CONVERSION

State / Alphab et	0	1	2	ϵ
$\rightarrow A$	A	-	-	B
B	-	B	-	C
*C	-	-	C	-

Gateway Classes : 7455961234

ϵ	1	ϵ
A	$A \xrightarrow{1} -$ $B \xrightarrow{1} B$ $C \xrightarrow{1} -$	B, C

ϵ	2	ϵ
A	A B C	-

epsilon closure (A)
 $\{A, B, C\}$

$$\delta(A, 1) = \emptyset$$

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

$$\delta(C, 1) = \emptyset$$

$$\emptyset \cup B \cup \emptyset$$

$$\{B\}$$

epsilon closure (B)
 $= B, C$

epsilon closure (A)
 A, B, C

$$\delta(A, 2) = \emptyset$$

$$\delta(B, 2) = \emptyset$$

$$\delta(C, 2) = C$$

$$\{\emptyset \cup \emptyset \cup C\}$$

epsilon (C) = C

Q1 EPSILON NFA TO NFA CONVERSION

State / Alphab et	0	1	2	ϵ
$\rightarrow A$	A	-	-	B
B	-	B	-	C
*C	-	-	C	-

	ϵ	0	ϵ
B	B C	\emptyset	-

Q1
84
96
95
74

	ϵ	1	ϵ
B	$B \xrightarrow{1} B \xrightarrow{\epsilon} B, C$ $C \xrightarrow{1} -$	-	-

Closure

	ϵ	2	ϵ
B	$B \xrightarrow{2} -$ $C \xrightarrow{2} C$	-	-

Q1 EPSILON NFA TO NFA CONVERSION

State / Alphab et	0	1	2	ϵ
$\rightarrow A$	A	-	-	B
B	-	B	-	C
*C	-	-	C	-

	ϵ	0	ϵ
C	C	ϕ	

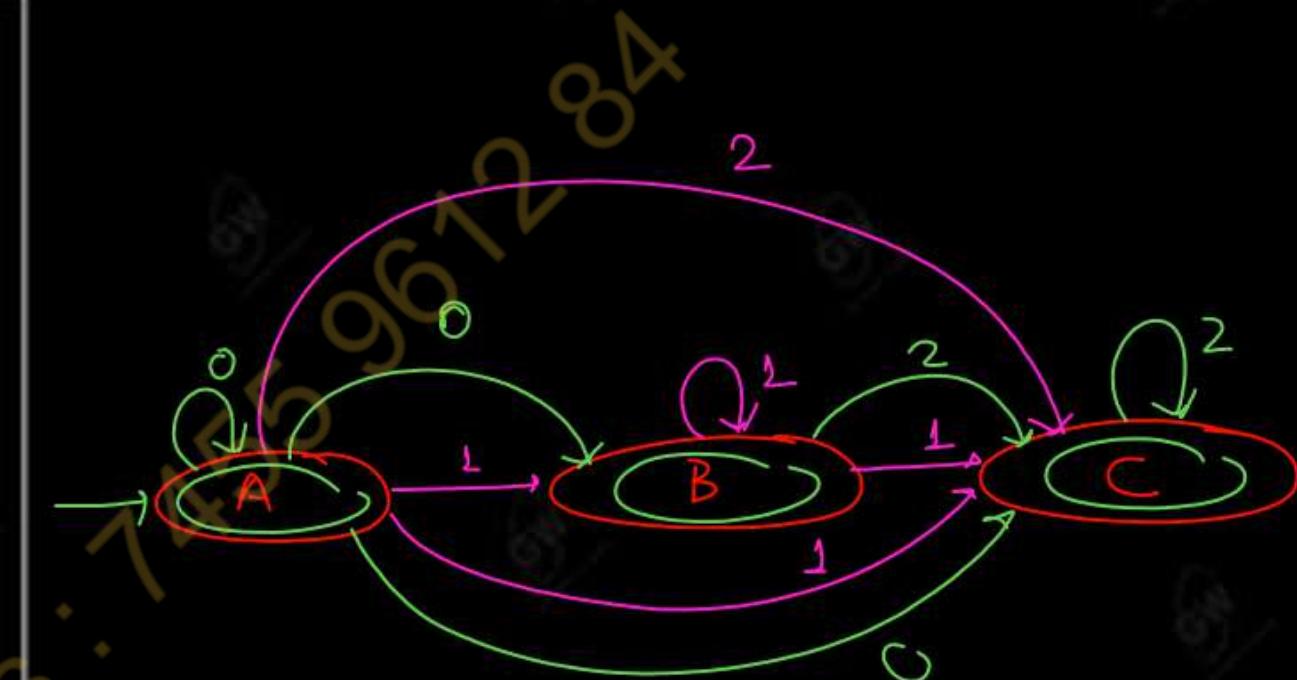
	ϵ	1	ϵ
C	C	ϕ	-

	ϵ	2	ϵ
B	C	C	C

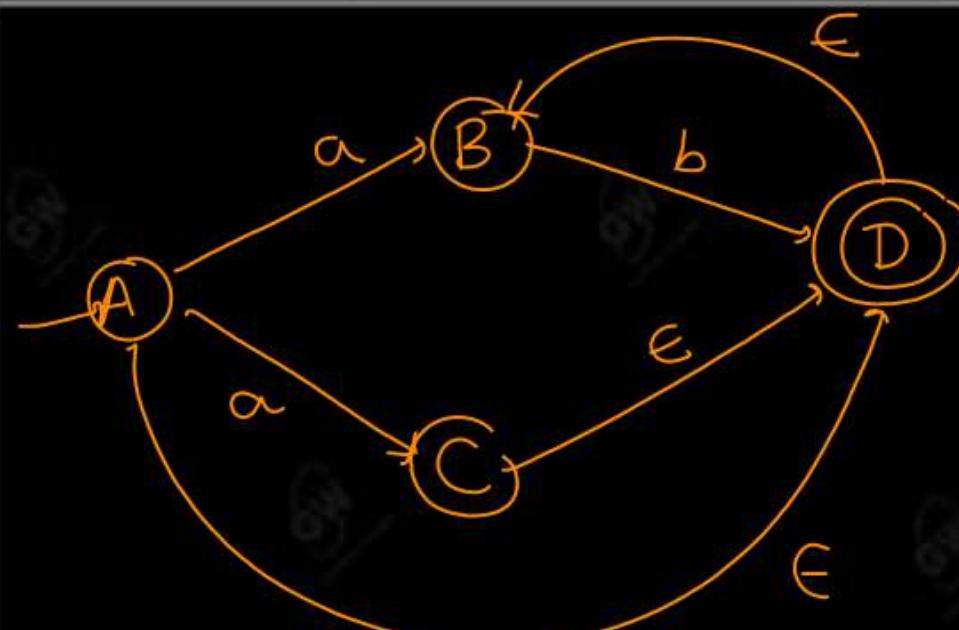
Q1 EPSILON NFA TO NFA CONVERSION

State / Alphabet	0	1	2
$\rightarrow^* A$	A, B, C	B, C	C
$* B$	-	B, C	C
$* C$	-	-	C

(NFA) transition table



Q2 EPSILON NFA TO NFA CONVERSION



	a	b	ϵ
$\rightarrow A$	{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 / Alphab et	a	b	ϵ
$\checkmark \rightarrow A$	{B,C}	-	D
B	-	D	-
C	-	-	D
*D	-	-	B

	ϵ	a	ϵ
A	$A \xrightarrow{a} B, C \rightarrow B, C, D$ $D \xrightarrow{a} -$ $B \xrightarrow{a} -$	-	-

Q4
Date: 14/05/2024

ϵ^*	b	ϵ^*
A	$A \xrightarrow{b} \emptyset$ $D \xrightarrow{b} \emptyset$ $B \xrightarrow{b} D$	B, D

Closure

ϵ^*	a	ϵ^*
B	$B \xrightarrow{a} \emptyset$	-

Q2 EPSILON NFA TO NFA CONVERSION

State / Alphabet	a	b	ϵ
$\rightarrow A$	{B,C}	-	D
B	-	D	-
C	-	-	D
*D	-	-	B

	ϵ	b	ϵ
B	$B \xrightarrow{b} D \xrightarrow{\epsilon} B_{ID}$		

1455 9612 94

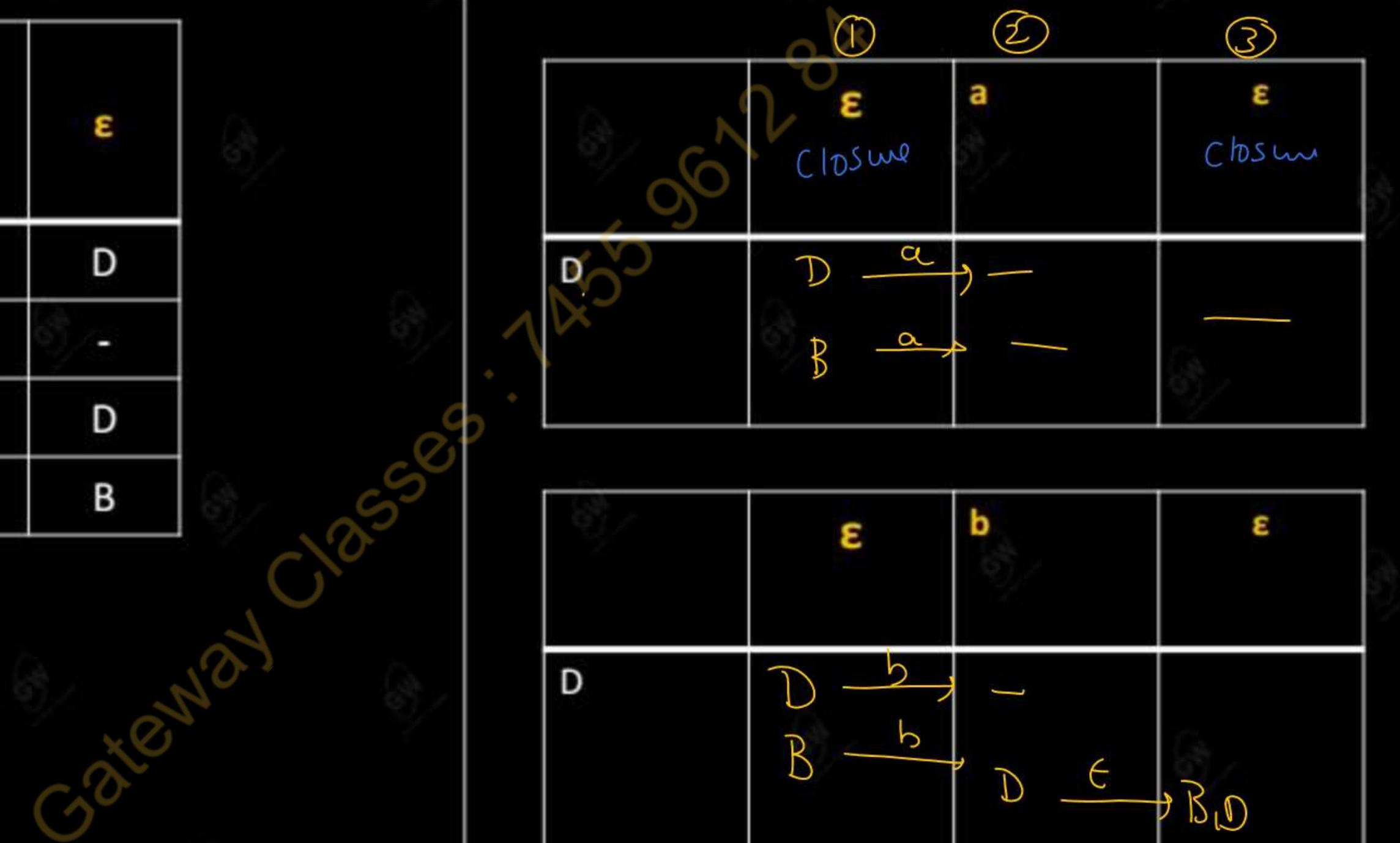
ϵ	a	ϵ
Closure		
C	$C \xrightarrow{a} \emptyset$	
D	$D \xrightarrow{a} \emptyset$	
B	$B \xrightarrow{a} \emptyset$	

Closure

ϵ	b	ϵ
Closure		
C	$C \xrightarrow{b} \emptyset$	
D	$D \xrightarrow{b} \emptyset$	
B	$B \xrightarrow{b} D \xrightarrow{\epsilon} D_{IB}$	

Q2 EPSILON NFA TO NFA CONVERSION

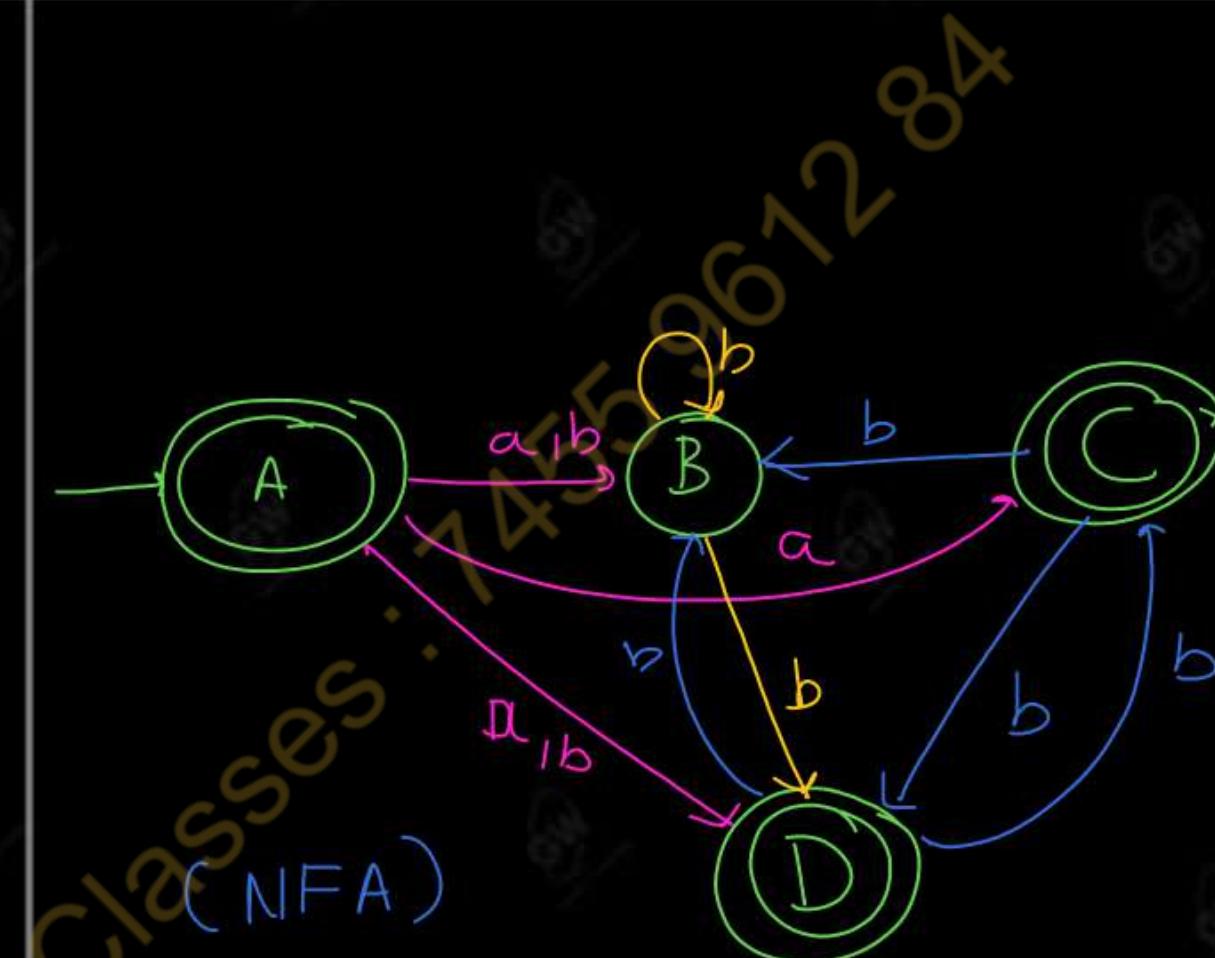
State / Alphab et	a	b	ϵ
$\rightarrow A$	{B,C}	-	D
B	-	D	-
C	-	-	D
*D	-	-	B



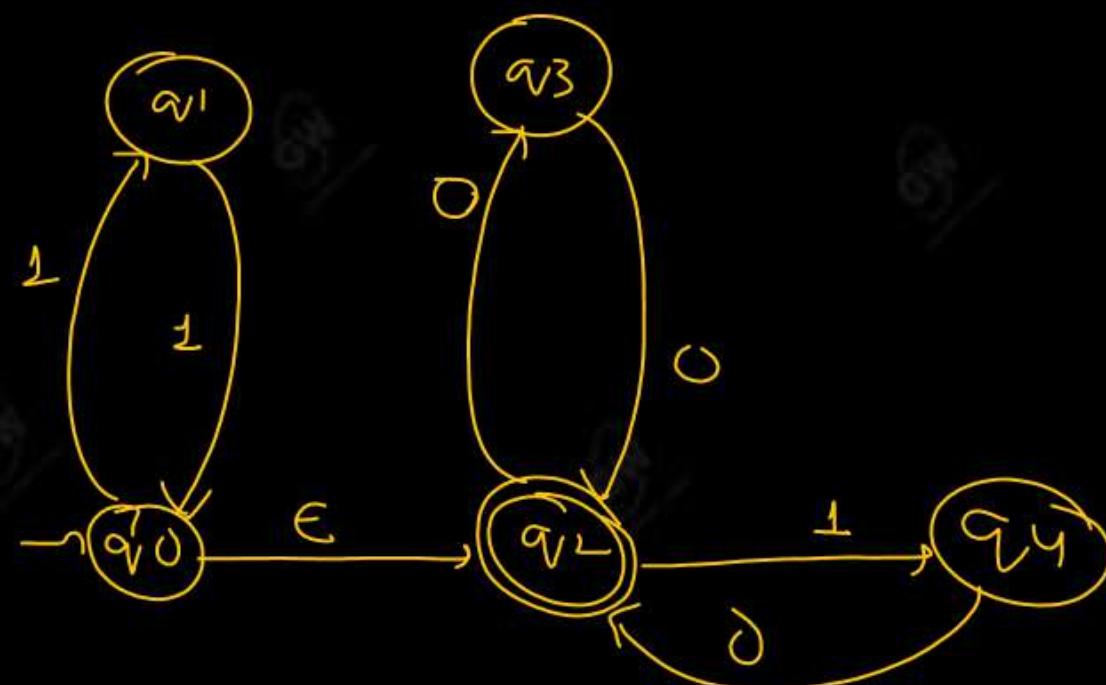
Q2 EPSILON NFA TO NFA CONVERSION

State / Alphab et	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
0	-	q_1	q_2
1	-	q_0	-
q_0	-	q_1	q_2
q_1	-	q_0	-
$*q_2$	q_3	q_4	-
q_3	q_2	-	-
q_4	q_2	-	-

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
q_0	q_{v0} q_{v2}	- q_{v3}	$q_{v3} \rightarrow q_{v3}$

	ϵ Closure	1	ϵ Closure
q_0	q_{v0} q_{v2}	q_{v1} q_{v4}	q_{v1}, q_{v4}

	0	1	ϵ
q_0	- $q_{v1}, q_{v5}, q_{v6}, q_{v12}, q_{v4}$	q_{v1} q_{v2}	q_{v2}
q_1	- q_{v3}	q_{v0}	- q_{v3}
$*q_2$	- q_{v3}	q_{v3} q_{v4}	- q_{v3}
q_3	- q_{v2}	q_{v2}	- q_{v2}
q_4	- q_{v2}	- q_{v2}	- q_{v2}

Q3 EPSILON NFA TO NFA CONVERSION

	ϵ	0	ϵ
q_1	$q_V 1$	\emptyset	-

	ϵ	1	ϵ
q_1	$q_V 1$	$\xrightarrow{+} q_V 0$	Closure $\{q_V 0, q_V 2\}$

	0	1	ϵ
q_0	-	q_1	q_2
q_1	-	q_0	-
* q_2	q_3	q_4	-
q_3	q_2	-	-
q_4	q_2	-	-

Q3 EPSILON NFA TO NFA CONVERSION

	ϵ	0	ϵ Closure
q_2	$q_{12} \xrightarrow{0} q_{13}$	q_{13}	

	ϵ	1	ϵ Closure
q_2	$q_{12} \xrightarrow{1} q_{14}$	q_{14}	q_{11}

	0	1	ϵ
q_0	-	q_1	q_2
q_1	-	q_0	-
$*q_2$	q_3	q_4	-
q_3	q_2	-	-
q_4	q_2	-	-

Q3 EPSILON NFA TO NFA CONVERSION

	ϵ	0	ϵ
q3	q_1^3	q_1^2	q_1^2

	ϵ	1	ϵ
q3	q_3	—	—

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

Q3 EPSILON NFA TO NFA CONVERSION

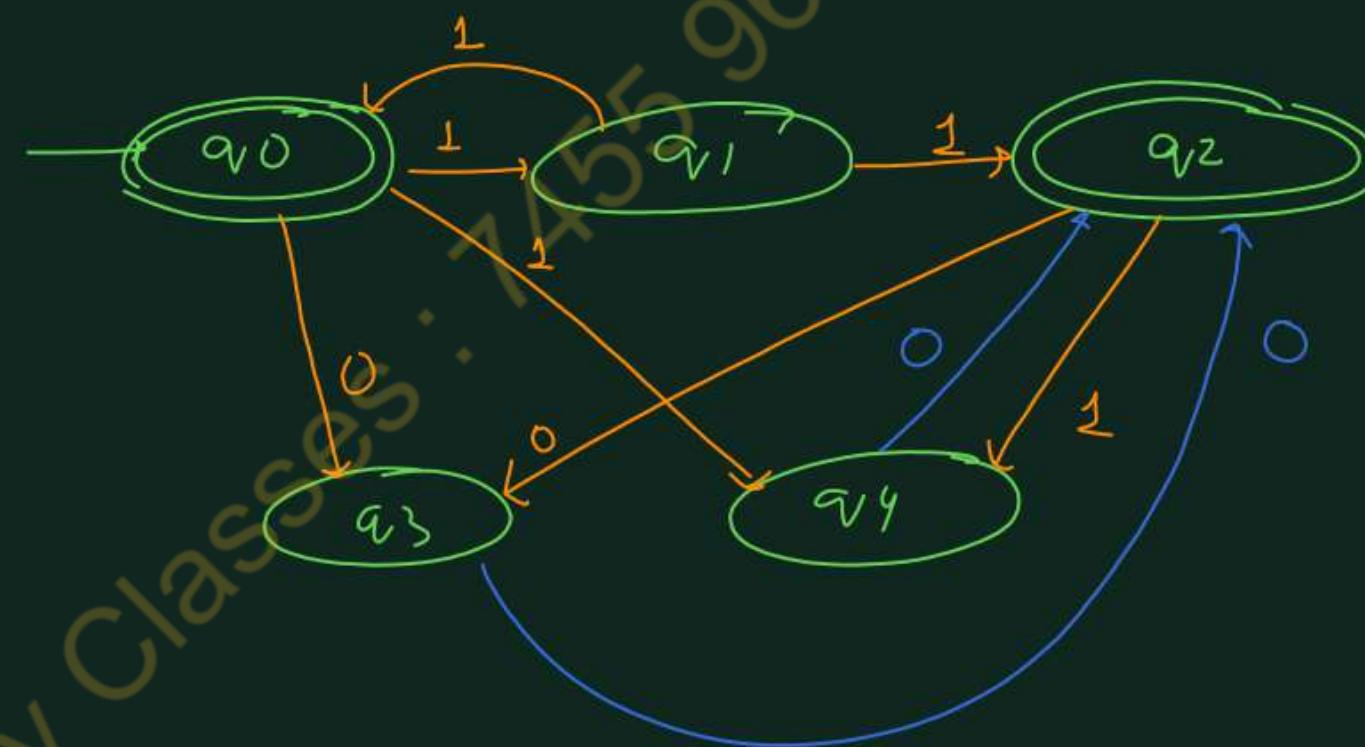
	ϵ	0	ϵ
q_4	q_4	q_2	q_2

	ϵ	1	ϵ
q_4	q_4	-	q_4

	0	1	ϵ
q_0	-	q_1	q_2
q_1	-	q_0	-
* q_2	q_3	q_4	-
q_3	q_2	-	-
q_4	q_2	-	-

NFA

	0	1
q_0	q_3	q_1, q_4
q_1	-	q_0, q_2
q_2	q_3	q_4
q_3	q_2	-
q_4	q_1	-



BASIC CONCEPTS AND AUTOMATA THEORY

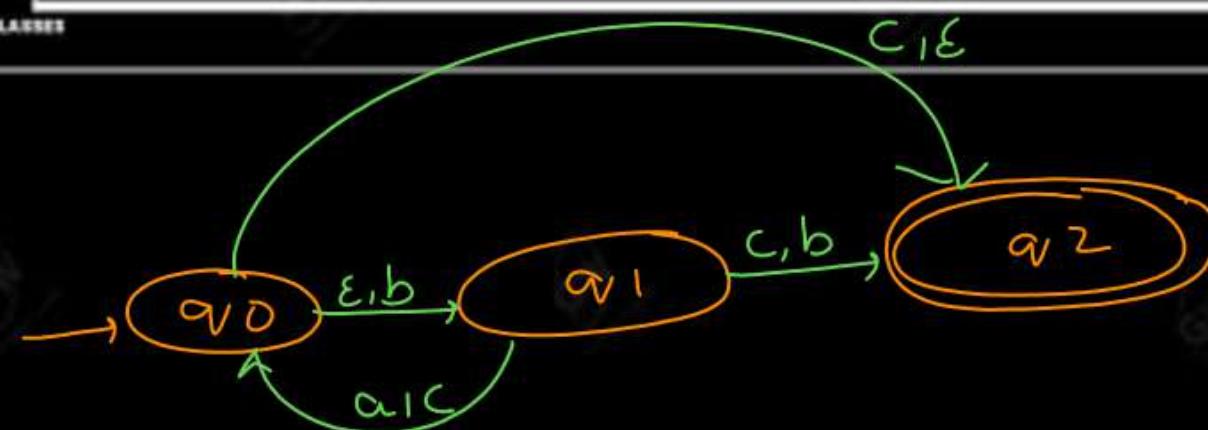
TODAY's TARGET

- EPSILON NFA TO Minimization DFA
- CONVERSION PRACTICE PROBLEM
- NFA TO Minimization DFA CONVERSION
- PRACTICE PROBLEM

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E)

Convert following EPSILON NFA to equivalent MDFA



(transition diagram)
for ϵ -NFA
 $\Sigma = \{a, b, c\}$

TRANSITION TABLE OF ϵ NFA

State / Alphab et	a	b	c	ϵ
$\rightarrow q_0$	-	q_1	q_2	q_1, q_2
q_1	q_0	q_2	q_0, q_2	-
$*q_2$	-	--	--	-

q_0	q_1	q_2
q_0, q_1, q_2		
q_1		
q_2		

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF ϵ NFA

	a	b	c	ϵ
$\rightarrow q_0$	-	q_1	q_2	q_1, q_2
q_1	q_0	q_2	q_0, q_2	-
* q_2	-	--	--	-

	ϵ closure	b	ϵ closure
q_0	$q_0 \xrightarrow{b} q_1 \xrightarrow{\epsilon \text{ closure}} q_1$ $q_1 \xrightarrow{b} q_2 \xrightarrow{\epsilon \text{ closure}} q_2$ $q_2 \xrightarrow{b} \emptyset$	q_1	$\{q_1, q_2\}$

	ϵ closure	a	ϵ closure
q_0	$q_0 \xrightarrow{a} \emptyset$ $q_1 \xrightarrow{a} q_0$ $q_2 \xrightarrow{a} \emptyset$	$q_0 \cup q_1 \cup q_2$	$q_0 \xrightarrow{\epsilon \text{ closure}} q_0 q_1 q_2$

	ϵ closure	c	ϵ closure
q_0	$q_0 \xrightarrow{c} q_2$ $q_1 \xrightarrow{c} \{q_0, q_2\}$ $q_2 \xrightarrow{c} q_2$	$q_0 \xrightarrow{\epsilon \text{ closure}} q_0 q_1 q_2$	q_0, q_1, q_2

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF ϵ NFA

	a	b	c	ϵ
$\rightarrow q_0$	-	q_1	q_2	q_1, q_2
q_1	q_0	q_2	q_0, q_2	-
$*q_2$	-	--	--	-

84
 ϵ closure

	a	b	ϵ closure
q_1	$q_1 \xrightarrow{b} q_2$		q_2

ϵ closure

	ϵ closure	a	ϵ closure
q_1	$q_1 \xrightarrow{a} q_0$	$\xrightarrow{\epsilon \text{ closure}} \{q_0, q_1, q_2\}$	

ϵ closure

	ϵ closure	c	ϵ closure
q_1	$q_1 \xrightarrow{c} q_0$	$\xrightarrow{\epsilon \text{ closure}} \{q_0, q_1, q_2\}$	$q_2 \xrightarrow{\epsilon \text{ closure}} q_2$

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF ϵ NFA

	a	b	c	ϵ
$\rightarrow q_0$	-	q_1	q_2	q_1, q_2
q_1	q_0	q_2	q_0, q_2	-
$*q_2$	-	--	--	-

84
94
96
12
1455
Gateway Classes

	ϵ closure	b	ϵ closure
q_2	q_{12}	ϕ	-

	ϵ closure	a	ϵ closure
q_2	q_{12}	ϕ	-

84
94
96
12
1455
Gateway Classes

	ϵ closure	c	ϵ closure
q_2	q_{12}	ϕ	-

TRANSITION TABLE OF NFA

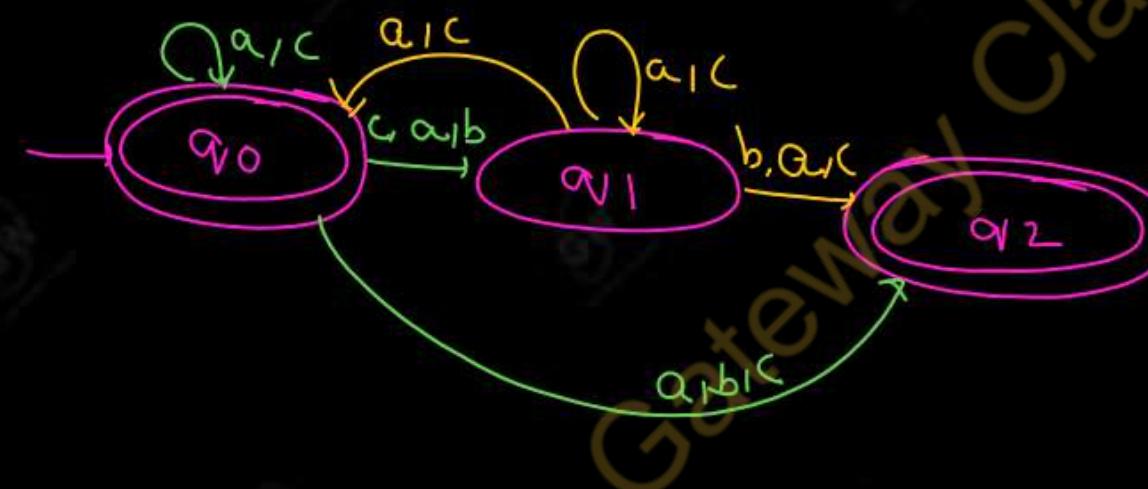
Input →	a	b	c
Start ↓	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$\rightarrow^* q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
q_1	$\{q_0, q_1, q_2\}$	q_2	$\{q_0, q_1, q_2\}$
$* q_2$	—	—	—

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

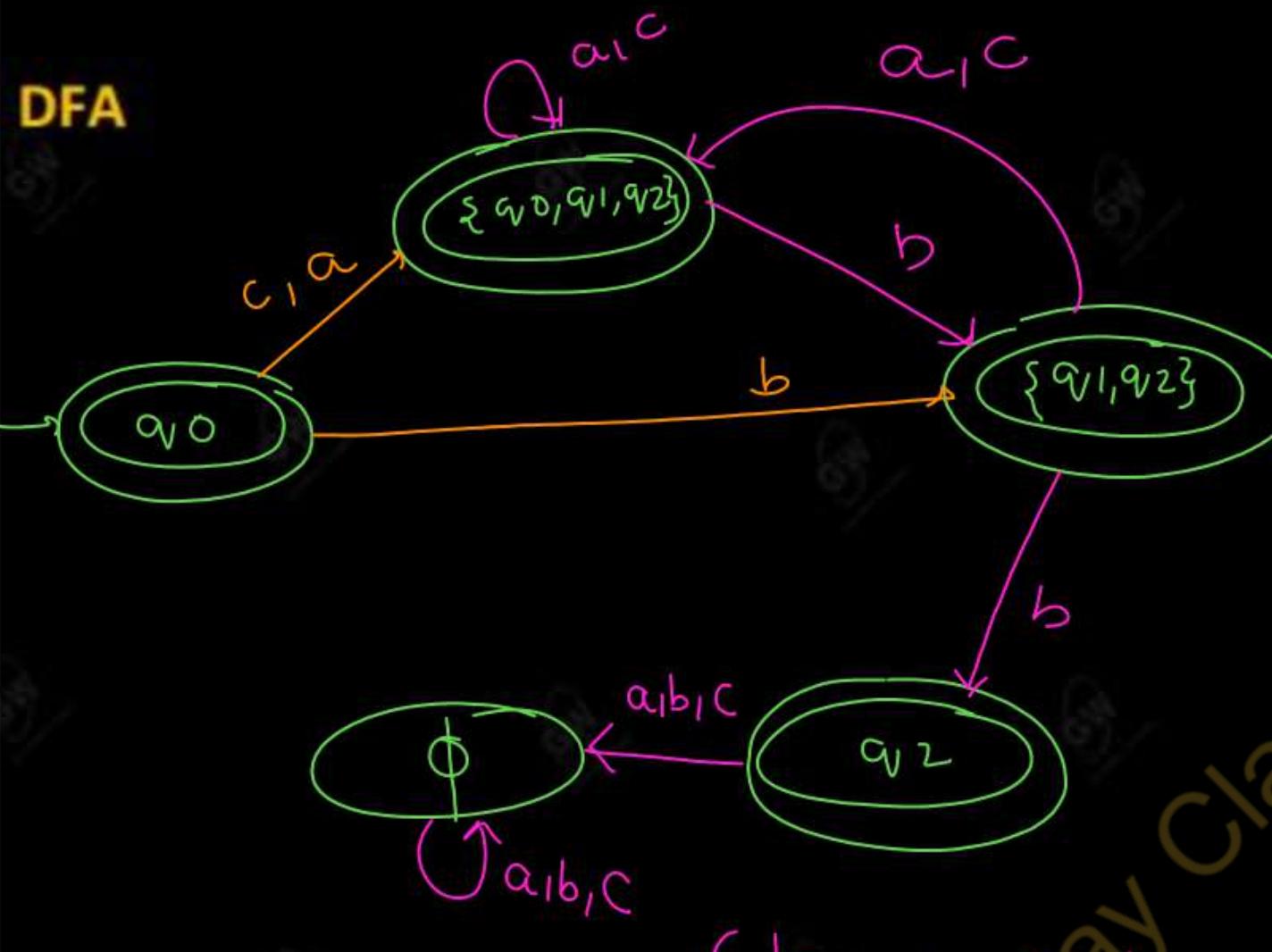
TRANSITION TABLE OF NFA

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

DFA (NFA TO DFA)

	a	b	c
$\rightarrow^* q_{v0}$	$\{q_{v0}, q_{v1}, q_{v2}\}$	$\{q_{v1}, q_{v2}\}$	$\{q_{v0}, q_{v1}, q_{v2}\}$
$\{q_{v0}, q_{v1}, q_{v2}\}$	$\{q_{v0}, q_{v1}, q_{v2}\}$	$\{q_{v1}, q_{v2}\}$	$\{q_{v0}, q_{v1}, q_{v2}\}$
$\{q_{v1}, q_{v2}\}$	$\{q_{v0}, q_{v1}, q_{v2}\}$	q_{v2}	$\{q_{v0}, q_{v1}, q_{v2}\}$
$*q_{v2}$	\emptyset	\emptyset	\emptyset
\emptyset	\emptyset	\emptyset	\emptyset

Convert following NFA to equivalent DFA and MDFA



(Transition Diagram
for DFA)

DFA (NFA TO DFA)

(transitntable DFA)

	a	b	c
* q_0	{ q_0, q_1, q_2 }	{ q_1, q_2 }	{ q_0, q_1, q_2 }
*{ q_0, q_1, q_2 }	{ q_0, q_1, q_2 }	{ q_1, q_2 }	{ q_0, q_1, q_2 }
*{ q_1, q_2 }	{ q_0, q_1, q_2 }	q_2	{ q_0, q_1, q_2 }
* q_2	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)

	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

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

MINIMIZATION OF DFA

$$\pi_0 = \{A, B, C, D\} \setminus \{\phi\}$$

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

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

$$\pi_3 = \{A, B\} \setminus \{C\}, \{D\}, \{\phi\}$$

Convert following NFA to equivalent DFA and MDFA

TRANSITION TABLE OF DFA

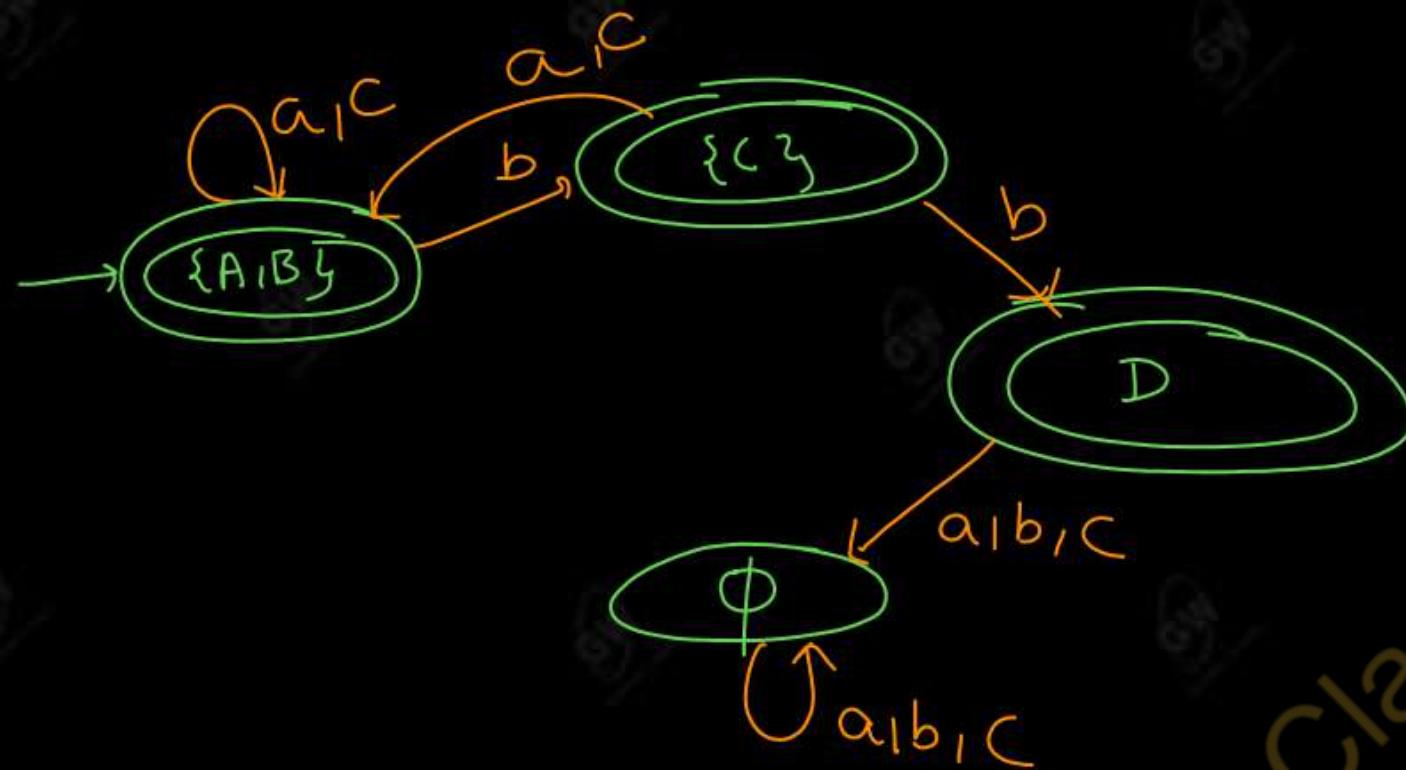
	a	b	c
* A	B	C	B
* B	B	C	B
* C	B	D	B
* D	\emptyset	\emptyset	\emptyset
\emptyset	\emptyset	\emptyset	\emptyset

	a	b	c
* {A,B}	{A,B}	C	{A,B}
* C	{A,B}	D	{A,B}
* D	\emptyset	\emptyset	\emptyset
\emptyset	\emptyset	\emptyset	\emptyset

{minimized DFA table}

Convert following NFA to equivalent DFA and MDFA

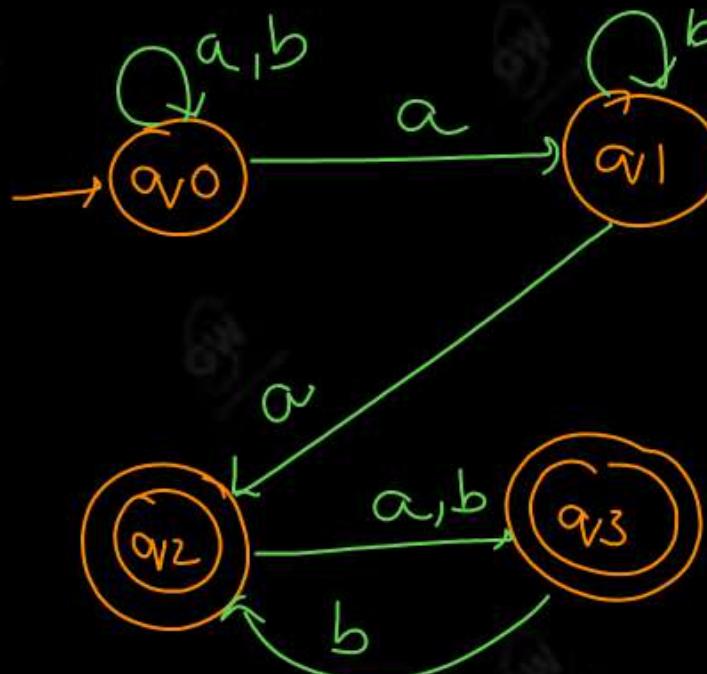
MINIMIZED DFA



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

{minimized DFA table}

Construct the minimum automata equivalent to NFA



(transition
Diagram
for(NFA))

Transition table for NFA

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

Construct the minimum automata equivalent to NFA

Transition table DFA

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

Transition table for NFA

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

Construct the minimum automata equivalent to NFA

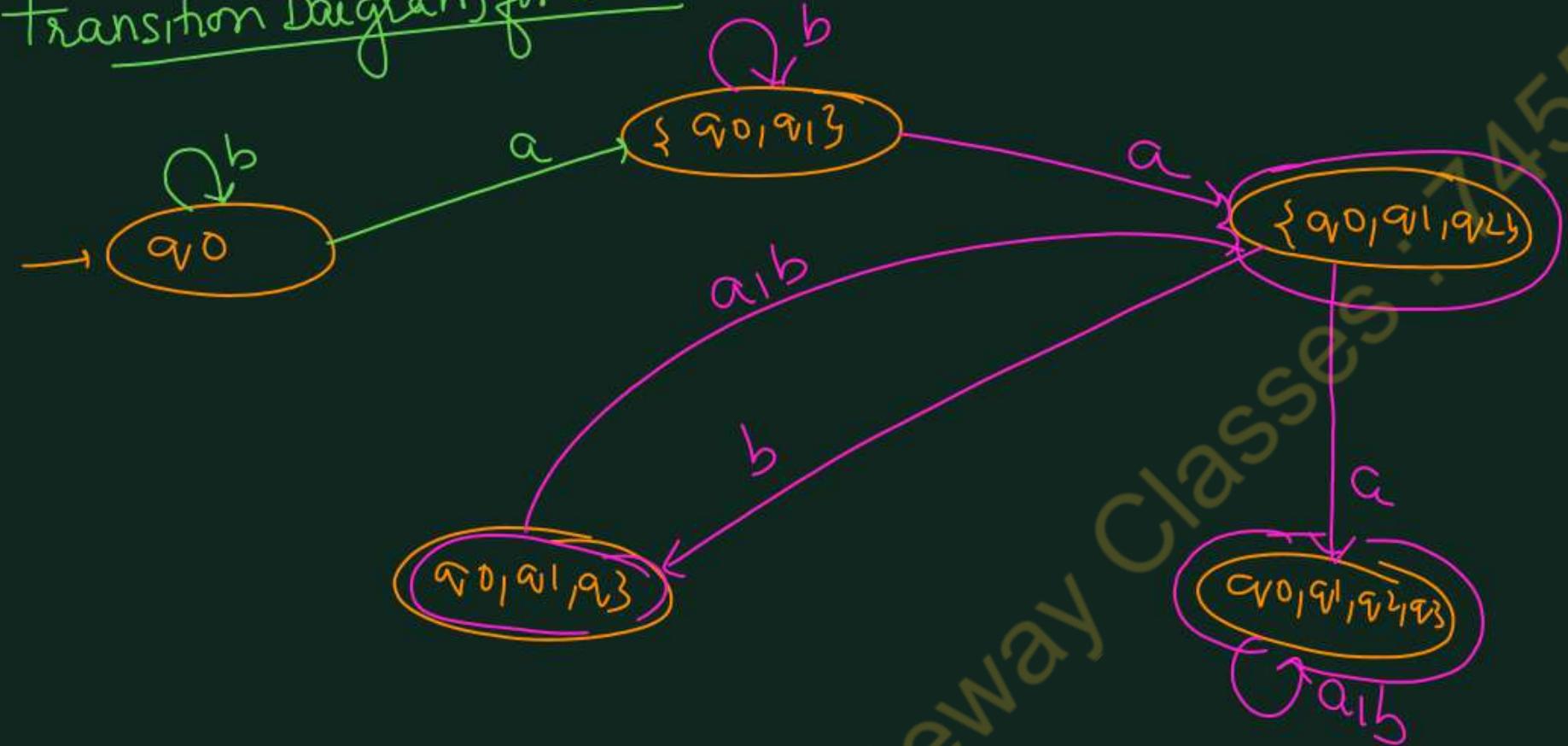
Transition table DFA

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

Transition table for DFA

	a	b
$\rightarrow 0$	1	0
1	2	1
2	3	4
3	3	3
4	2	2

transition Diagram for DFA



MINIMIZATION OF DFA

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

0-equivalence class

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

1-equivalence class

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

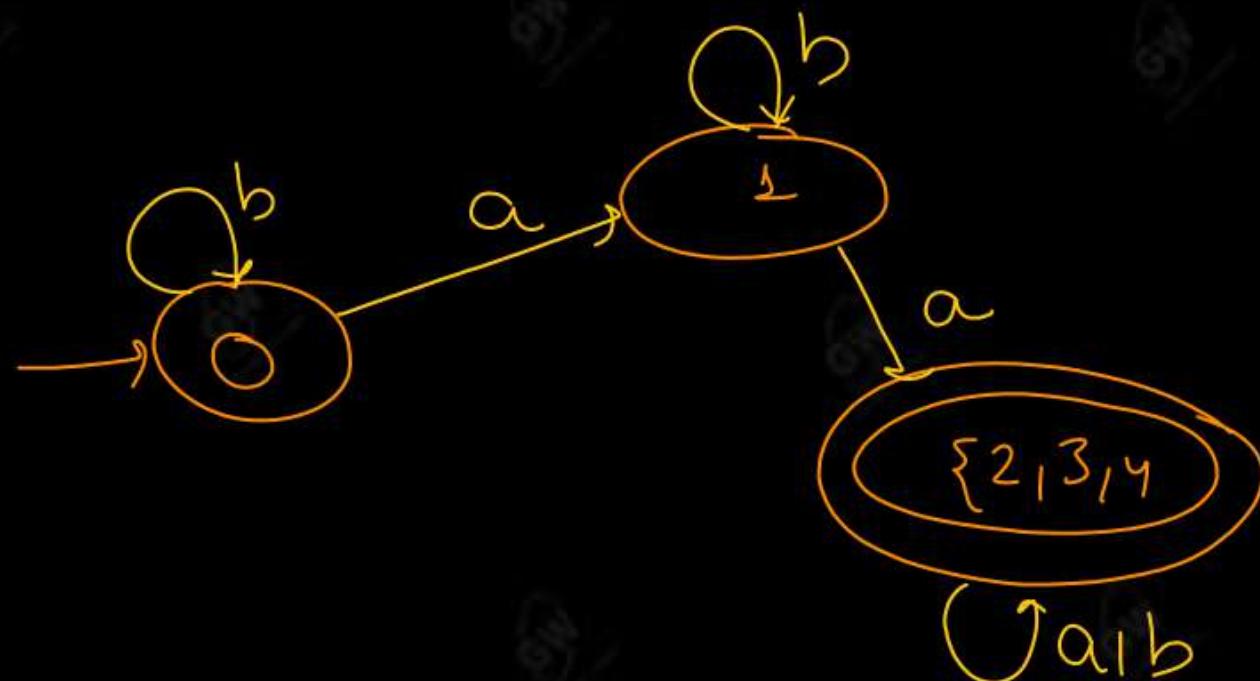
2-equivalence class

Transition table for DFA

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

Gateway Classes

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}

Gateway Classes

BASIC CONCEPTS AND AUTOMATA THEORY

TODAY's TARGET

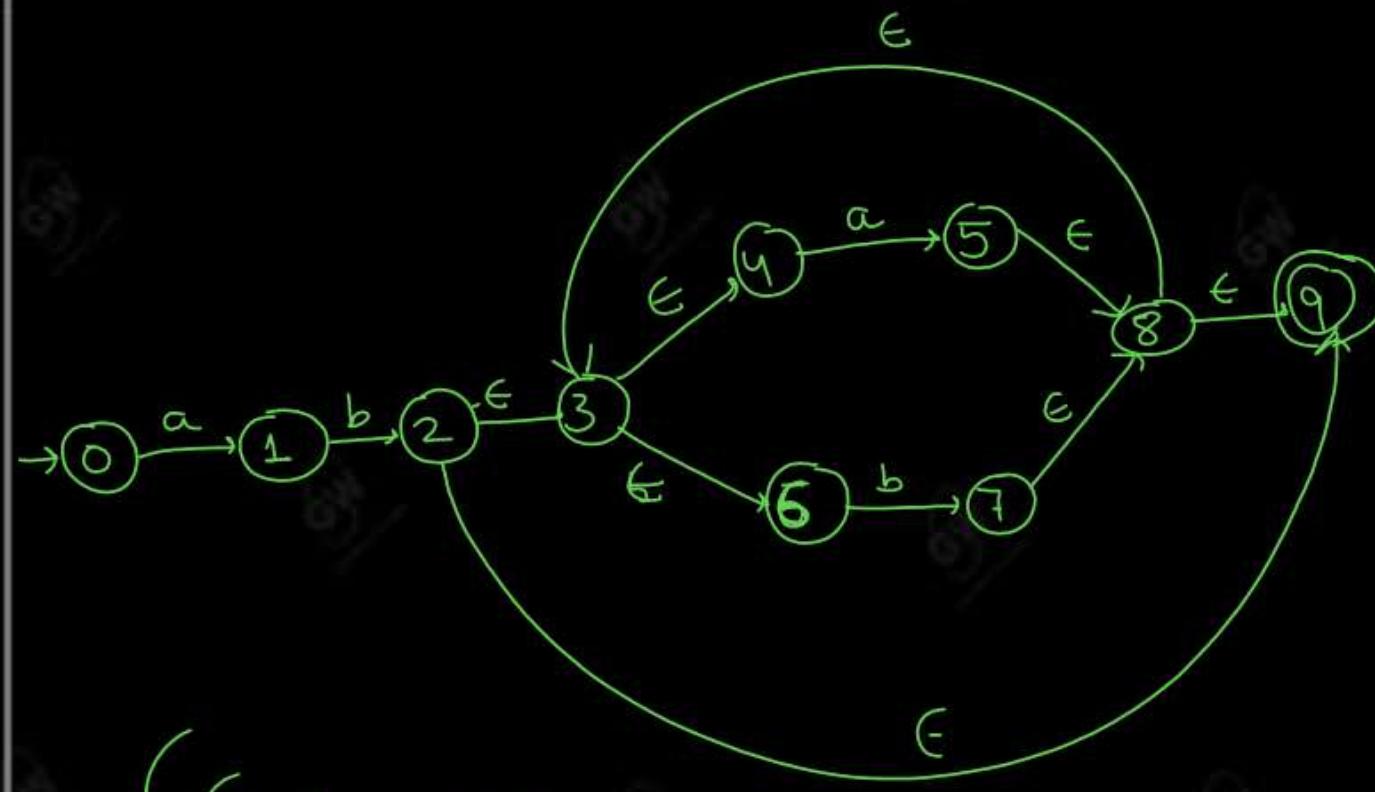
- PRACTICE PROBLEM ON EPSILON NFA TO DFA

WHEN TOO MANY STATES ARE GIVEN

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

Question. EPSILON NFA TO DFA CONVERSION

 $\Sigma = \{0, 1, 3\}$


(Epsilon NFA
transition Diagram)

	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
Q0	Q0	1	1	∅	-

	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	\emptyset	-	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	$7,8,9,3,4,6$
\emptyset				-

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

$\epsilon \text{NFA} \rightarrow \hat{q}_1$ $\text{DFA} \rightarrow \text{Final state } q_{\text{final}}$

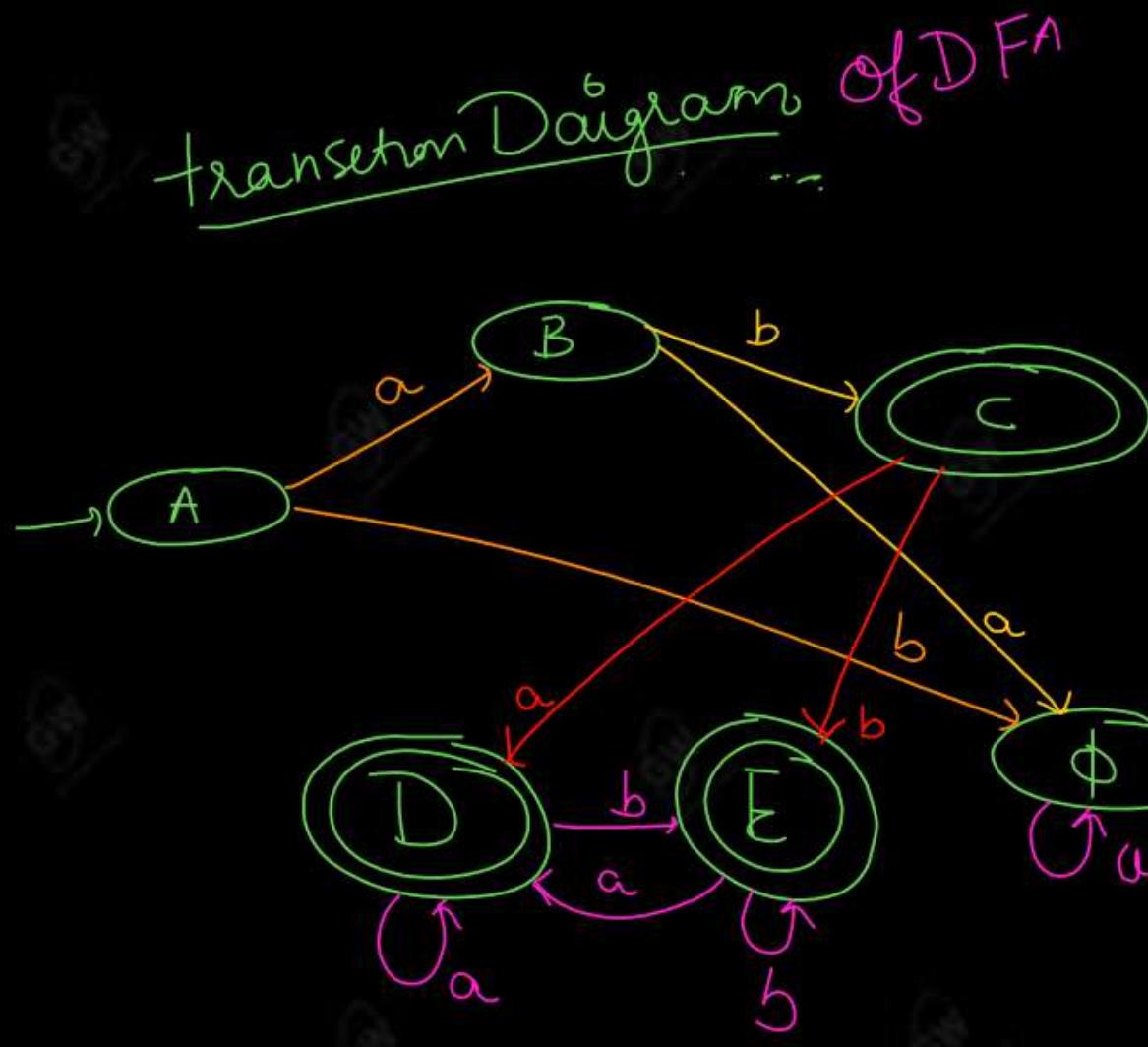
	a	b
$\rightarrow \circ$	L	ϕ
1	ϕ	$\{2, 3, 9, 4, 6\}$
$\ast \{2, 3, 9, 4, 6\}$	$\{5, 8, 9, 3, 4, 6\}$	$\{7, 8, 9, 3, 4, 6\}$
$\ast \{5, 8, 9, 3, 4, 6\}$	$\{5, 8, 9, 3, 4, 6\}$	$\{7, 8, 9, 3, 4, 6\}$
$\ast \{7, 8, 9, 3, 4, 6\}$	$\{5, 8, 9, 3, 4, 6\}$	$\{7, 8, 9, 3, 4, 6\}$
ϕ	ϕ	ϕ

DFA table

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

DFA Table

	a	b
$\rightarrow A$	B	ϕ
B	ϕ	C
$*C$	D	E
$*D$	D	E
$*E$	D	E
ϕ	ϕ	ϕ



34

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

Only if ENFA to minimize 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

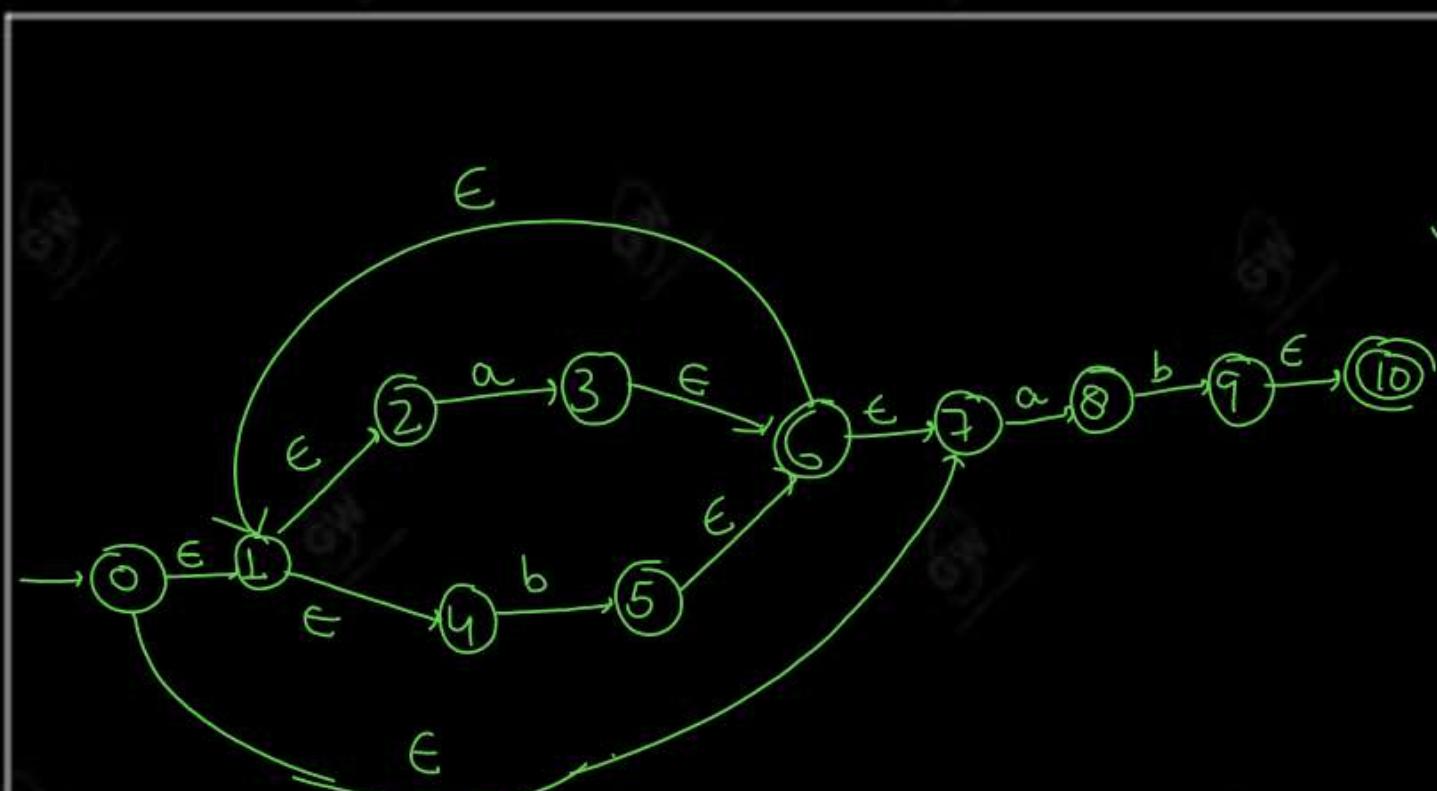
	a	b
→A	B	φ
B	φ	C
*C	D	E
*D	D	E
*E	D	E
φ	φ	φ

	a	b
$\rightarrow A$	B	\emptyset
B	\emptyset	$\{C, D, E\}$
$*\{C, D, E\}$	$\{C, D, E\}$	$\{C, D, E\}$
\emptyset	\emptyset	\emptyset

Minimized DFA table

	a	b
$\rightarrow A$	B	\emptyset
B	\emptyset	C
$*C$	D	E
$*D$	D	E
$*E$	D	E
\emptyset	\emptyset	\emptyset

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
$\rightarrow 0$	-	-	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 ∪ 0, 8

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

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

	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
$\rightarrow \{0, 1, 2, 4, 7\}$	$\{1, 2, 3, 4, 6, 7, 8\}$	$\{1, 2, 4, 5, 6, 7\}$
$\{1, 2, 3, 4, 6, 7, 8\}$	$\{1, 2, 3, 4, 6, 7, 8\}$	$\{1, 2, 4, 5, 6, 7, 9\}$
$\{1, 2, 4, 5, 6, 7\}$	$\{1, 2, 3, 4, 6, 7, 8\}$	$\{1, 2, 4, 5, 6, 7\}$
$\{1, 2, 4, 5, 6, 7, 9\}$	$\{1, 2, 3, 4, 6, 7, 8\}$	$\{1, 2, 4, 5, 6, 7, 10\}$
* $\{1, 2, 4, 5, 6, 7, 10\}$	$\{1, 2, 3, 4, 6, 7, 8\}$	$\{1, 2, 4, 5, 6, 7\}$

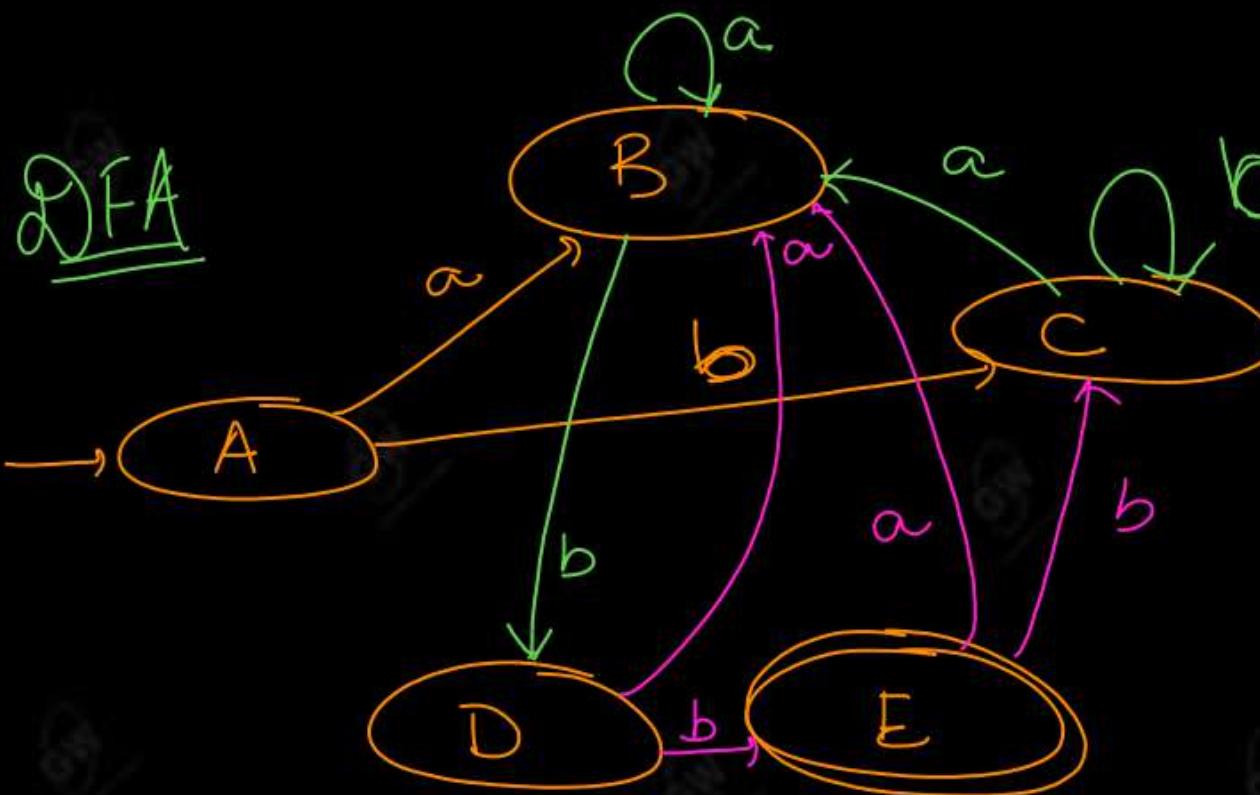
DFA table

	a	b
- let A > {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} 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

Minimized DFA

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

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

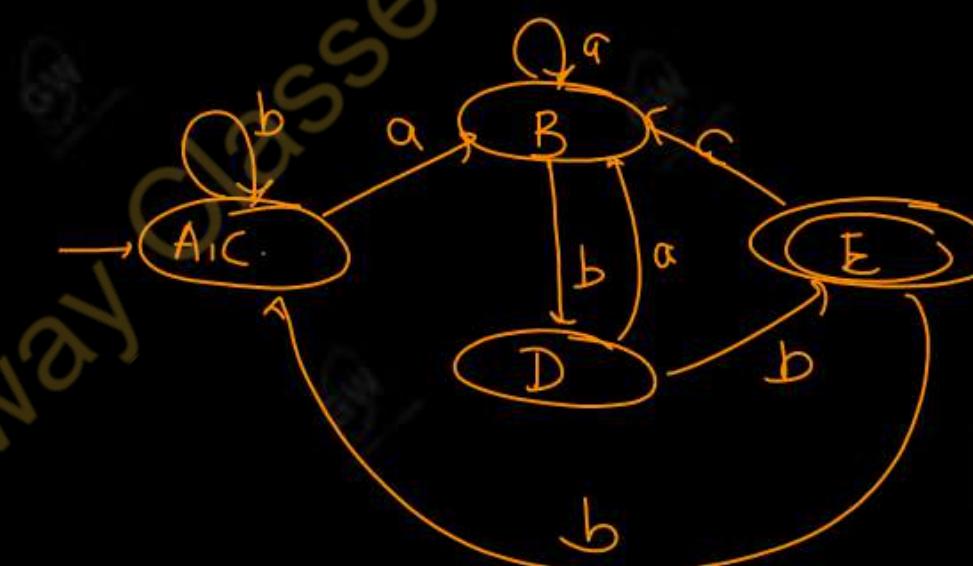
$$\pi_2 = \{A, C\}, \{B\}, \{D\}, \{E\}$$

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

	a	b
a	B	A, C
b	B	D
D	B	E
*E	B	A, C

(Minimized
DFA
Transition
table)

	a	b
a	B	C
b	B	D
C	B	C
D	B	E
*E	B	C



BASIC CONCEPTS AND AUTOMATA THEORY

TODAY's TARGET

- Mealy and Moore machine
- Designing of Mealy and Moore machine

By PRAGYA RAJVANSI
B.Tech, M.Tech(C.S.E)

- 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'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

$$(Q, q_0, \Sigma, \Delta, \delta, \lambda)$$

Q is a finite set of states $\{q_0, q_1, q_2, q_3\}$

q₀ is the initial state q_0

Σ is the input alphabet $\{0, 1\}$

Δ is the output alphabet $\{0, 1\}$

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

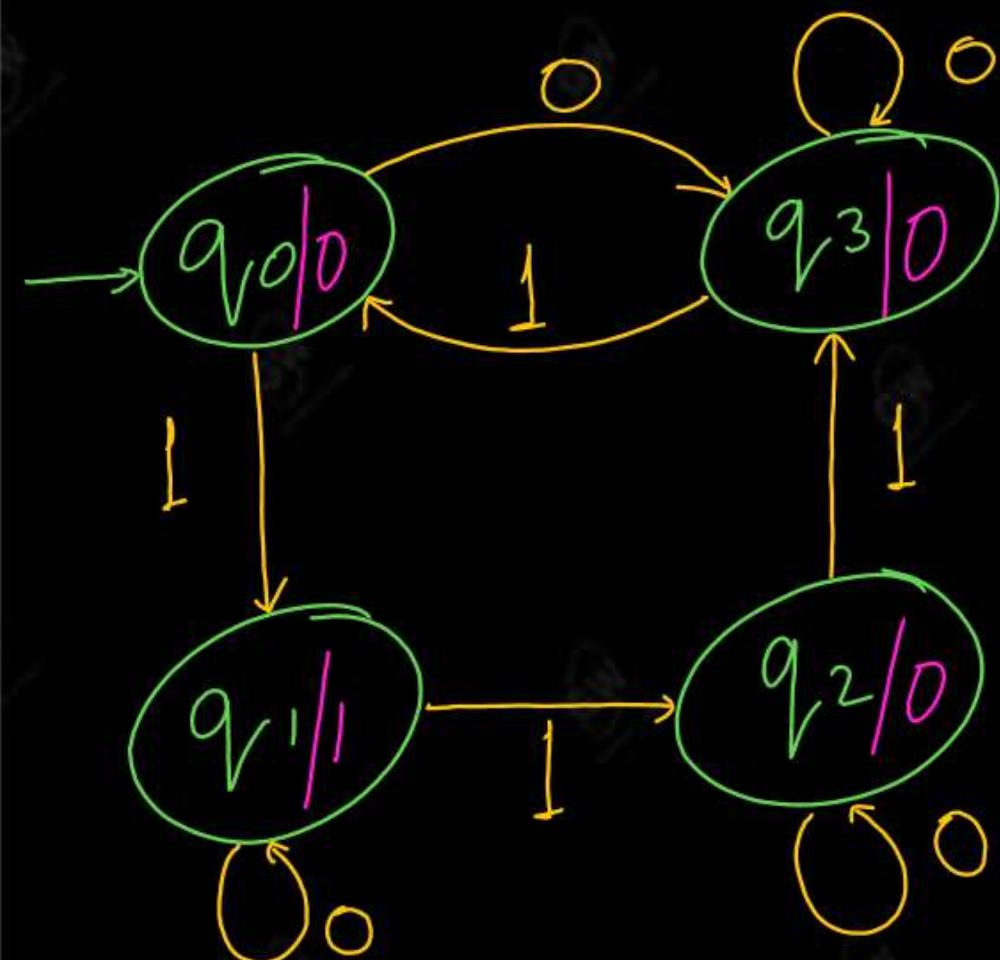
λ is the output function that maps $Q \rightarrow \Delta$

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

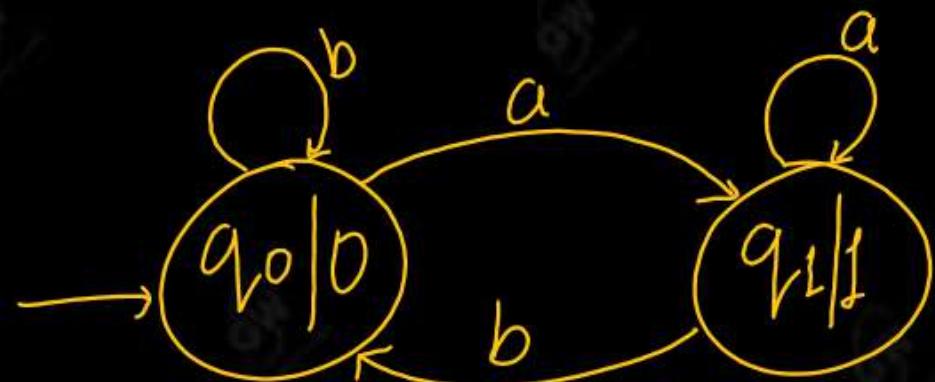
n 1110 - Output?

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

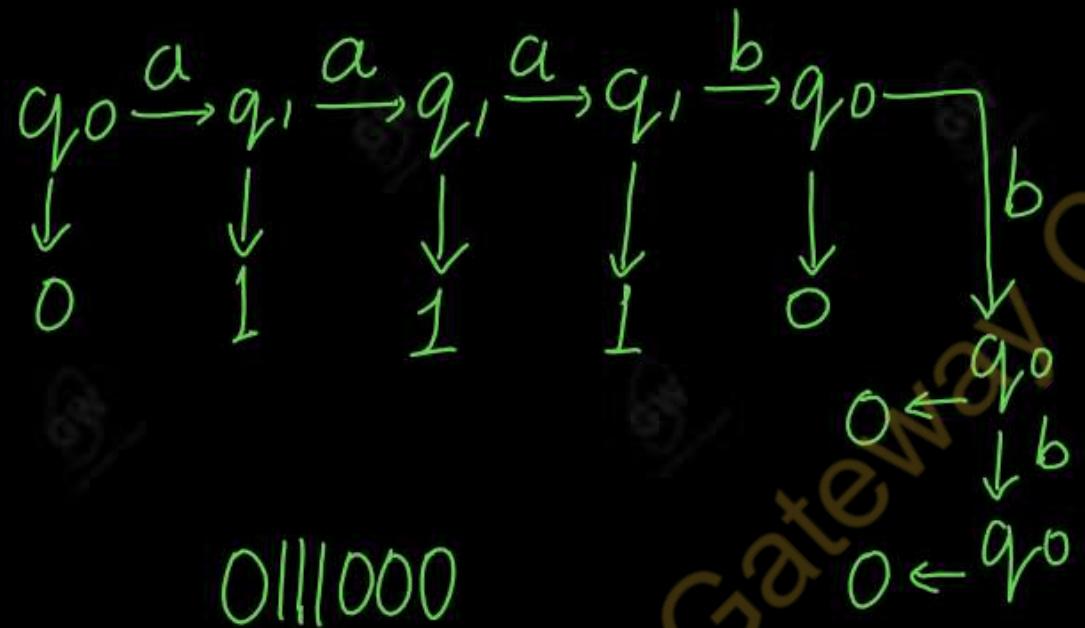
0 0 0 0 n+1

Present state	Next state $a=0$	Next state $a=1$	$\Lambda(\text{output})$
$\rightarrow q_0$	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

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 $\Sigma = \{a, b\}$, $\Delta = \{0, 1\}$

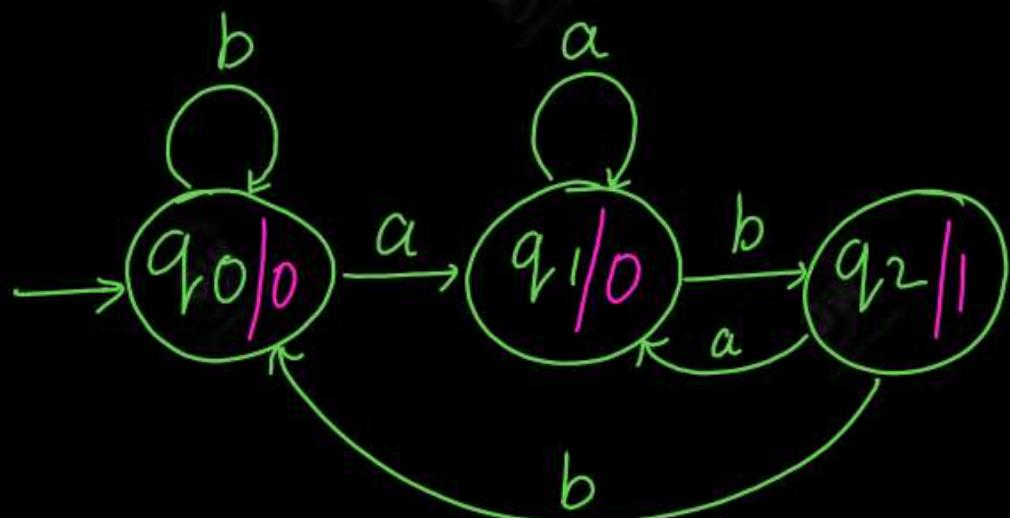


Present state	Next state a	Next state b	Δ (output)
$\rightarrow q_0$	q_1	q_0	0
q_1	q_1	q_0	1



aaabbbb
Gateway Classes
0111000
 q^0
 $0 < q^0$

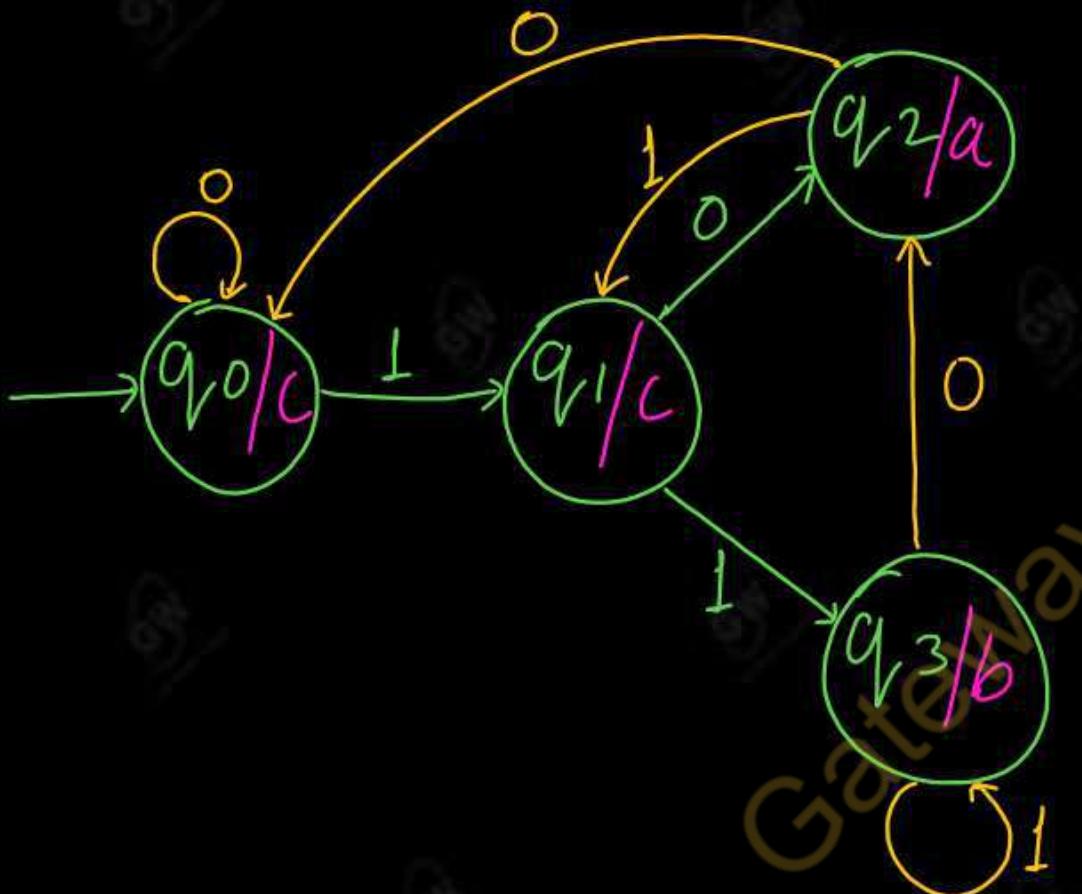
Construct a Moore machine that takes all string of a and b as input and count number of occurrence of sub-string 'ab' in term of 1 $\Sigma = \{a, b\}$, $\Delta = \{0, 1\}$



Present state	a	b	Δ (output)
$\rightarrow q_0$	q_1	q_0	0
q_1	q_1	q_2	0
q_2	q_1	q_0	1

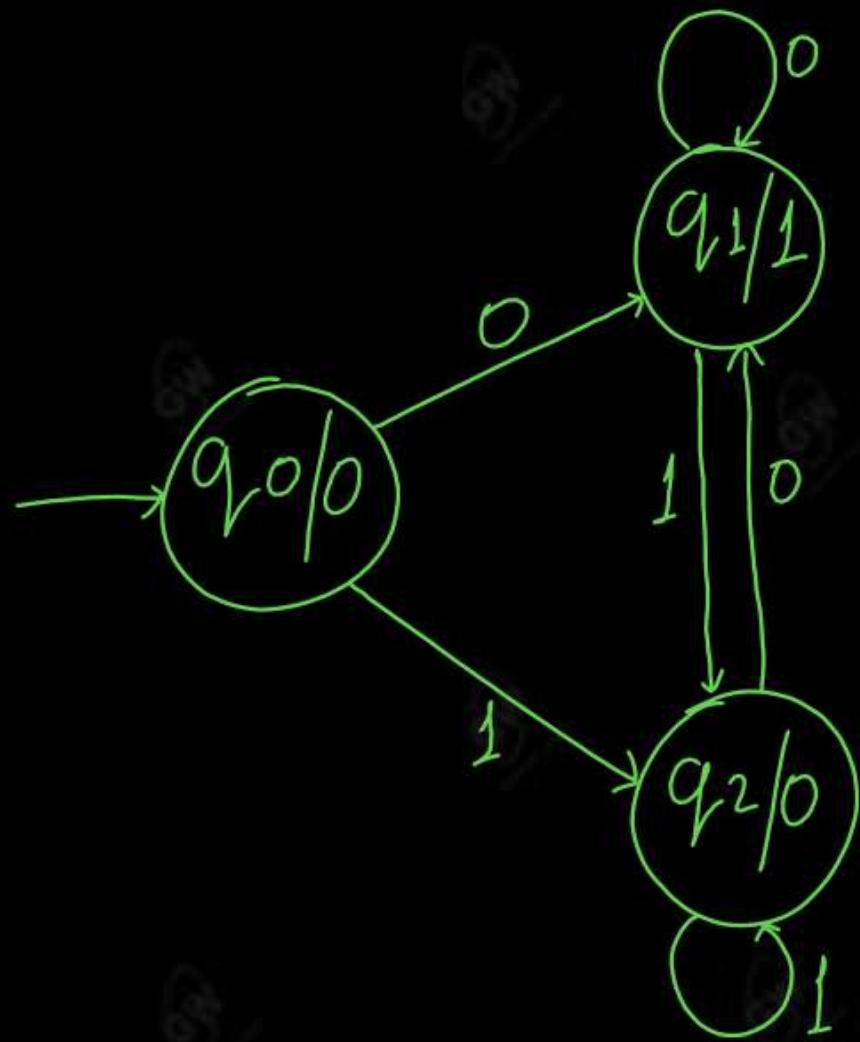
Construct a Moore machine that should give output a if the input string end with 10 ,b if the input string ends with 11 $\Sigma = \{0,1\}$, $\Delta = \{a,b,c\}$

10 $\rightarrow a$
11 $\rightarrow b$
other than this c



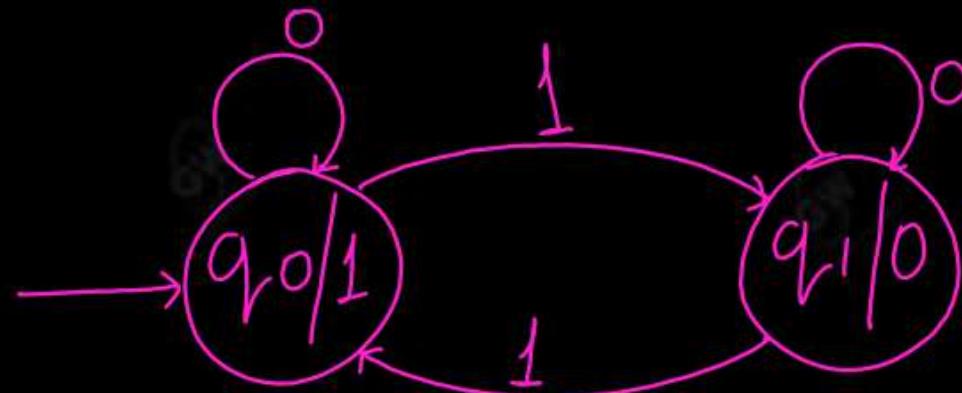
Present state	Next state 0	Next state 1	$\Delta(\text{output})$
$\rightarrow q_0$	q_0	q_1	c
q_1	q_2	q_3	c
q_2	q_0	q_1	a
q_3	q_2	q_3	b

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

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


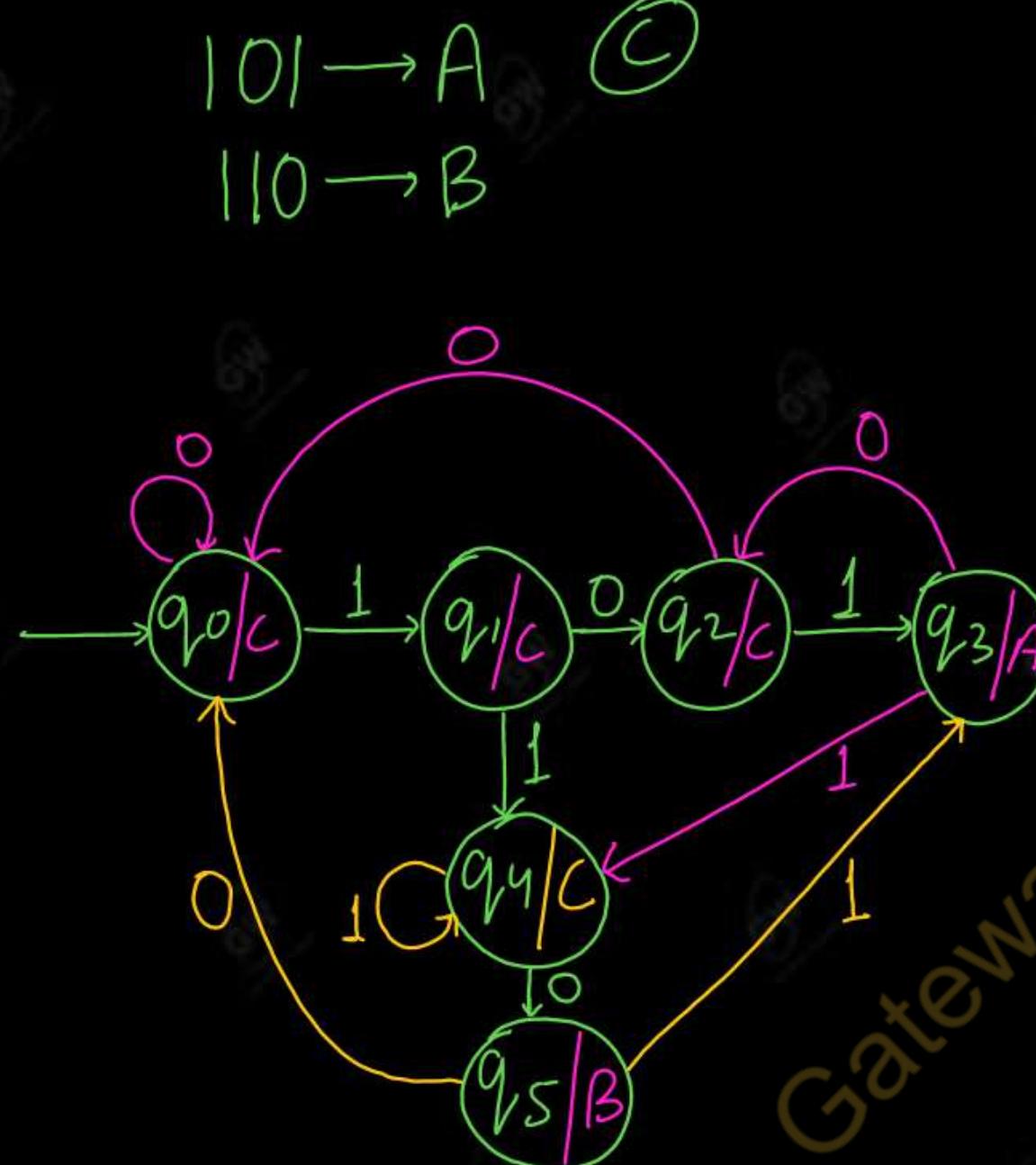
Present state	Next state 0	Next state 1	Λ (output)
$\rightarrow q_0$	q_1	q_2	0
q_1	q_1	q_2	1
q_2	q_1	q_2	0

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	Next state 1	Λ (output)
$\rightarrow q_0$	q_0	q_1	1
q_1	q_1	q_0	0

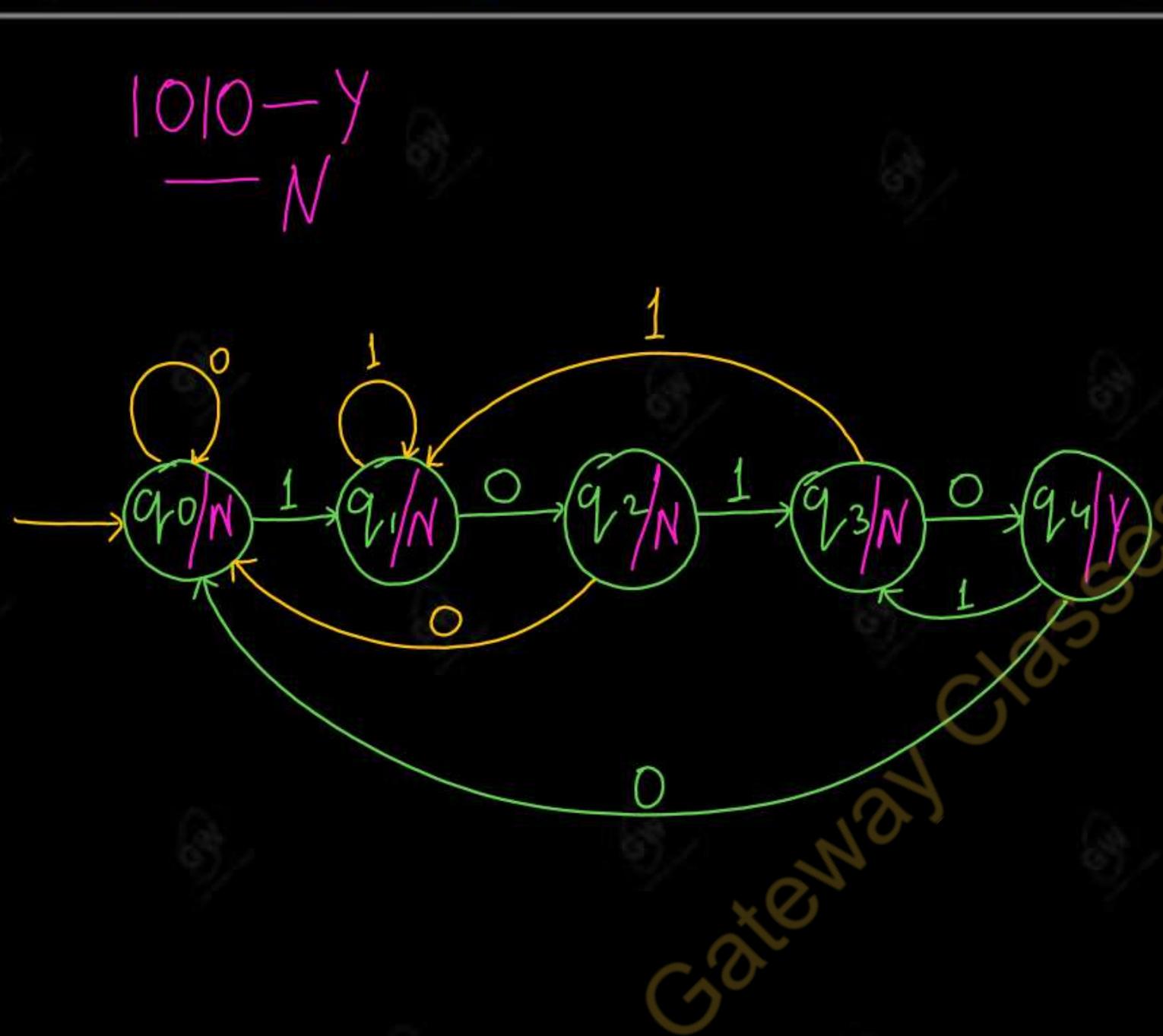
Design a Moore machine for a binary input sequence such that if it has a substring 101 then the machine output A, if the input has substring 110, it output B otherwise output C



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

Present state	Next state 0	Next state 1	Λ (output)
$\rightarrow q_0$	q_0	q_1	C
q_1	q_2	q_4	C
q_2	q_0	q_3	C
q_3	q_2	q_4	A
q_4	q_5	q_4	C
q_5	q_0	q_3	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	0	1	Λ (output)
$\rightarrow q_0$	<u>q_0</u>	<u>q_1</u>	N
q_1	<u>q_2</u>	<u>q_1</u>	N
q_2	<u>q_0</u>	<u>q_3</u>	N
q_3	<u>q_4</u>	<u>q_1</u>	N
q_4	<u>q_0</u>	<u>q_4</u>	Y

BASIC CONCEPTS AND AUTOMATA THEORY

TODAY's TARGET

- Mealy Machine
- Designing of Mealy machine

By PRAGYA RAJVANSI
B.Tech, M.Tech(C.S.E)

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 $(Q, q_0, \Sigma, O, \delta, \lambda')$ where:

Q is a finite set of states.

q_0 is the initial state.

Σ is the input alphabet.

O is the output alphabet.

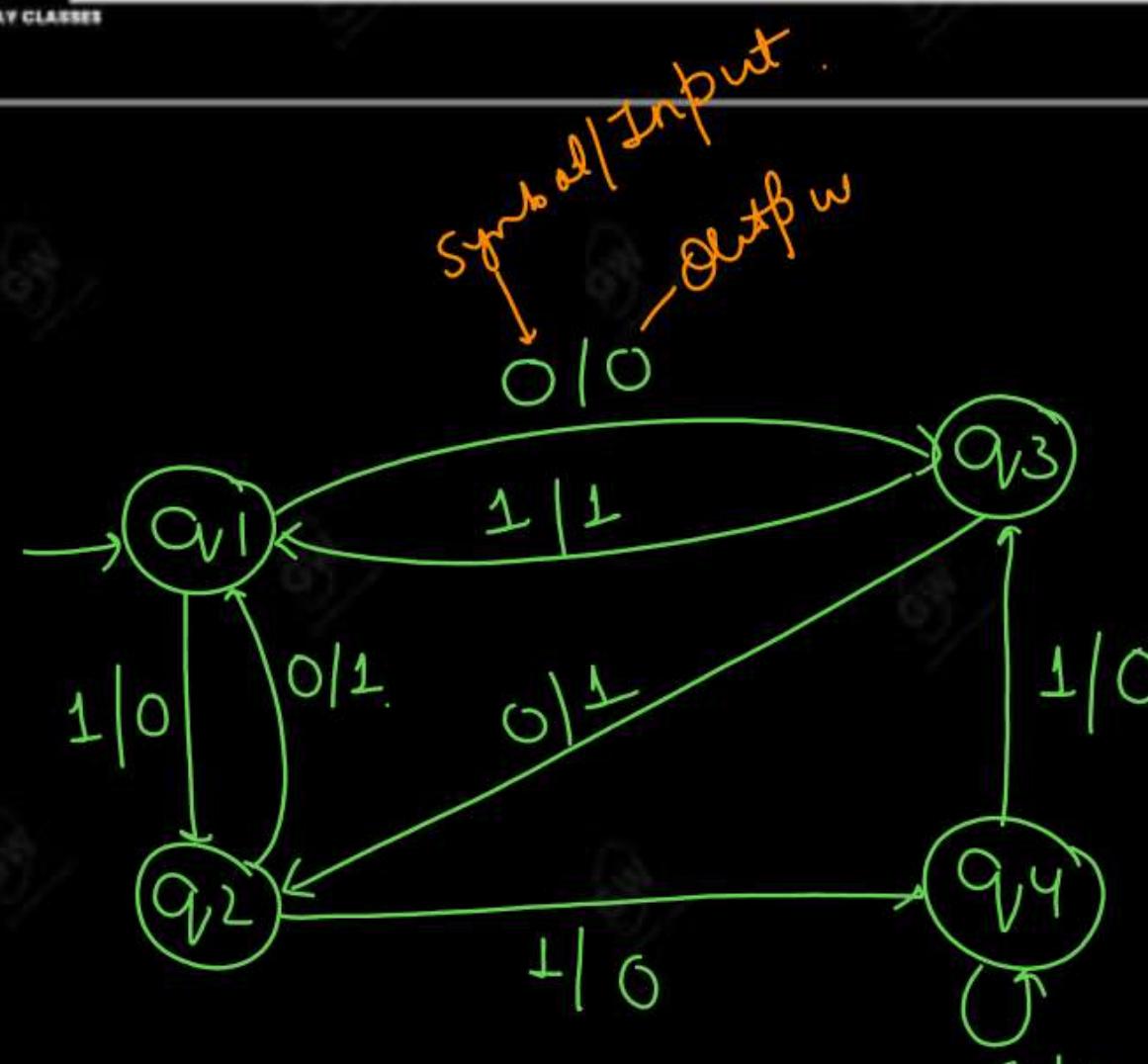
δ is the transition function which maps $Q \times \Sigma \rightarrow Q$.

λ' is the output function that maps $Q \times \Sigma \rightarrow O$.

NOTE:

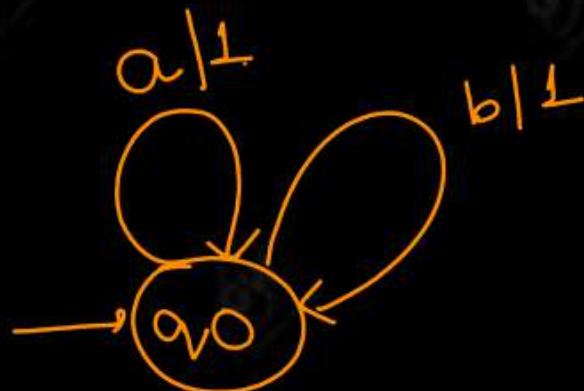
- If the length of input string is n , then length of output string will be n
- Mealy machine do not response for empty string

Mealy machine



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

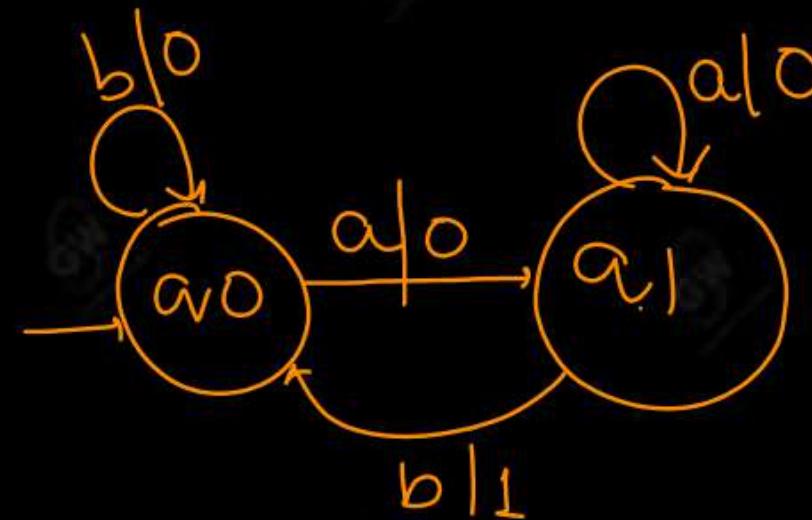
Construct a Mealy machine that takes all string of a and b as input and count number of a in the input string in term of 1 $\Sigma = \{a, b\}$, $\Delta = \{0, 1\}$



	a State	output	b State	output
q0	q0	1	q0	b

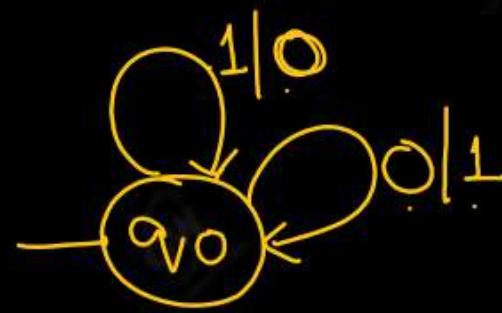
aab
 $q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0$

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 $\Sigma = \{a,b\}$, $\Delta = \{0,1\}$



	a State	output	b State	output
$\rightarrow q_0$	q_1	0	q_0	0
q_1	q_1	0	q_0	1

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

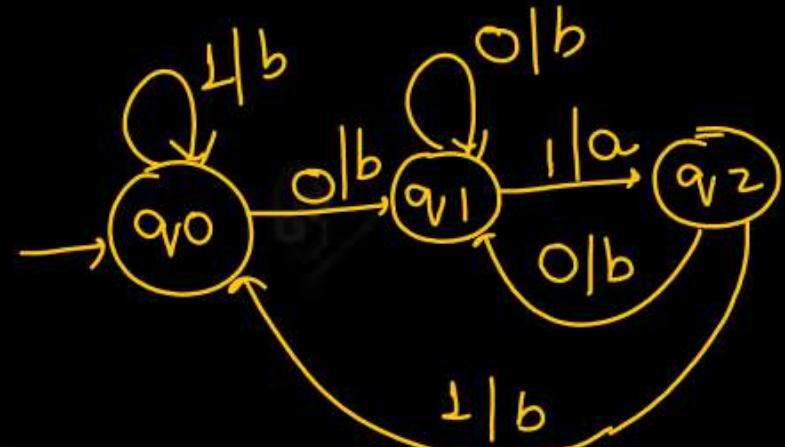


	0 State	output	1 State	output
-> q_0	q_0	1	q_0	0

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

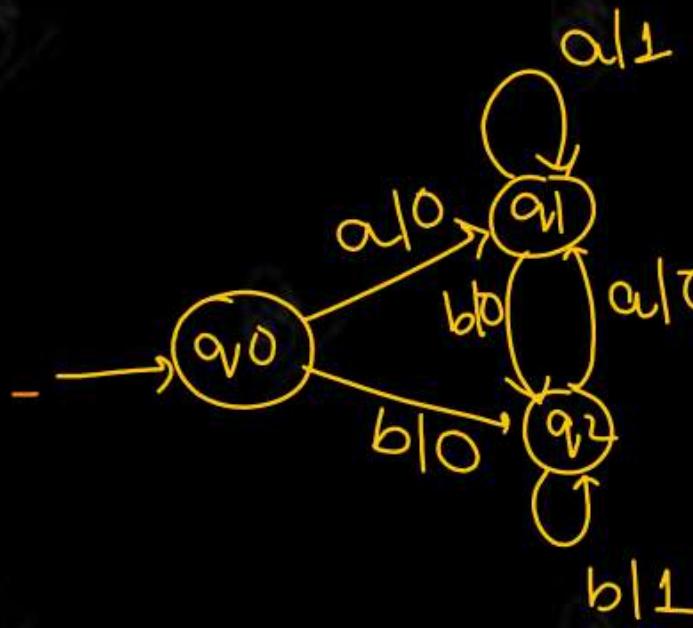
$$\Sigma = \{0, 1\}$$

Output a/b

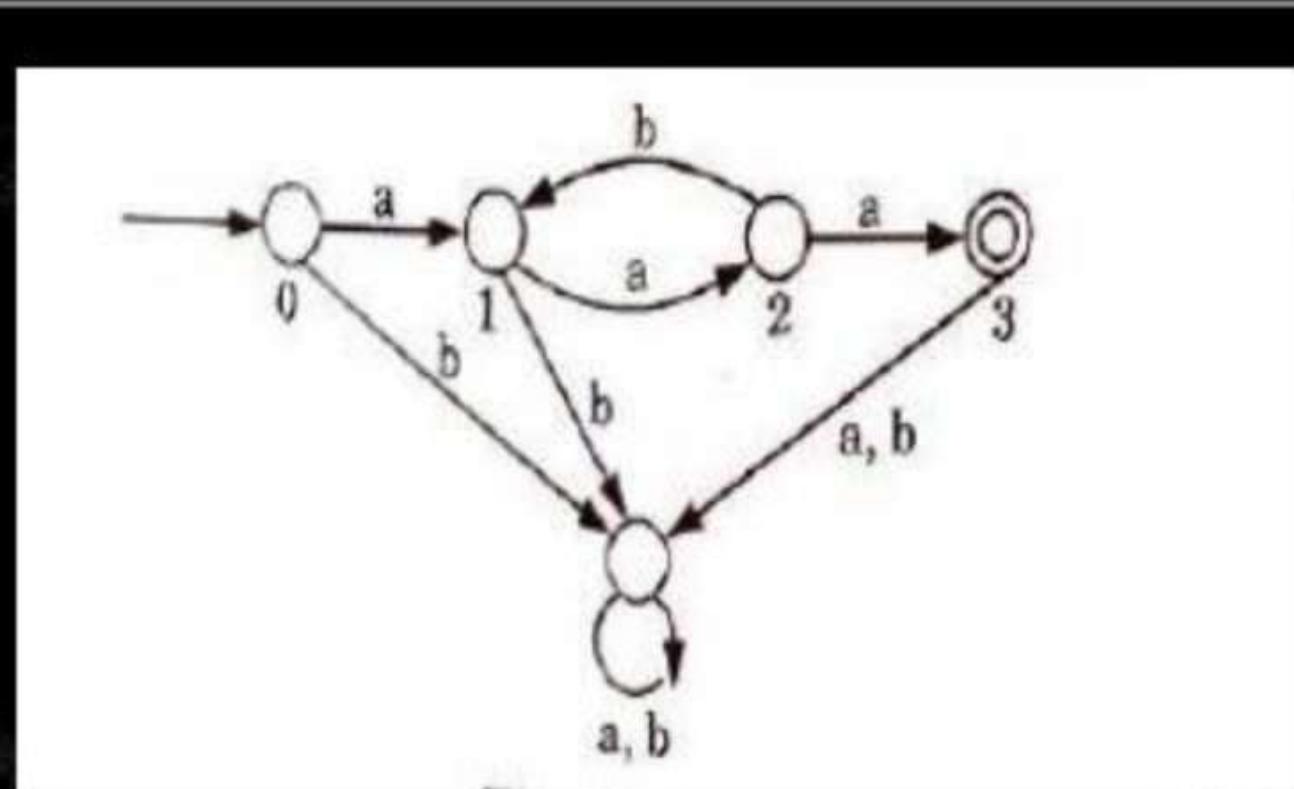


	0 State	output	1 State	output
-> q_0	q_1	b	q_0	b
q_1	q_1	b	q_2	a
q_2	q_1	b	q_0	b

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
-> q_0	q_1	0	q_2	0
q_1	q_1	1	q_2	0
q_2	q_1	0	q_2	1

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

DFA that accept all string
that start & end with a

$L = \{w \in \{a, b\}^* \mid w \text{ accept all the string}$
 $\text{that start and end with } a\}$

DIFFERENCE 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

ϵ N.F

DFA allows only one move for single input alphabet.

There can be choice (more than one move) for single input alphabet.

$\delta: Q \times \Sigma \rightarrow Q$ i.e. next possible state belongs to Q.

$\delta: Q \times (\Sigma \cup \epsilon) \rightarrow 2^Q$ 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 than 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.

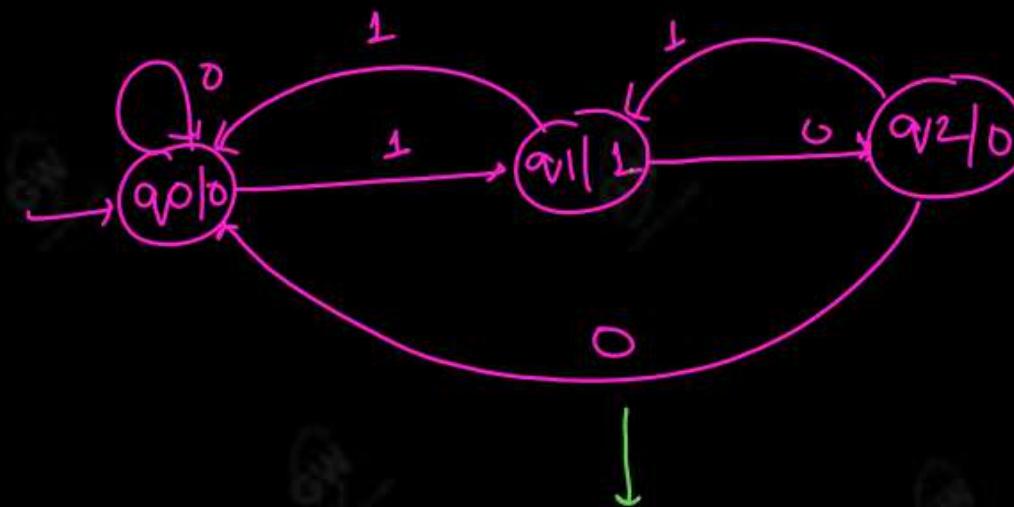
BASIC CONCEPTS AND AUTOMATA THEORY

TODAY's TARGET

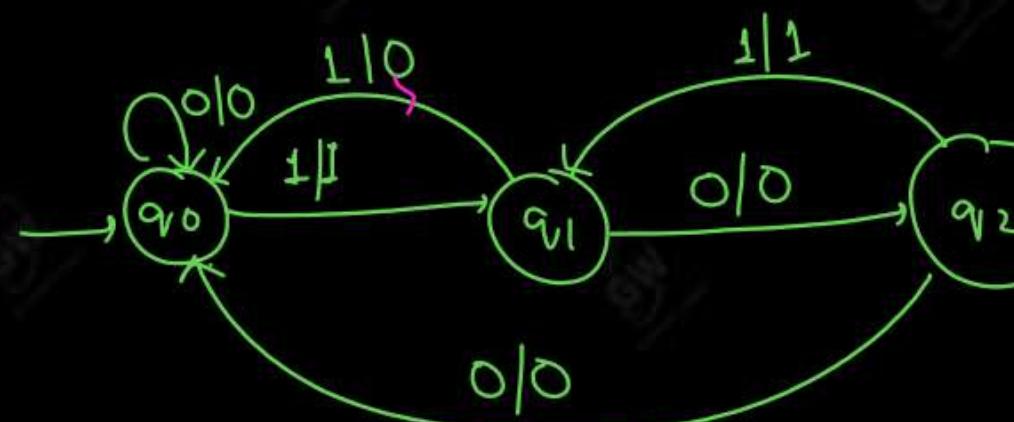
- Moore to Mealy conversion
- Mealy to Moore conversion

By PRAGYA RAJVANSHI
B.Tech, M.Tech(C.S.E)

Q1Moore to mealy machine conversion



(Moore Machine)



(Mealy Machine)

Present state	Next state	(output)
0	1	
→ q_0	q_0	0
q_1	q_2	1
q_2	q_1	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	0	1	(output)
Next state	q0	q1	0
Present state	q0	q1	1
Next state	q1	q0	0
Present state	q1	q0	1
Next state	q0	q1	0

Moore transition table

Q2 Moore to mealy machine conversion

	State 0 output	State 1 output
Present state	0	1
$\rightarrow q_0$	q_1	○
q_1	q_3	1
q_2	q_2	1
q_3	q_0	1

Mealy machine transition table

Present state	0	1	(output)
Next state	q1	q2	1
$\rightarrow q_0$	q1	q2	1
q_1	q3	q2	0
q_2	q2	q1	1
q_3	q0	q3	1

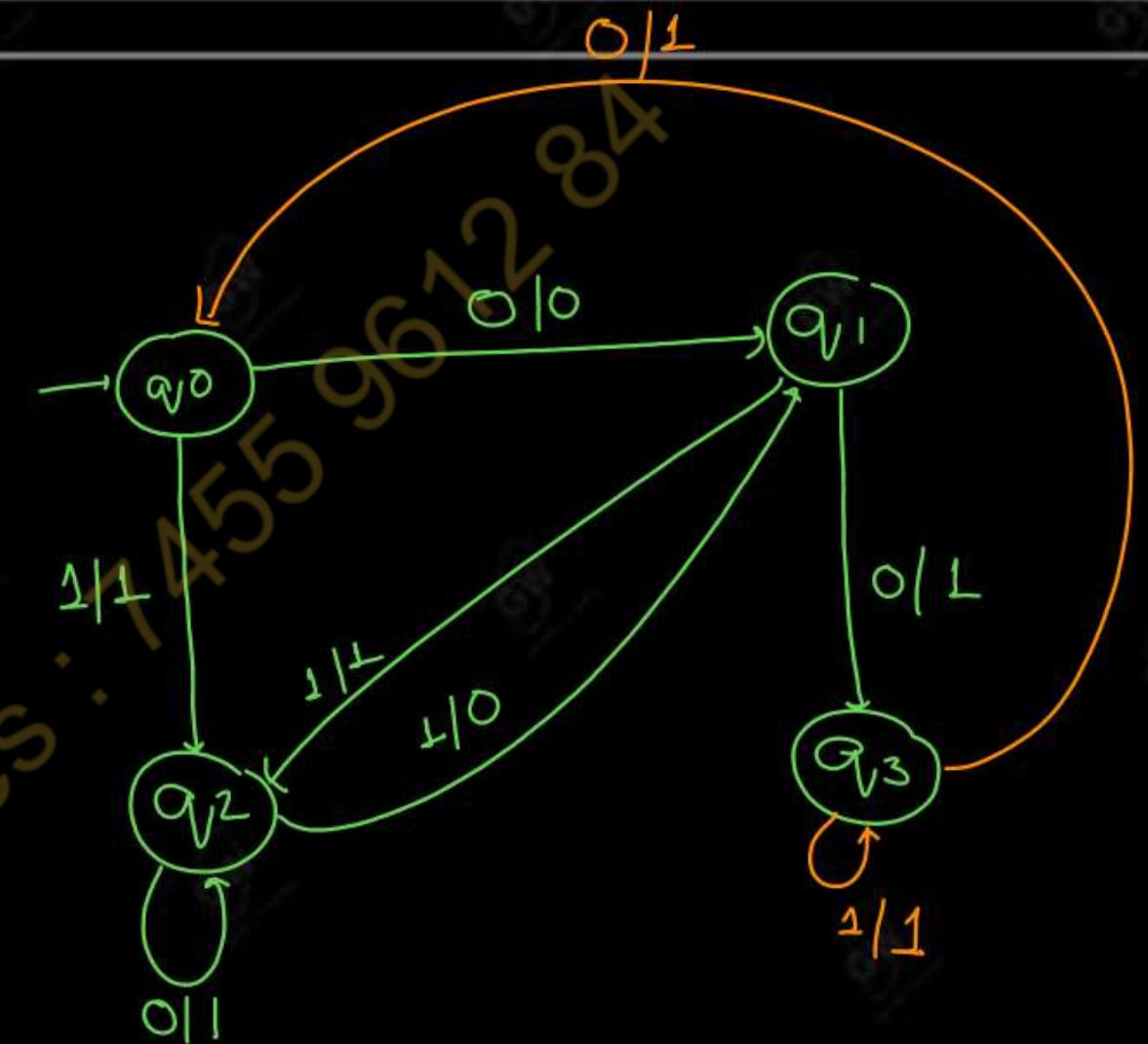
Moore transition table

Q2 Moore to mealy machine conversion

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

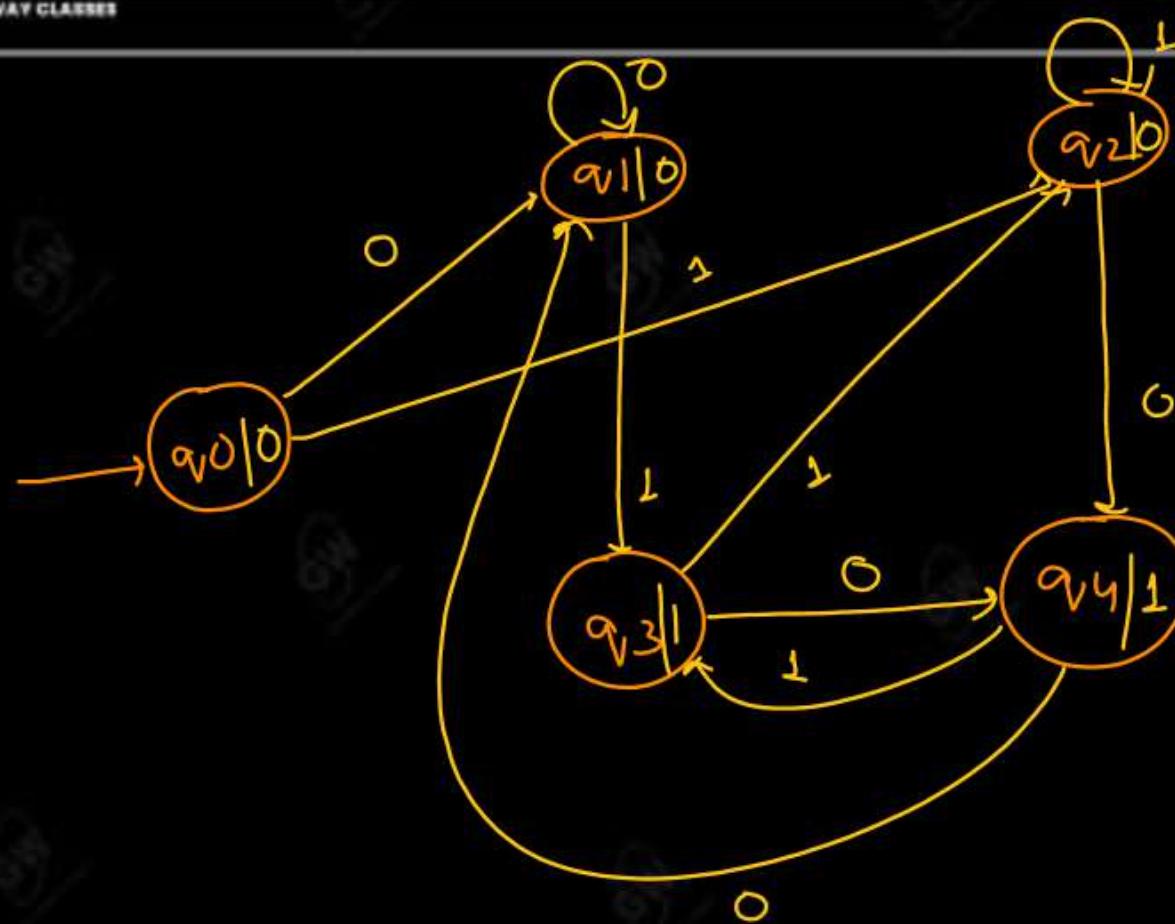
Mealy machine transition table

Gateway Classes



Mealy transition diagram

Q3Moore to mealy machine conversion



Present state	Next state 0	Next state 1	(output)
$\rightarrow q_0$	q_1	q_2	0
q_1	q_1	q_3	0
q_2	q_4	q_2	0
q_3	q_4	q_2	1
q_4	q_1	q_3	1

Moore machine transition table

Q3 Moore to mealy machine conversion

	State	0 output	1 output	
	State	0 output	1 output	
q_0	q_1	0	q_2	0
q_1	q_1	0	q_3	1
q_2	q_4	1	q_2	0
q_3	q_4	1	q_2	0
q_4	q_1	0	q_3	1

Mealy machine transition table

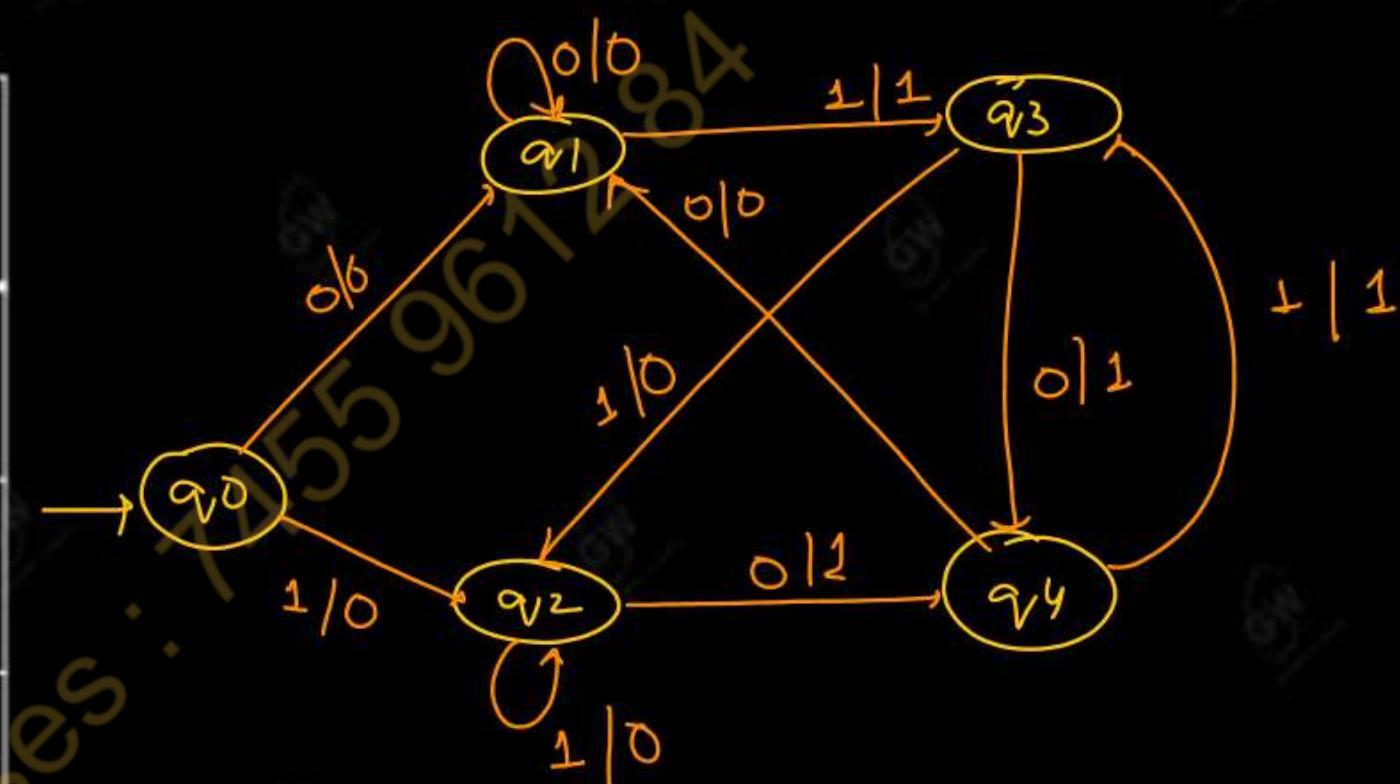
Present state	Next state 0	Next state 1	(output)
$\rightarrow q_0$	q_1	q_2	0
q_1	q_1	q_3	0
q_2	q_4	q_2	0
q_3	q_4	q_2	1
q_4	q_1	q_3	1

Moore transition table

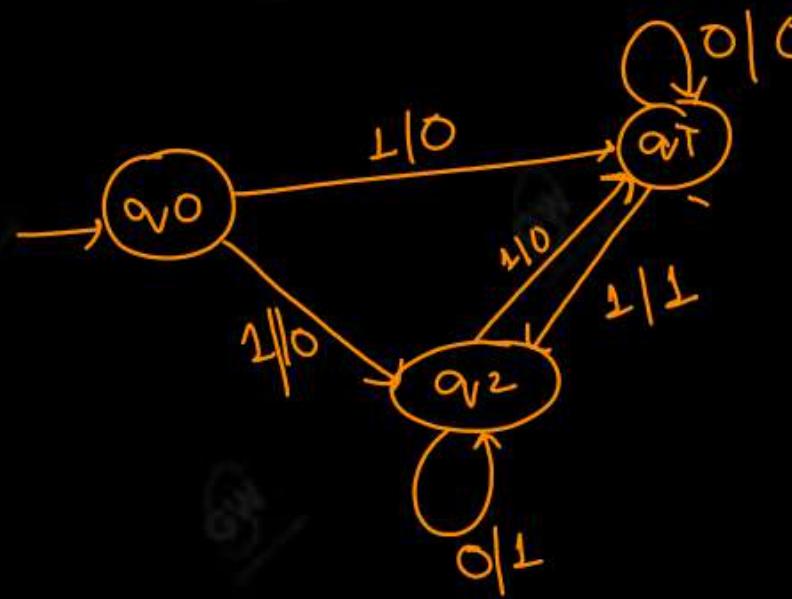
Q3 Moore to mealy machine conversion

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

Mealy machine transition table



Q1 Mealy to Moore conversion

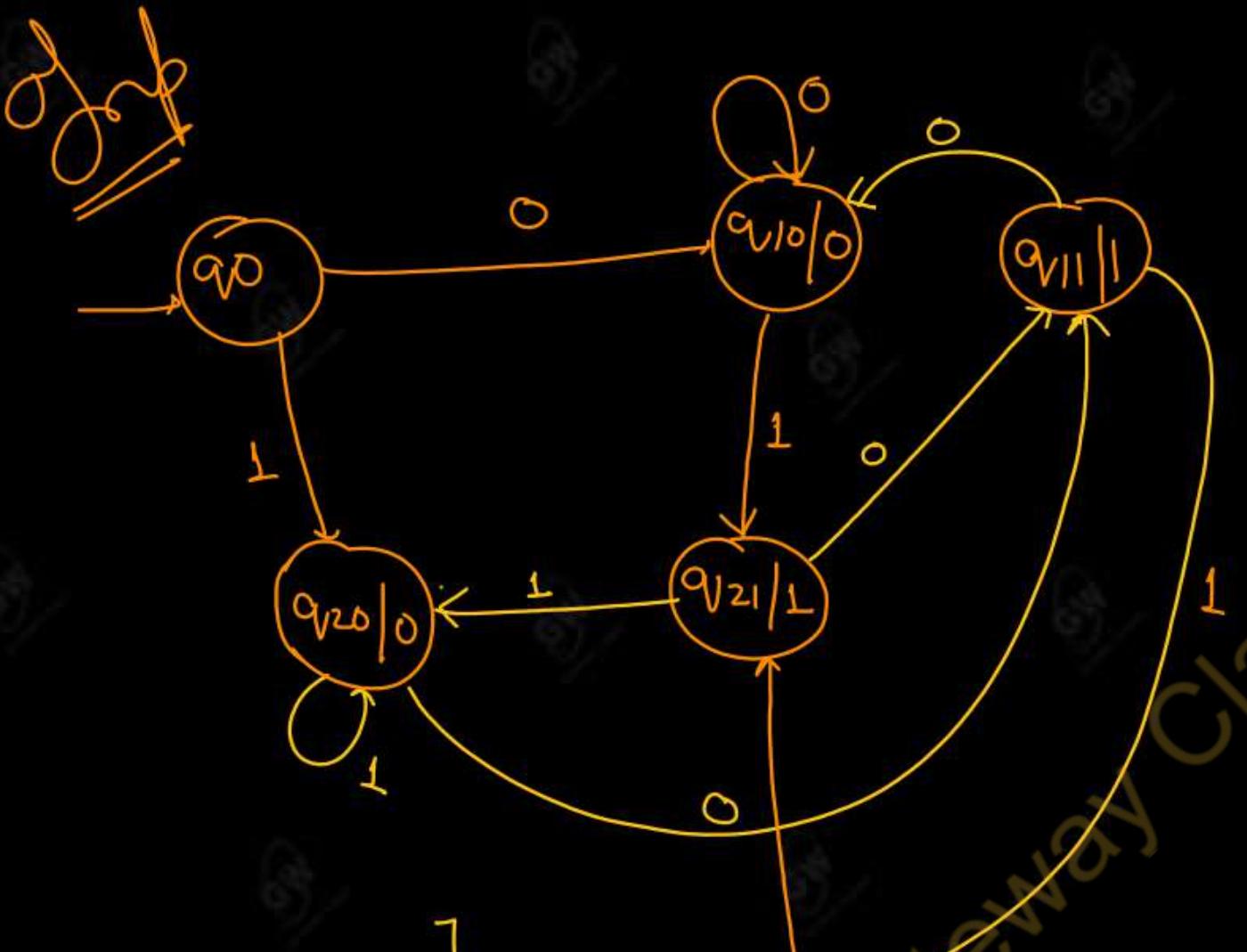


(Mealy Machine)

(Mealy table)

	0 State	output	1 State	output
$\rightarrow q_0$	q_1	0	q_2	0
q_1	q_1	0	q_2	1
q_2	q_2	1	q_1	0

Q1Mealy to Moore conversion



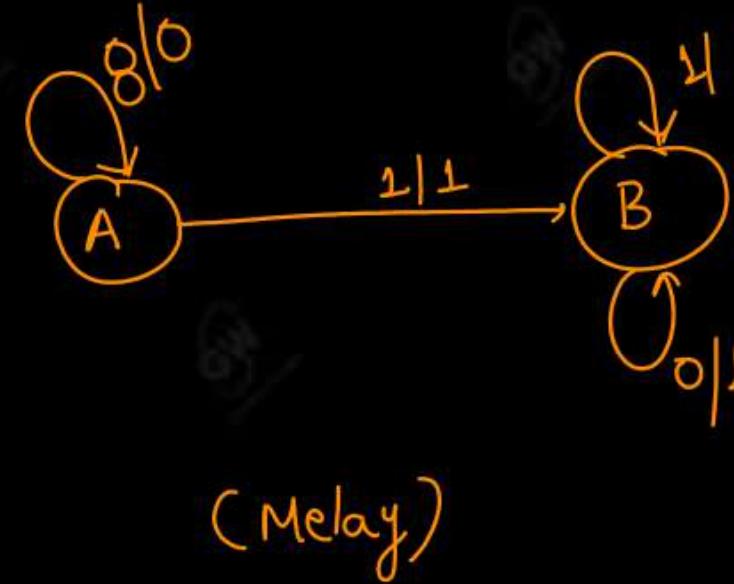
		0 State	output	1 State	output
-> q_0	q_1	0	q_2	0	
q_1	q_1	0	q_2	1	
q_2	q_1	1	q_2	0	

Q1 Mealy to Moore conversion

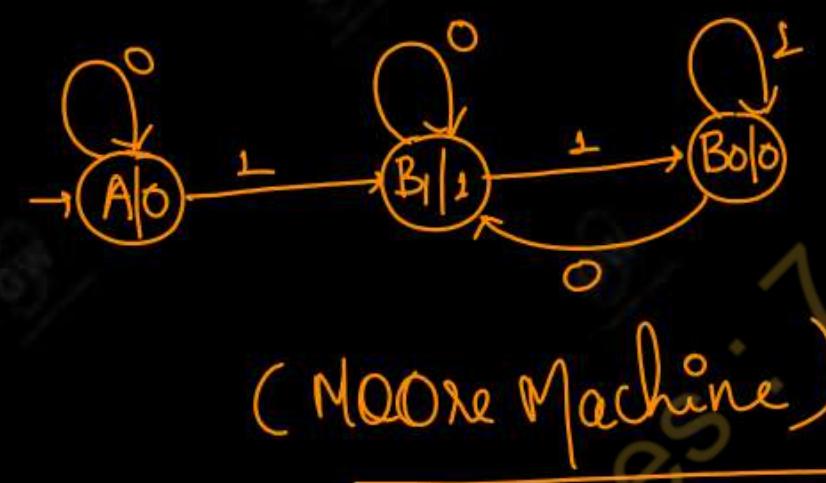
Present state	Next state		(output)
	0	1	
$\rightarrow q_0$	q_{10}	q_{20}	-
q_{10}	q_{10}	q_{21}	0
q_{11}	q_{10}	q_{21}	1
q_{20}	q_{11}	q_{20}	0
q_{21}	q_{11}	q_{20}	1

Moore transition table

Q1 Mealy to Moore conversion



(Delay)



(Moore Machine)

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

Q1 Mealy to Moore conversion

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

Moore transition table

Thank You

Gateway Classes 7455961284