



Gateway Classes

**Semester -IV****CS IT & Allied Branches****BCS402 Theory of Automata and Formal Languages**

UNIT-2 Regular Expressions and Languages



Gateway Series **for Engineering**

Topic Wise Entire Syllabus

Long - Short Questions Covered

AKTU PYQs Covered

DPP

Result Oriented Content

**Download App****For Full Courses including Video Lectures**



Gateway Classes



BCS402 Theory of Automata and Formal Languages

Unit-2

Introduction to Regular Expressions and Languages

Syllabus

Regular Expressions and Languages: Regular Expressions, Transition Graph, Kleen's Theorem, Finite Automata and Regular Expression- Arden's theorem, Algebraic Method Using Arden's Theorem, Regular and Non-Regular Languages- Closure properties of Regular Languages, Pigeonhole Principle, Pumping Lemma, Application of Pumping Lemma, Decidability- Decision properties, Finite Automata and Regular Languages



Download App

For Full Courses including Video Lectures



Regular Expressions and Languages

V

TODAY's TARGET

- Identification of Regular Languages

By PRAGYA RAJVANSI

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

- A regular language is a language that can be expressed with a regular expression or a deterministic or non-deterministic finite automata or state machine.

Whether the below following language are regular or not

- $L = \{a^m b^n \mid m, n \geq 0\}$ yes (able to learn order)
X No comparison, infinite
- $L = \{a^p b^q c^r \mid p, q, r \geq 0\}$ yes Order, No comparison, infinite
- $L = \{a^{x_1}, b^{x_2}, \dots, z^{x_{26}} \mid x_i \geq 0 \mid 1 \leq i \leq 26\}$ yes
- $L = \{a^m b^n \mid 1 \leq m \leq 500, 1 \leq n \leq 1000\}$ yes {language is finite, able to learn order}
- $L = \{a^n b^n \mid n \geq 0\}$ no {able to do counting / comparison, infinite}
- $L = \{a^n b^n \mid 1 \leq n \leq 10\}$ yes finite, Comp
- $L = \{a^n b^n \mid 1 \leq n \leq 2^{29 \text{ prime number}}\}$ yes {comparison, finite}
Counting
- $L = \{a^n b^n \mid 1 \leq n \leq 2^{|gate|}\}$ yes

- $L = \{a^m b^n \mid m > n, n \geq 0\}$ no language is infinite , comparison
- $L = \{a^m b^n \mid m, n \geq 0, m \neq n\}$ { counting / comparison not do in infinite } (No)
- $L = \{a^m b^n \mid m \text{ is divisible by } n\}$ { infinite counting and comparison not possible in infinite } No
- $L = \{a^m b^n \mid m = n^p, p \geq 1\}$ {infinite counting and comparison not possible in infinite} Nb
- $L = \{a^m b^n c^q \mid m = n = q\}$ {infinite counting and comparison not possible in infinite}
- $L = \{a^m b^n c^q \mid m + n = q\}$ {infinite counting and comparison not possible in infinite}
- $L = \{a^m b^n \mid m + n = \text{even}\}$ {infinite , yes}
- $L = \{a^m b^n \mid m + n = \text{odd}\}$ yes
- $L = \{wcw^r \mid w \in \Sigma^*\}$ not regular
- $L = \{wcw \mid w \in \Sigma^*\}$ not regular
- $L = \{wcw^r \mid w \in \Sigma^+\}$ not regular
- $L = \{wcw \mid w \in \Sigma^+\}$ not regular

- $L = \{ww \mid w \in \Sigma^*\}$ no
- $L = \{ww^r \mid w \in \Sigma^+\}$ no
- $L = \{wcw^r \mid c, w \in \Sigma^*\}$ yes
- $L = \{cww^r \mid c, w \in \Sigma^*\}$ yes
- $L = \{ww^rc \mid c, w \in \Sigma^*\}$ yes
- $L = \{wcw^r \mid c, w \in \Sigma^+\}$ yes
- $L = \{cww^r \mid c, w \in \Sigma^+\}$ no
- $L = \{ww^r c \mid c, w \in \Sigma^+ \}$ no
- $L = \{wcw \mid c, w \in \Sigma^*\}$ yes
- $L = \{cww \mid c, w \in \Sigma^*\}$ yes
- $L = \{wwc \mid c, w \in \Sigma^*\}$ yes
- $L = \{wcw \mid c, w \in \Sigma^+\}$ no
- $L = \{cww \mid c, w \in \Sigma^+\}$ no

$L = \{wwc \mid c, w \in \Sigma^+\}$ no

Regular Expression and language

TODAY's TARGET

- Find out the language for the following regular expression
- Find out the regular expression for the following regular language

By PRAGYA RAJVANSI

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

> A way of representing regular language

> Expression of string and operator

> * Kleene closure $[a^*]$

$$\{ \epsilon, a, aa, aaa, \dots \} \rightarrow \{ a^0, a^1, a^2, a^3, a^4, \dots \}$$

> + positive closure $[a^+]$

$$\{ a, aa, aaa, aaaa, \dots \} \rightarrow \{ a^1, a^2, a^3, a^4, \dots \}$$

> . Concatenation $[a.b]$

$$\text{code} \quad cde \cdot ft$$

$\{ \}$

> + union $[a+b]$

$$a+b$$

$$\{ a, b \}$$

> $(a+b).(a+b)$

$$\{ a, b \}$$

> $(a+b)^*$ = $\{ (a+b)^0, (a+b)^1, (a+b)^2, (a+b)^3, \dots \}$

$$\{ \epsilon, a, b, aa, ab, ba, bb, \\ aaa, aab, aba, \dots \}$$

$$(a+b) \downarrow \cdot (a+b) \downarrow = (a+b)^2 - \text{double length all strings}$$

$$\begin{array}{c} aa \\ ab \\ ab \\ ba \\ bb \end{array}$$

Find out the language for the following regular expression

1. $r = a, L(r) = \{a\}$

2. $r = a+b, L(r) = \{a, b\}$

3. $r = a.b, L(r) = \{ab\}$

4. $r = a+b+c, L(r) = \{a, b, c\}$

5. $r = \{ab+a\}.b, L(r) = \{abb, ab\}$

6. $r = a^+, L(r) = \{a, aa, aaa, \dots\}$

7. $r = a^*, L(r) = \{\epsilon, a, aa, aaa, \dots\}$

8. $r = (a+ba).(b+a), L(r) = \{ab, aa, bab, baa\}$

9. $r = \emptyset, L(r) = \emptyset$

10. $r = \epsilon, L(r) = \{\epsilon\}$

11. $r = (a+\epsilon)(b+\emptyset), L(r) = \{ab, b\}$

12. $r = (a+b)^*, L(r) = \{\epsilon, a, b, ab, ba, bb, \dots\}$

13. $r = (a+b)^*(a+b), L(r) = (a+b)^+ = \{a, b, aa, ba, \dots\}$

14. $r = \underline{a^*.a^*}, L(r) = a^* = \{\epsilon, a, aa, aaa, \dots\}$

15. $r = (ab)^*, L(r) = \{\epsilon, ab, abab, ababab, \dots\}$

16. $r = (a+b)^2, L(r) = \{aa, ab, ba, bb\}$
 $(a+b)^3$

$(a+b)(a+b)(a+b)$

$\{aa, ab, ba, bb\} (a+b)$

$\{aaa, aba, baa, bba,$
 $aab, abb, bab, bbb\}$

Find out the regular expression for the following regular language

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

1. L = { start with ab} $\rightarrow ab \cdot (a+b)^*$
2. L = { start with bba} $bba(a+b)^*$
3. L = { end with abb} $(a+b)^*abb$
4. L = { contain a substring aab} $(a+b)^*aab \cdot (a+b)^k$
5. L = { start with and ends with a} $a + a(a+b)^*a$
6. L = { start and end with the same symbol}

$$a+b + a(a+b)^*a + b(a+b)^*b$$
7. L = { start with and ends with the different symbol}

$$a(a+b)^*b + b(a+b)^*a$$
8. L = { |w| = 3 } $(a+b) \cdot (a+b) \cdot (a+b)$

9. L = { |w| >= 3 } $(a+b) \cdot (a+b) \cdot (a+b) \cdot (a+b)^*$
10. L = { |w| <= 3 } $\epsilon + (a+b) + (a+b)^2 + (a+b)^3$

Find out the regular expression for the following regular language

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

1. $L = \{ |w|_a = 2 \} \quad b^* a b^* a b^*$
 $\qquad\qquad\qquad (a+b)^* a (a+b)^* a (a+b)^*$
2. $L = \{ |w|_a \geq 2 \} \quad (a+b)^* a (a+b)^* a (a+b)^*$
3. $L = \{ |w|_a \leq 2 \} \quad b^* + b^* a b^* + b^* a b^* a b^* |$
 $\qquad\qquad\qquad b^* (a+\epsilon) b^* (a+\epsilon) b^*$
4. 3rd symbol from the left end is b $(a+b)^2 b (a+b)^*$
5. 28 the symbol from the right end is a
 $(a+b)^* a (a+b)^{2+}$
6. $|w| = 0 \text{ mod } 3 \quad [(a+b)^3]^*$
7. $|w| = 2 \text{ mod } 3 \quad (a+b)^2 [(a+b)^3]^*$
8. $|w|_b = 0 \text{ mod } 2 \quad a^* (a^* b a^* b a^*)^* |$
 $\qquad\qquad\qquad a^* + (a^* b a^* b a^*)^*$

9. $|w|_a = 1 \text{ mod } 3 \quad b^* a b^* (b a b^* a b^* a b^*)^*$
10. $|w|_b = 2 \text{ mod } 3 \quad a^* b a^* b a^* (a^* b a^* b a^* b a^*)^*$

Find out the regular expression for the following regular language

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

1. Language accepting the string of length 2

$$(a+b) \cdot (a+b) \mid (a+b)^2$$

2. Language accepting the string of length at least 2

$$(a+b) \cdot (a+b) \cdot (a+b)^*$$

3. Language accepting the string of length at most 2

$$\begin{aligned} &\rightarrow \epsilon + a+b+ (a+b) \cdot (a+b) \\ &\rightarrow \epsilon + a+b + (a+b)^2 \\ &\rightarrow (\epsilon+a+b) \cdot (\epsilon+a+b) \end{aligned}$$

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

1. That starts either with 01 or end with 01

$$01(0+1)^* + (0+1)^* 01$$

2. having at least two zero

$$(0+1)^* 0 (0+1)^* 0 (0+1)^*$$

two consecutive zero
 $(0+1)^* 0 0 (0+1)^*$

3. Having two consecutive zero or one

$$(0+1)^* 0 0 (0+1)^* + (0+1)^* 1 (0+1)^*$$

Regular Expression and language

TODAY's TARGET

- Convert regular expression to epsilon NFA
- DPP

By PRAGYA RAJVANSHI

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

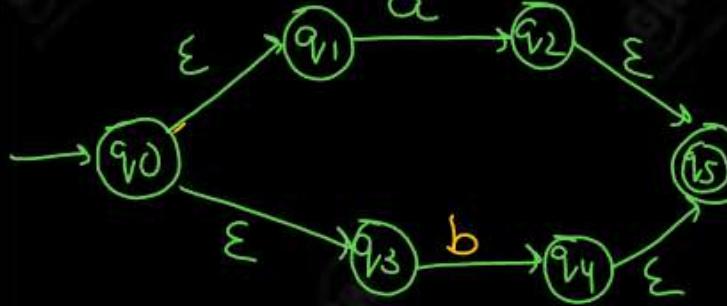
Convert regular expression to epsilon NFA

Φ, ϵ are primitive regular expression

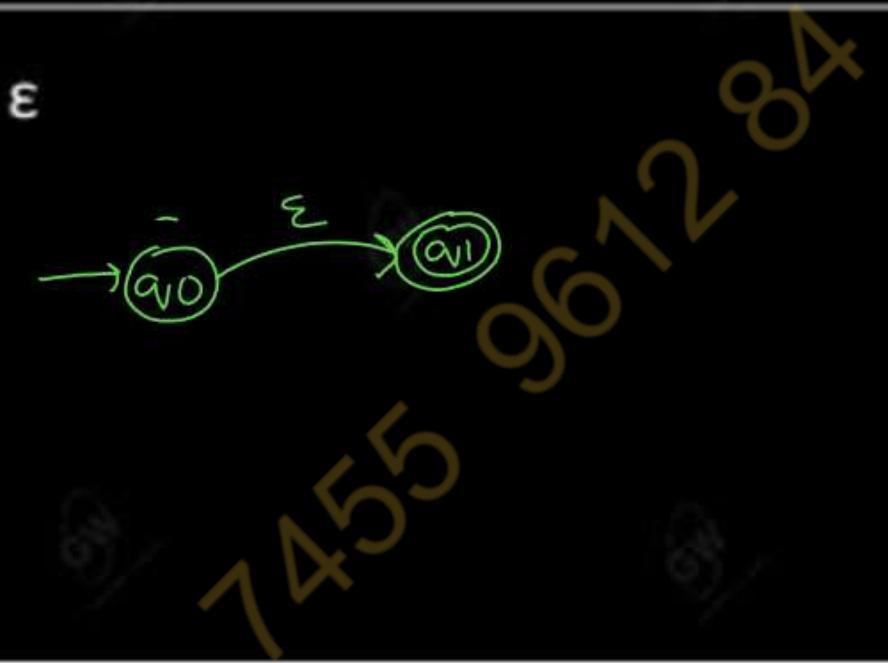
Φ



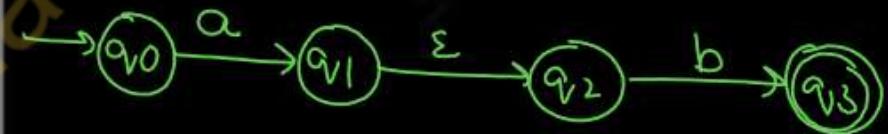
$a+b$

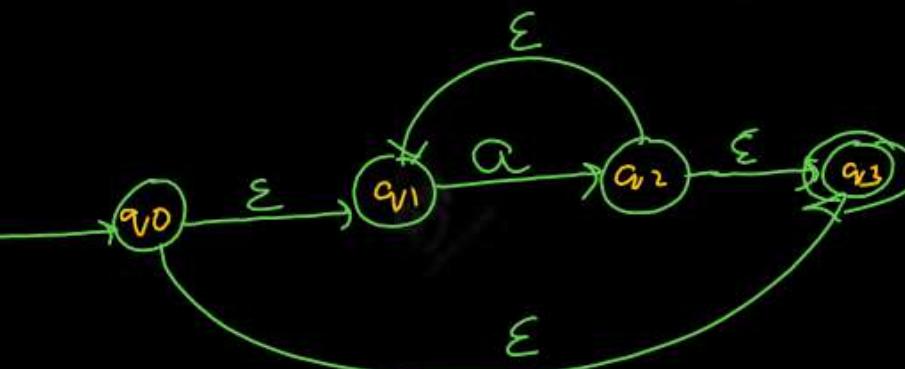


ϵ

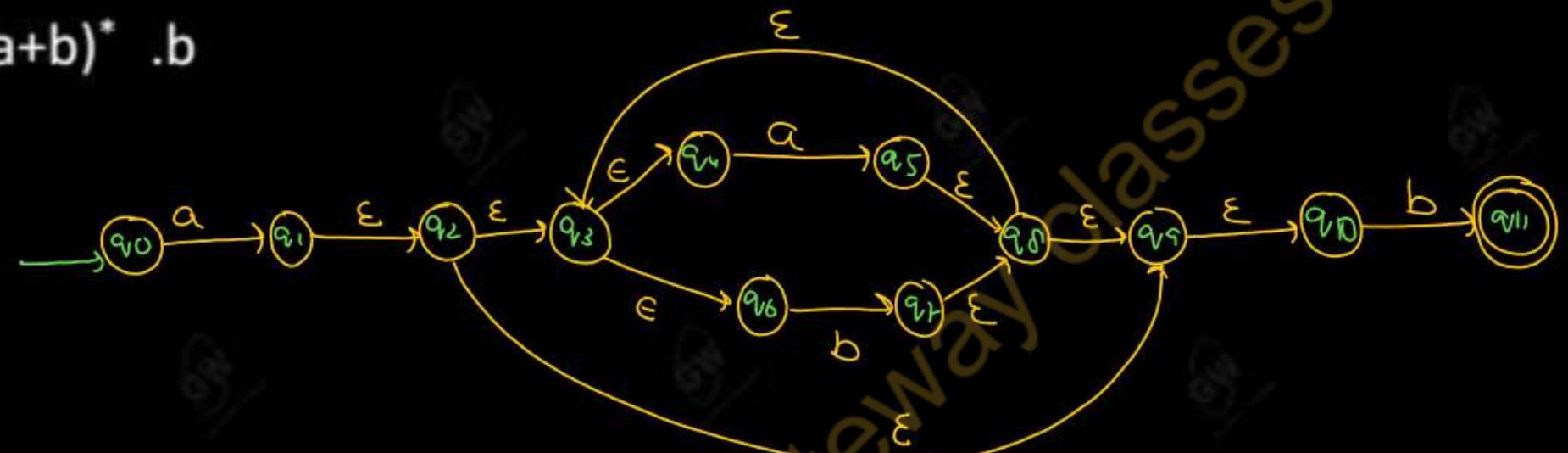


$a.b$

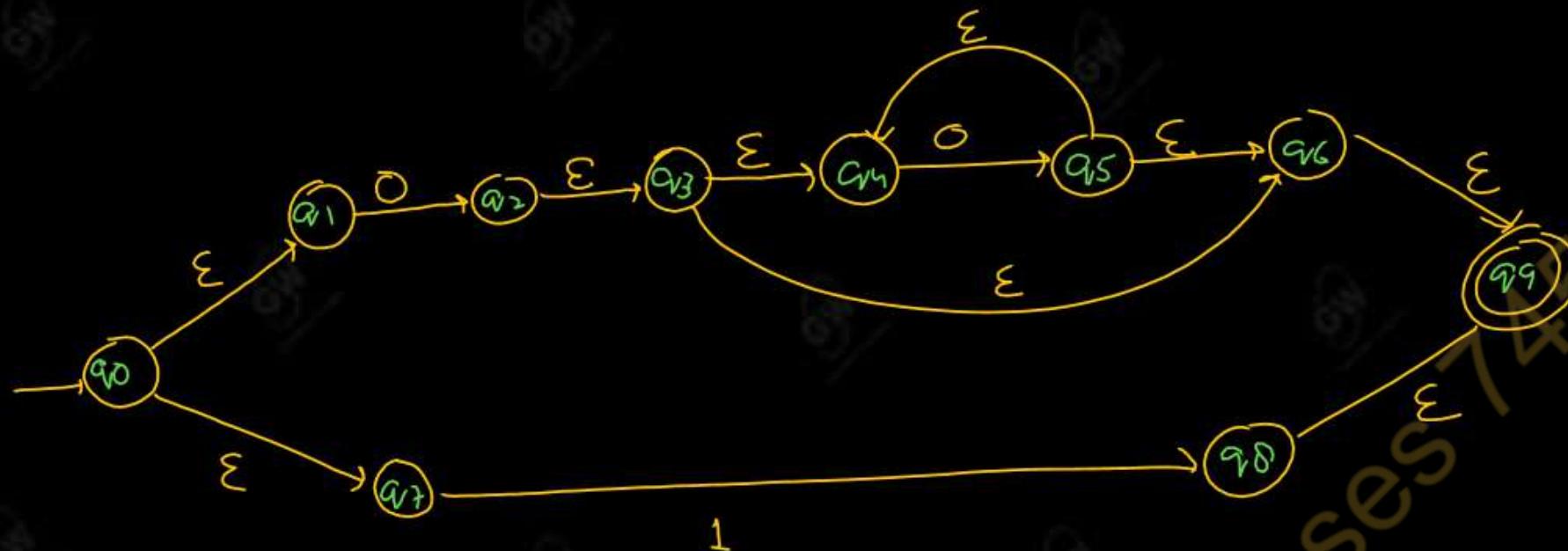




$a.(a+b)^* . b$



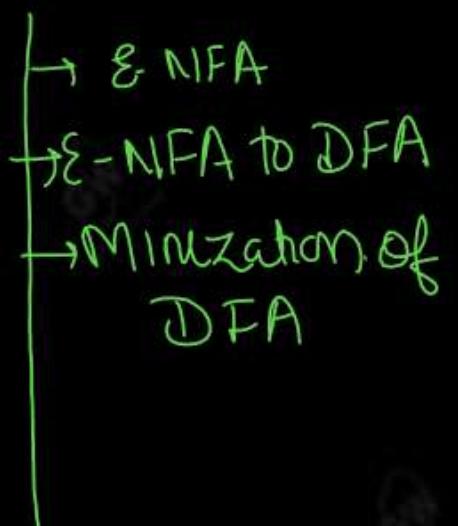
Convert regular expression to epsilon NFA

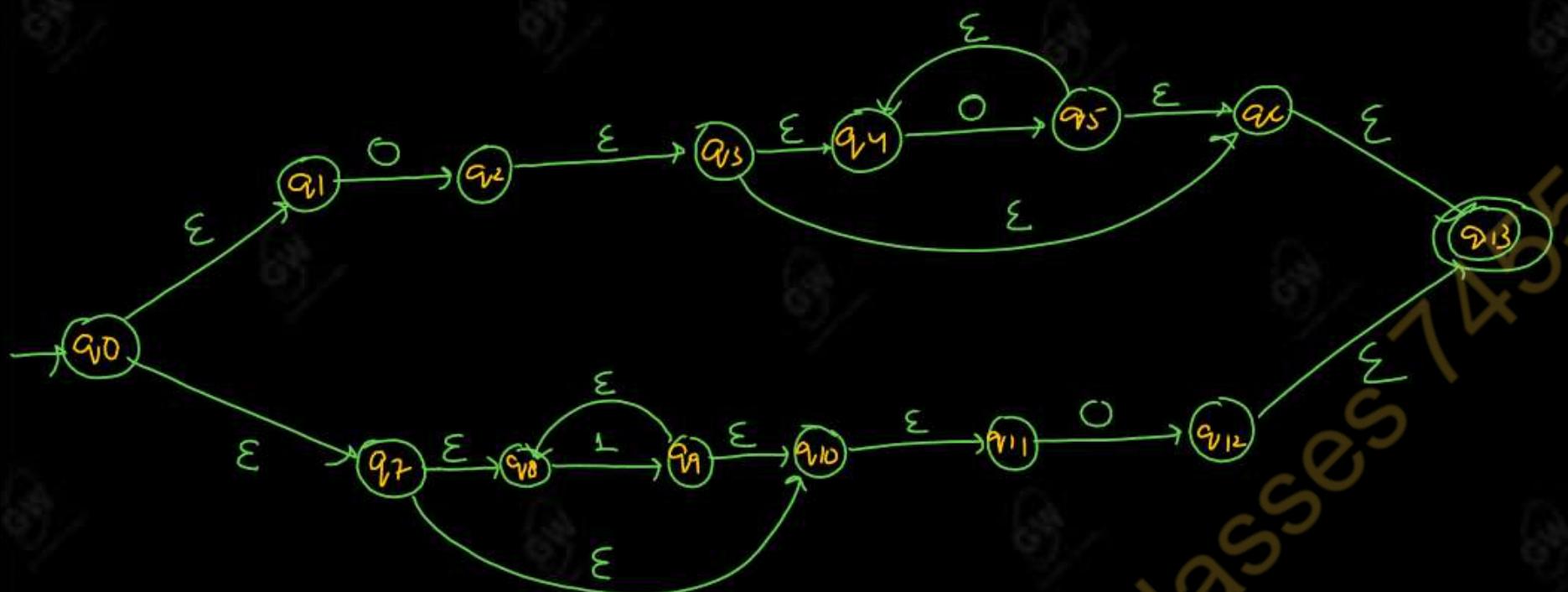
 $00^* + 1$ 

Convert the following Regular Expression
into DFA (Final Automata)

 a^*

- epsilon NFA
- epsilon NFA to DFA
- NFA to DFA
- Minimization of DFA



$00^* + 1^* 0$ 

Gateway Classes 7435961284

Convert regular expression to epsilon NFA

 $(0+1)^* \ 1 \ (0+1)$ 



Convert regular expression to epsilon NFA

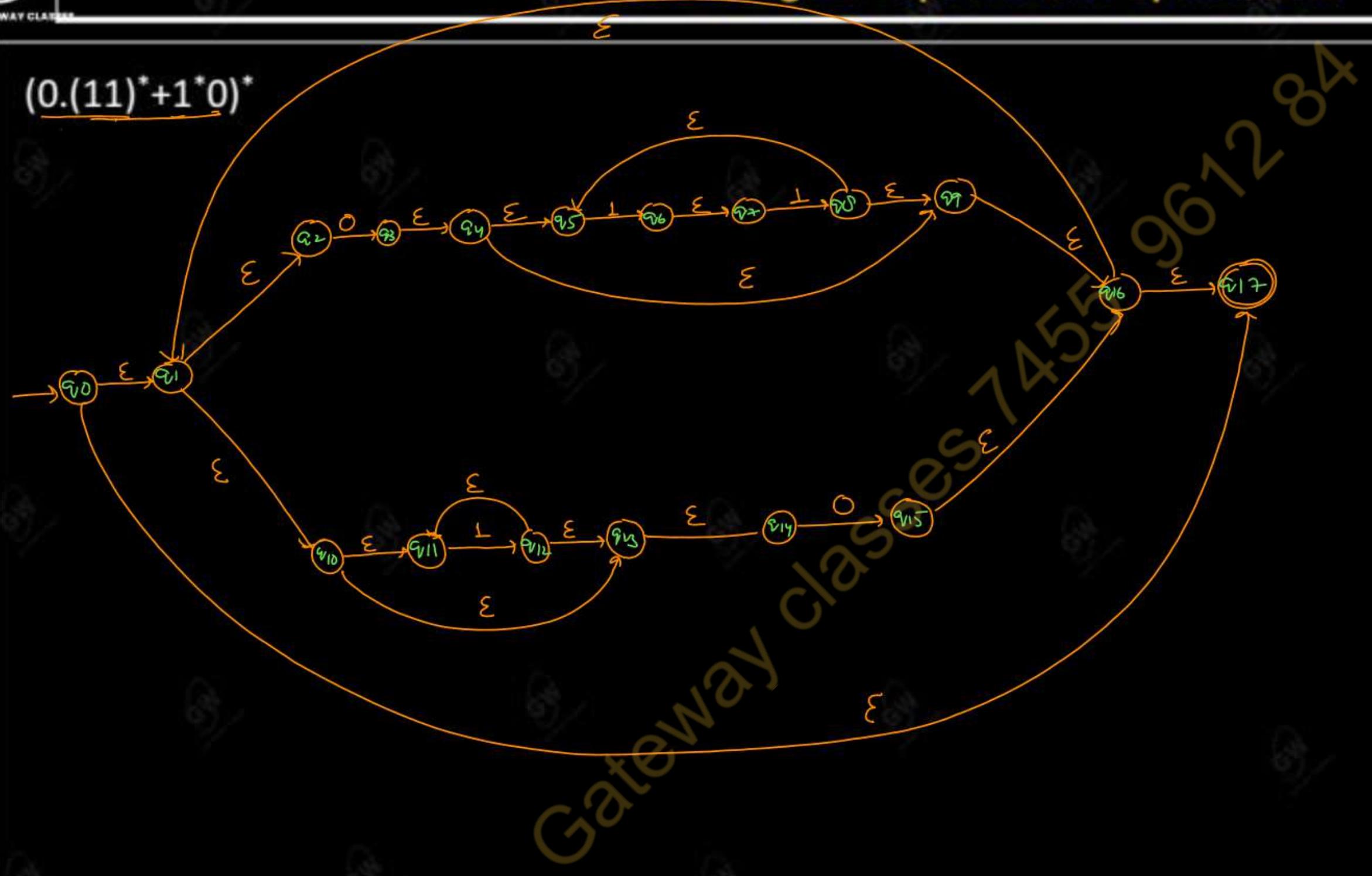
 $(0+1^*)^* 2^* 3^*$ α 

Gateway Classes 7456961284

Convert regular expression to epsilon NFA

 $(a|b)^*abb$ or $(a+b)^*abb$ 

Convert regular expression to epsilon NFA

 $(0.(11)^* + 1^* 0)^*$ 



Convert regular expression to epsilon NFA

 $(00)^*1(11)^*$ 

Gateway Classes 7455 961284

Design the regular expression for the et of all strings{a,b} with the following

- | | |
|---|-------------------------------------------|
| 1 | Having string length of 2 |
| 2 | Zero or more a followed by b |
| 3 | Zero or more instance of a or b $(a+b)^*$ |
| 4 | ending 2 a |
| 5 | Starting with 3 b |
| 6 | with aba as a substring |
| 7 | With a at least one a |
| 8 | At least one a followed by a or b |
| 9 | With three consecutive b at the end |

Construct the epsilon Nfa for the following

1 $(a+b) a(a+b)a(a+b)$

2 $(a^* | b^*)^*$ $(a^* + b^*)^*$

Regular Expression and language

TODAY's TARGET

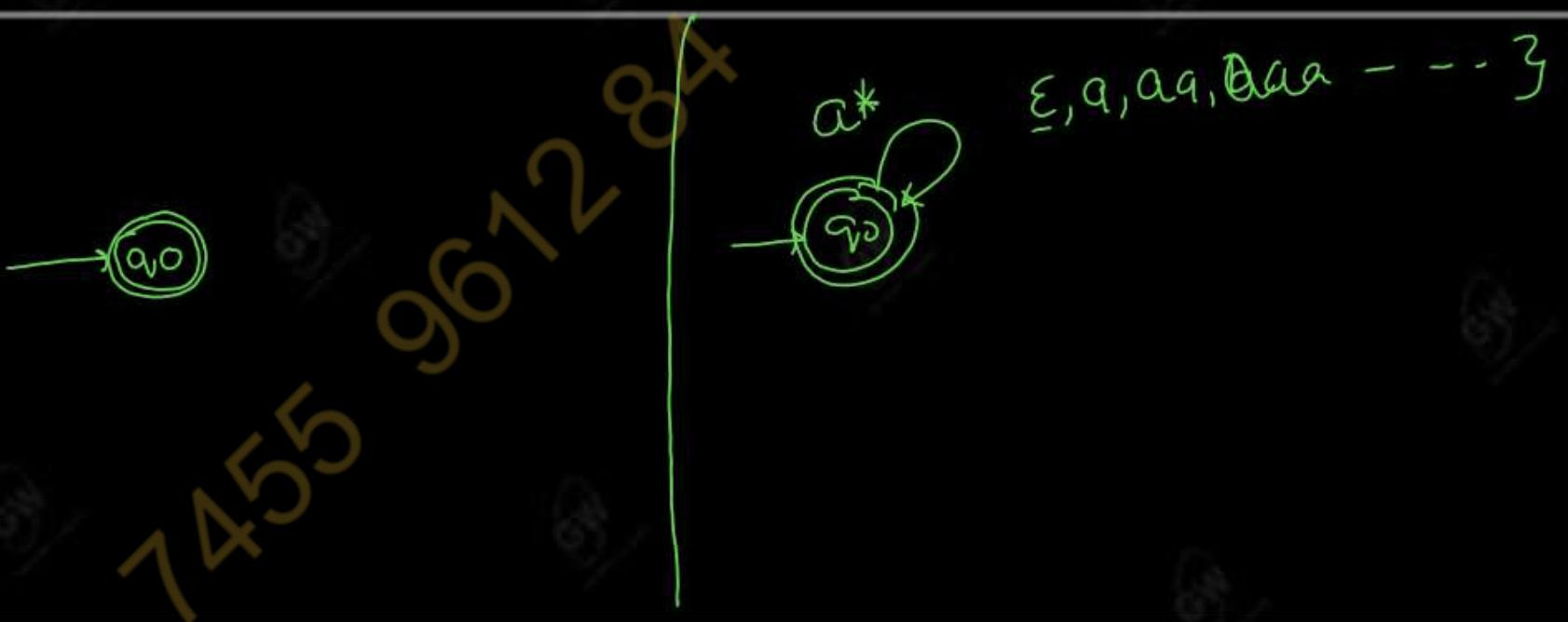
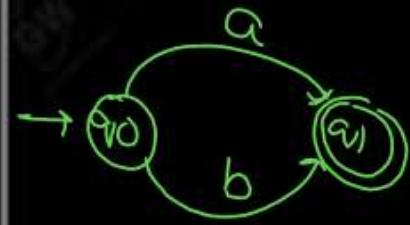
- Convert regular expression to NFA

By PRAGYA RAJVANSI

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

Convert regular expression to NFA

a

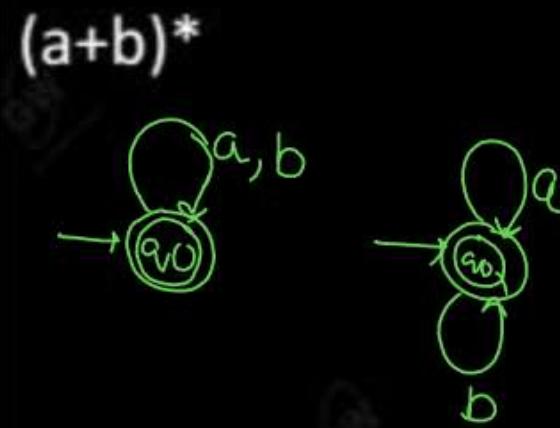
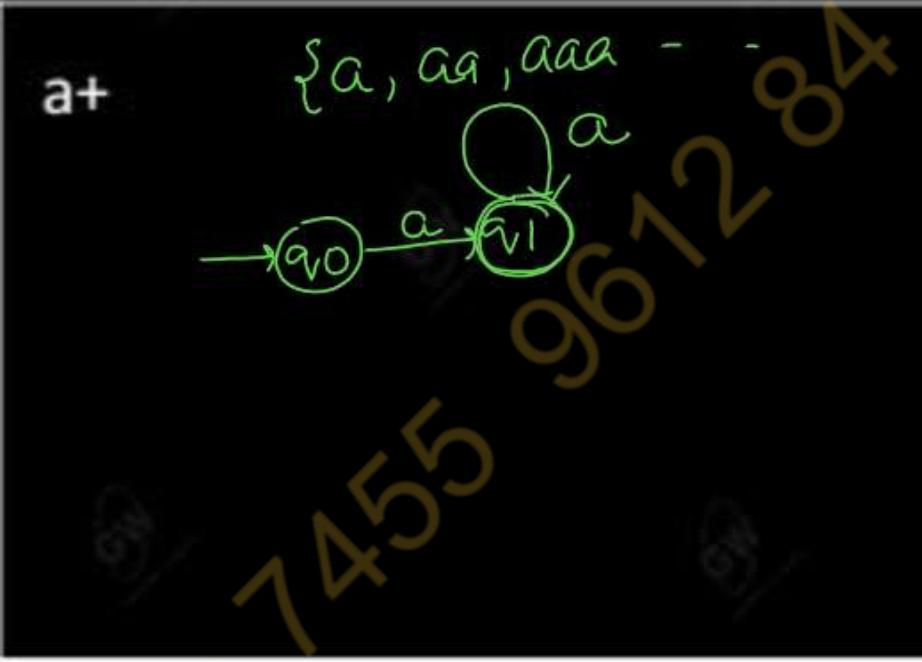
 ϵ  $a+b$ 

a.b

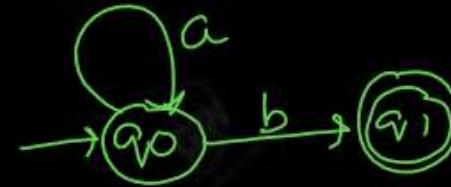
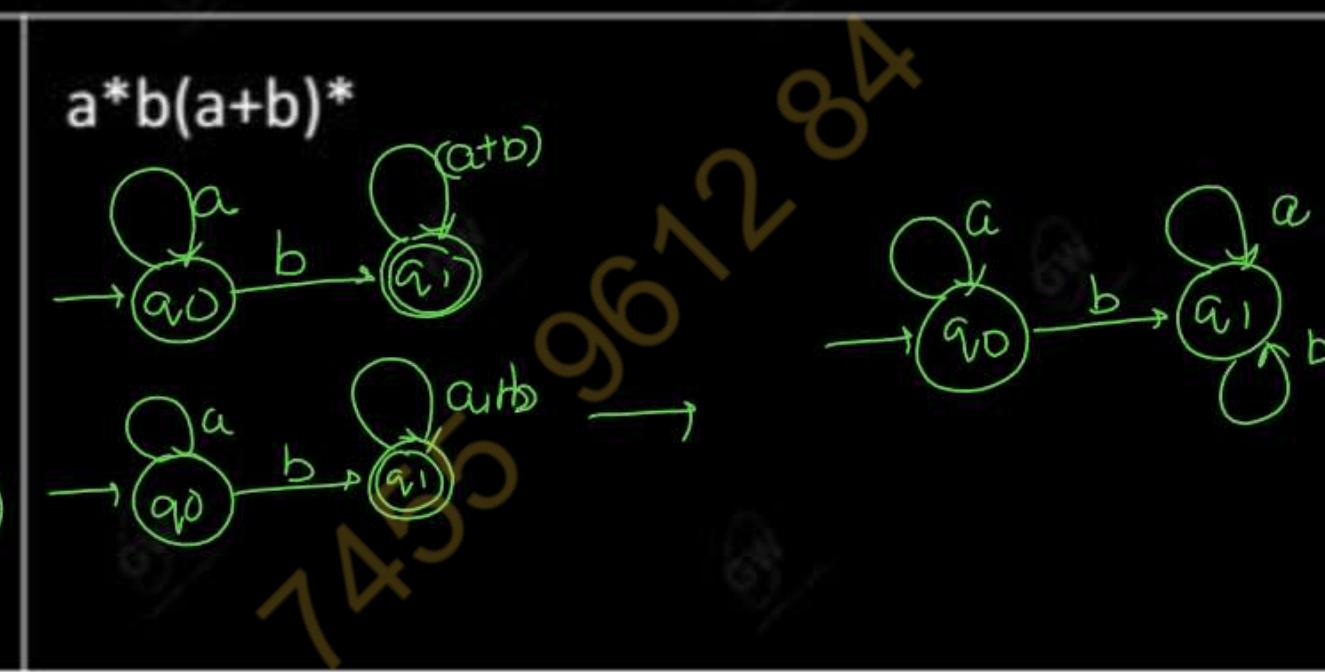
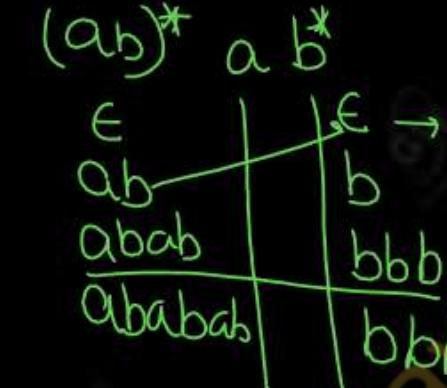
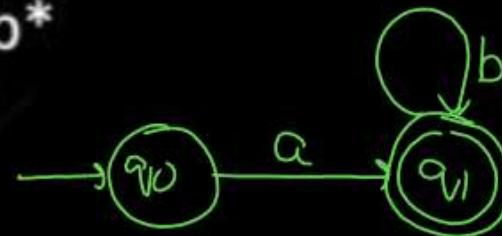
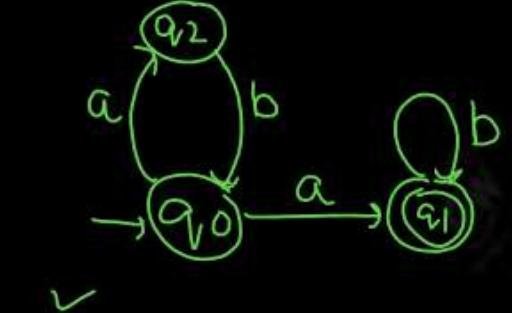


Gateway Classes

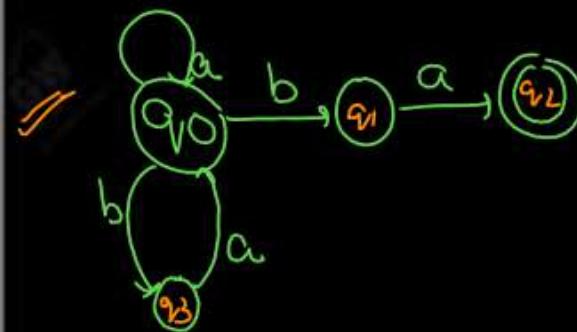
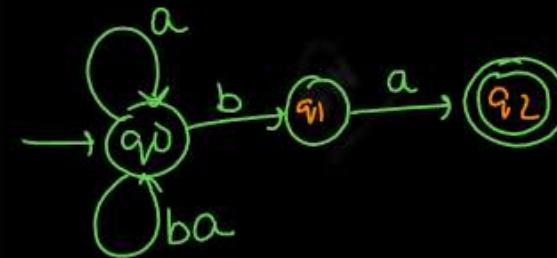
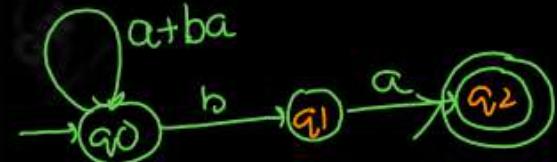
Convert regular expression to NFA



Convert regular expression to NFA

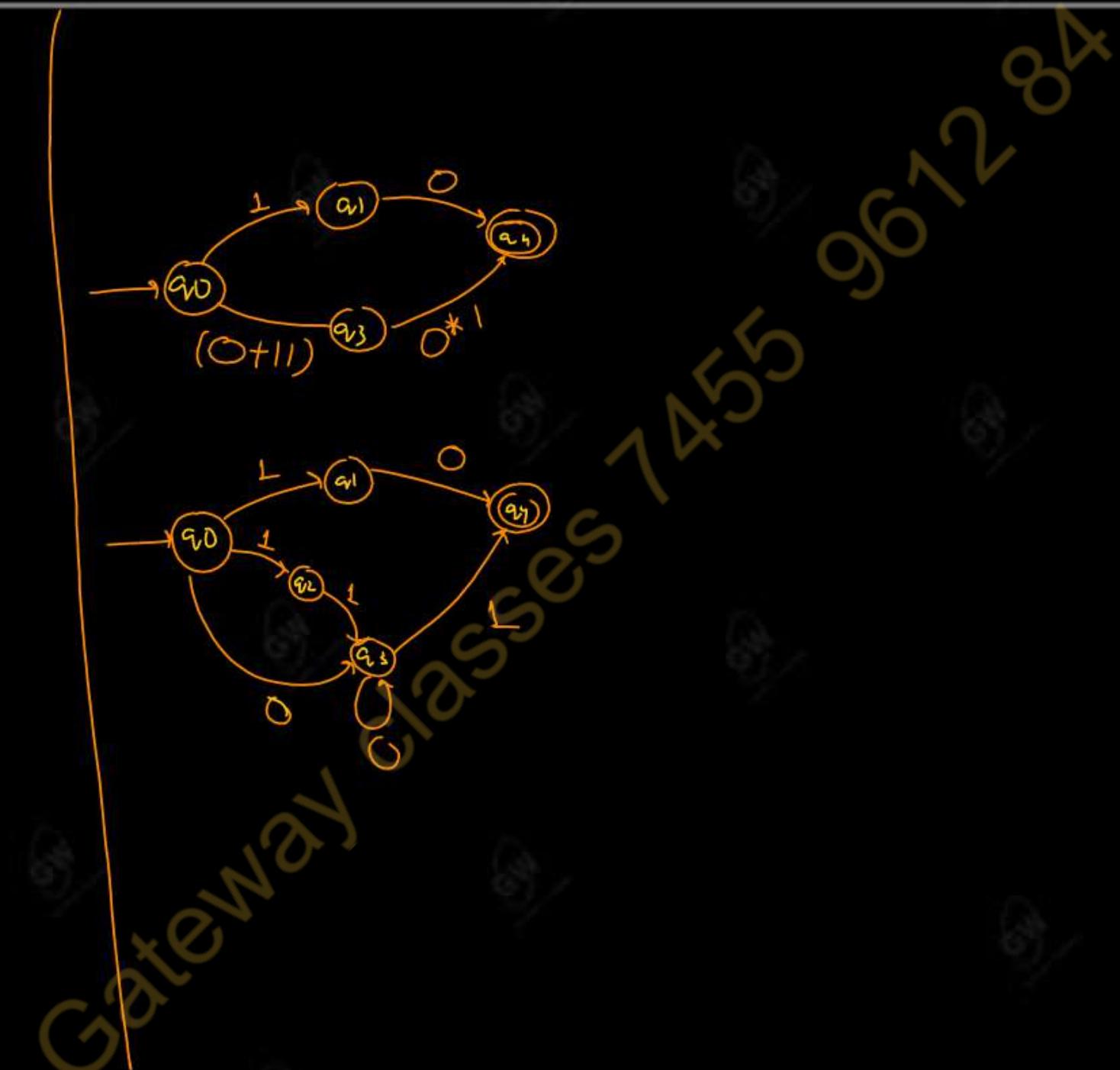
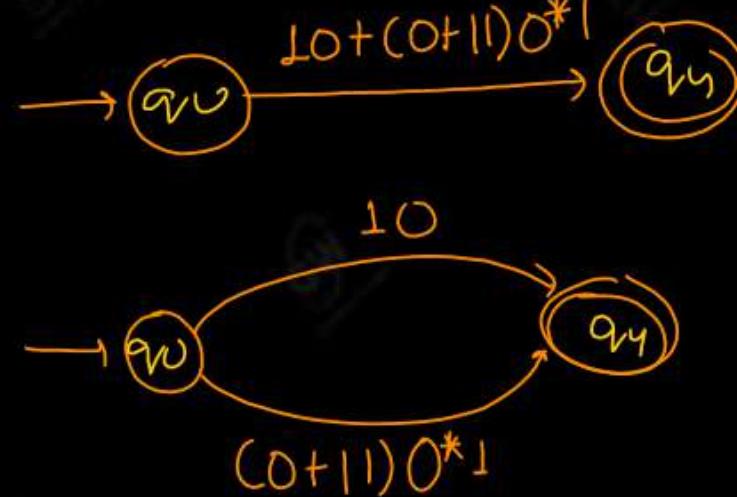
 a^*b  $a^*b(a+b)^*$ ①
② ab^*  $(a.b)^*ab^*$ 

Convert regular expression to NFA

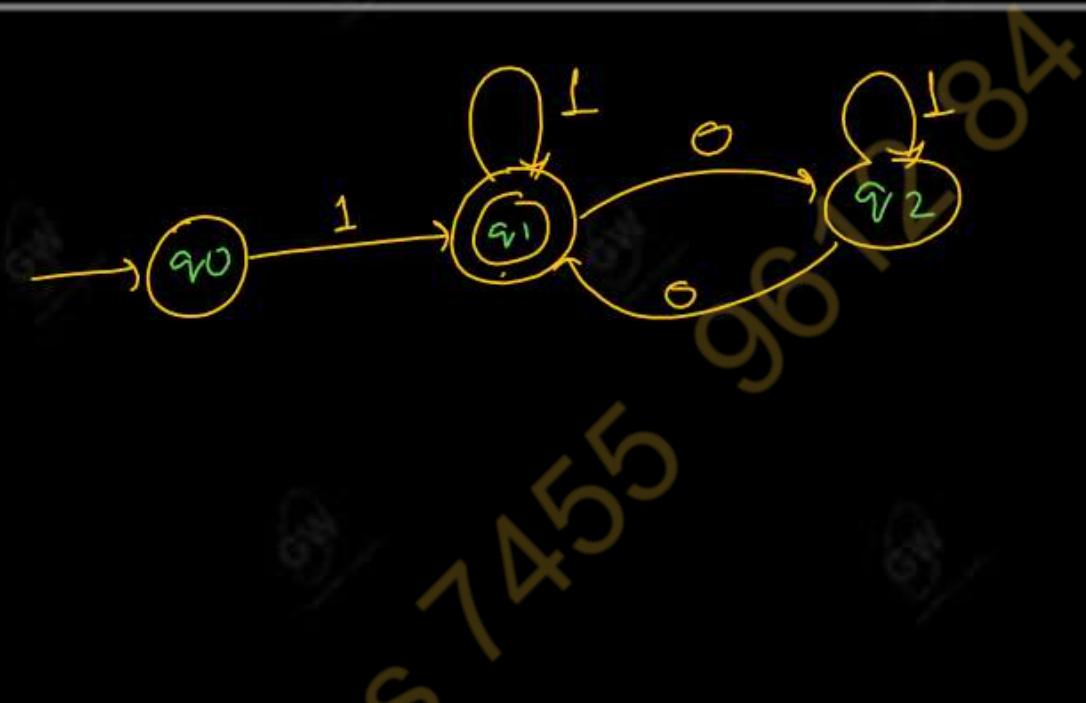
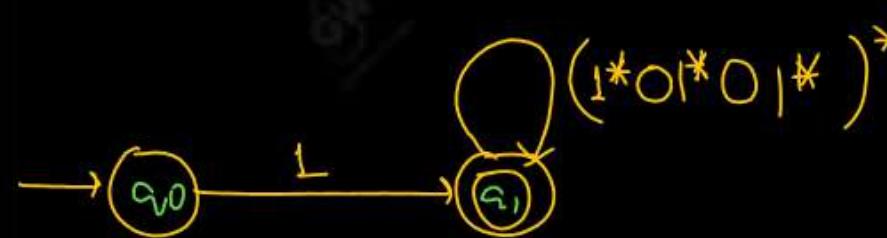
 $(a+ba)^*ba$  $(aa+aaa)^*$ 

Gateway Classes

Convert regular expression to NFA

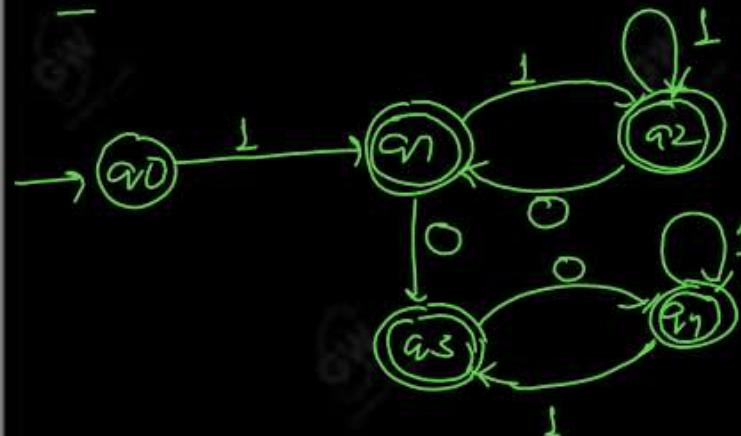
 $10 + (0+11)0^*1$ 

Convert regular expression to NFA

 $1(1^*01^*01^*)^*$ 

Gateway Classes 7455 96784

Convert regular expression to NFA

$$\underline{1}(\underline{1+10})^* + \underline{10}(\underline{0+11})^*$$


Regular Expression and Language

TODAY s' TARGET

- CLOSURE PROPERTIES OF REGULAR LANGUAGE

By PRAGYA RAJVANSI

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

CLOSURE PROPERTIES OF REGULAR LANGUAGE

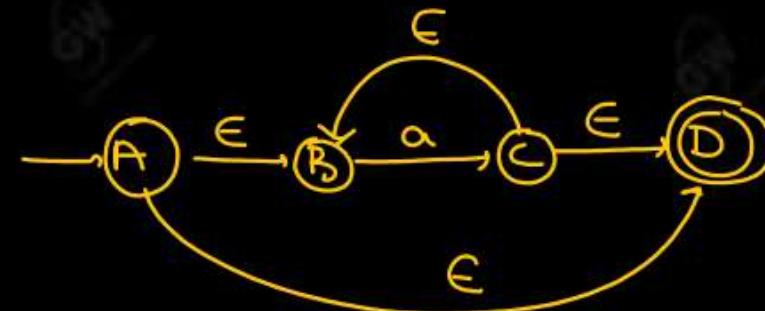
1 The union of two regular language is regular

$$\Sigma = \{\alpha_1 b\} \quad \Sigma^* = \{\epsilon, a, ab, ab, ba, aa, \dots\}$$

$$L_1 = \{\epsilon, a, aa, aaa, aaaa, \dots\}$$

$$R.E \rightarrow a^* (n_1)$$

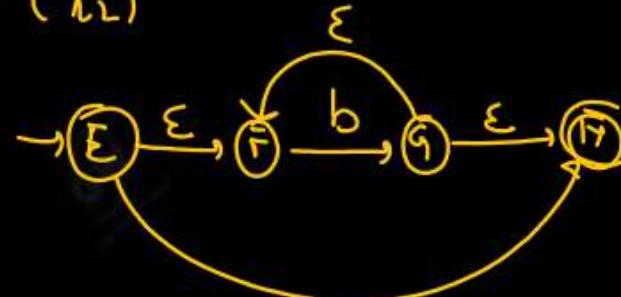
ϵ -NFA



$$L_2 = \{\epsilon, b, bb, bbb, bbbb, \dots\}$$

$$R.E = b^* (n_2)$$

ϵ -NFA



L1 is regular language b/c
We can make ϵ -NFA
R.E

L2 is regular language

$$L_1 \cup L_2 = \text{Regular}$$

$$L_1 \cup L_2 = \{\epsilon_1 + \epsilon_2\}$$

$$a^* + b^* (n_3)$$

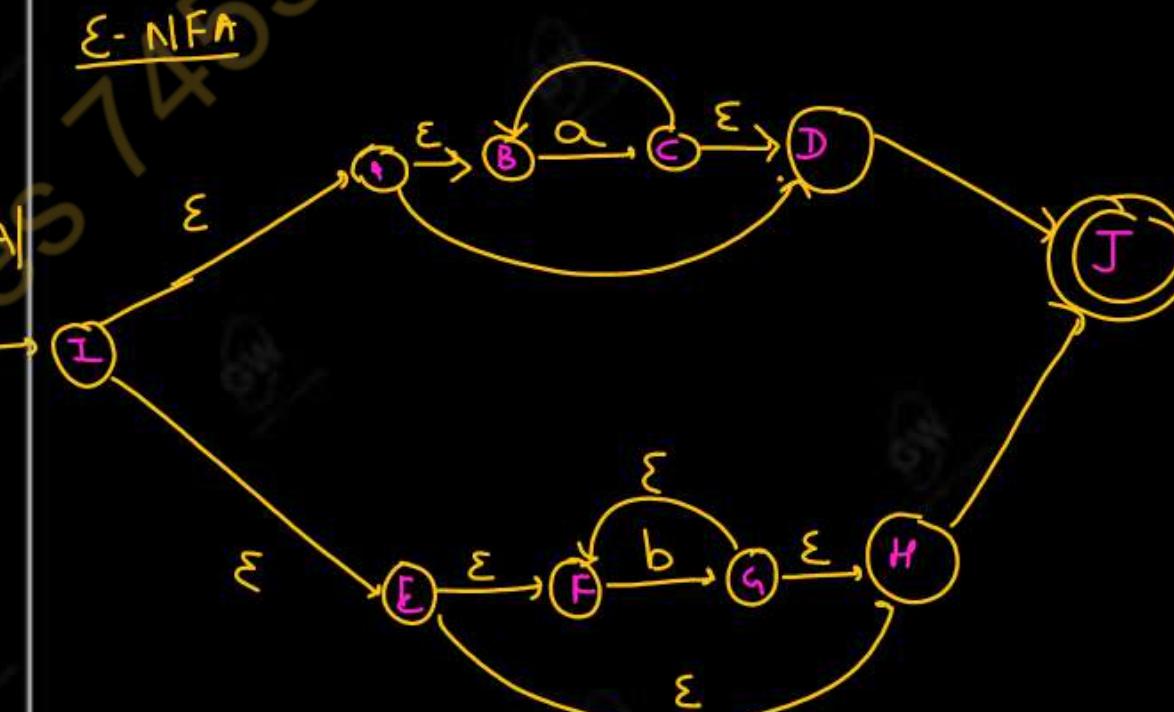
$$L_1 \cup L_2 = \{\epsilon, a, b, aa, bb, aaa, bbb, \dots\}$$

$$\dots$$

→ union of
two Regular
Lang
is Regular

$$L_3 = L_1 + L_2$$

↑
Regular



2. The complement of the regular language is also regular

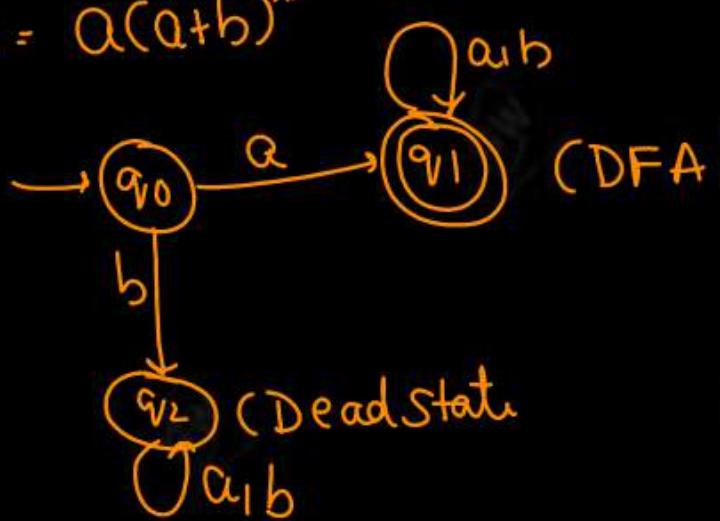
$$\Sigma = \{a, b\}$$

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

$$L_1 = \{a, ab, aab, abaa, aaa, \dots\}$$

$$L_1 = \{w \mid w \text{ starts with } a\}$$

$$R.E = a(a+b)^*$$



L_1 is regular language

$$\overline{L_1} = \{\epsilon, b, ba, bb, bab, \dots\}$$

$\overline{L_1} = \{w \mid w \text{ does not start with } a\}$



DFA
R.E $\epsilon + b(a+b)^*$

Complement of Regular language is Regular.

CLOSURE PROPERTIES OF REGULAR LANGUAGE

3. The intersection of two regular language is also regular

$$L_1 \cap L_2 = \overline{\overline{L}_1 \cup \overline{L}_2}$$

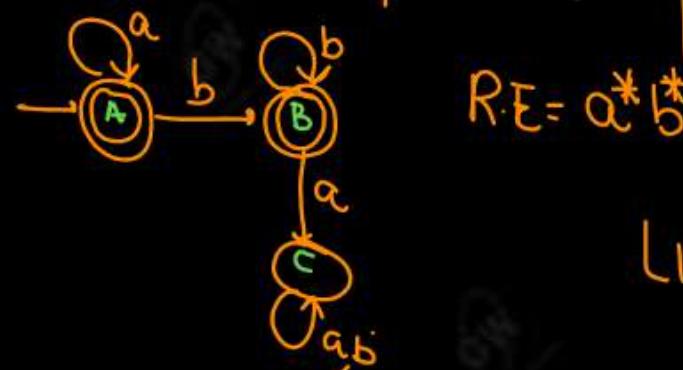
$$A \cdot B = \overline{\overline{A} + \overline{B}}$$

L_1, L_2 is a Regular Language
 $\overline{L}_1, \overline{L}_2$ are complement

$$\overline{\text{Reg} \cup \text{Reg}}$$

$$\overline{\text{Reg} \cup \text{Reg}} = \overline{\text{Reg}} = \text{Reg}$$

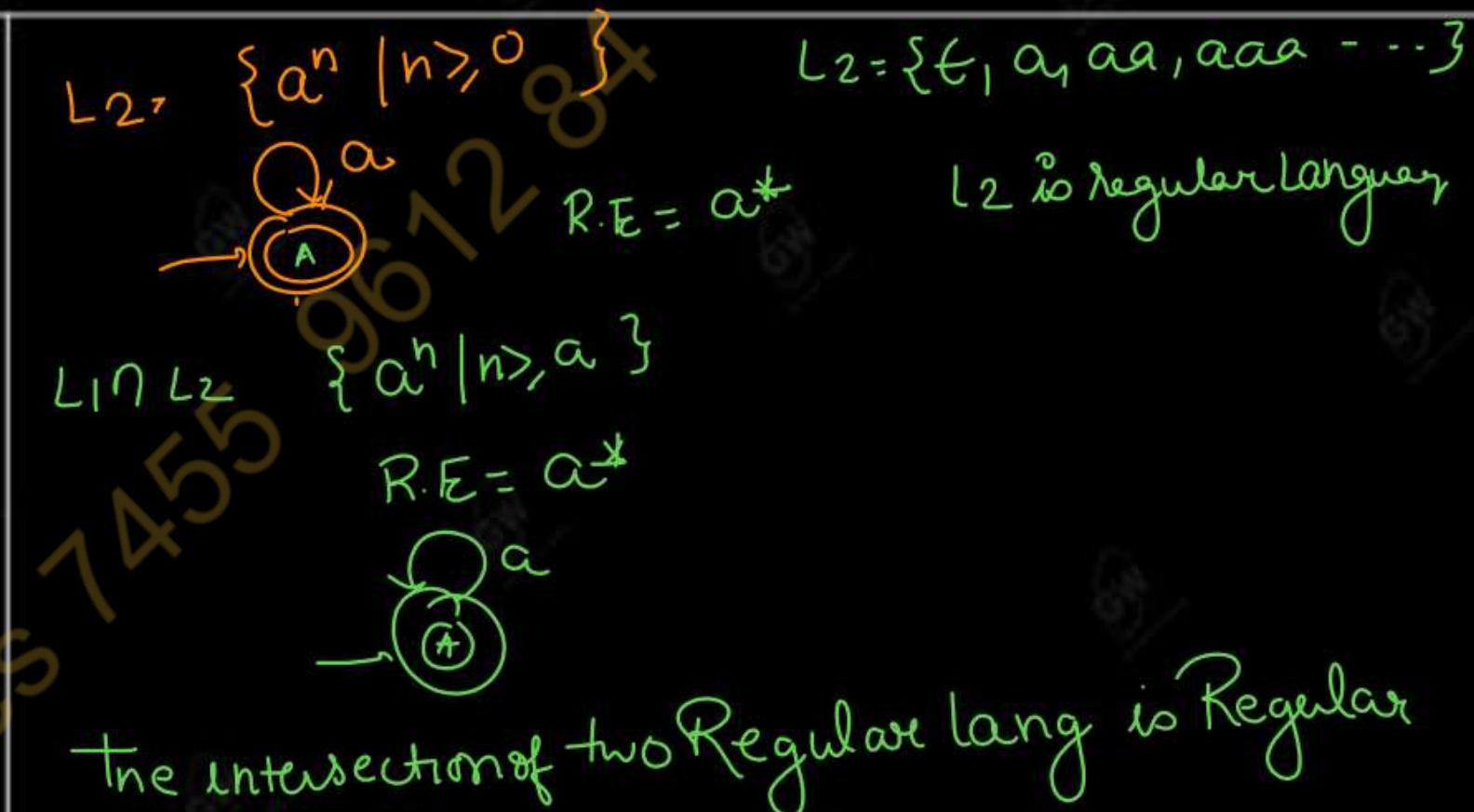
$$L_1 = \{a^n b^m \mid n > 0, m > 0\}$$



$$R.E = a^* b^*$$

L_1 is Regular

$$L = \{ \epsilon, a, aa, \dots, ab, abb, \dots, b, bb, bbb, \dots \}$$



The intersection of two Regular lang is Regular

CLOSURE PROPERTIES OF REGULAR LANGUAGE

4. The difference of two regular language is also regular

$$L_1 - L_2 = L_1 \cap \overline{L_2}$$

$L_1 \& L_2$ is Regular language

\rightarrow Regular \cap Regular

\therefore the complement of Regular lang is Regular

\rightarrow Regular \cap Regular

\rightarrow Regular

$$\Sigma = \{a\}$$

$$L_1 = \{\epsilon, a, aa, aaa, aaaa, \dots\}$$

$$\xrightarrow{\text{S} \xrightarrow{a} \text{S}} \quad RE = a^*$$

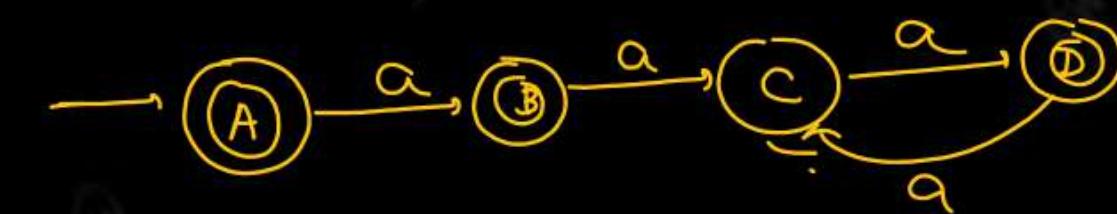
L_1 = Regular language

$$L_2 = \{aa, aaaa, aaaaaa, \dots\}$$

$$(aa)^+ \rightarrow RE$$



$$L_1 - L_2 = \{\epsilon, a, aaa, aaaaa, \dots\}$$



$$\epsilon, a, aaa, aaaaa, \dots$$

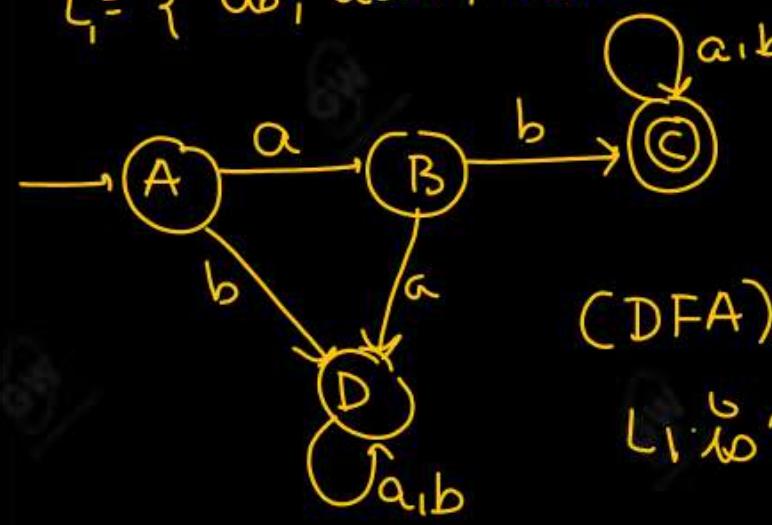
$L_1 - L_2$ - Regular lang

5. The reversal of a regular language is also regular

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

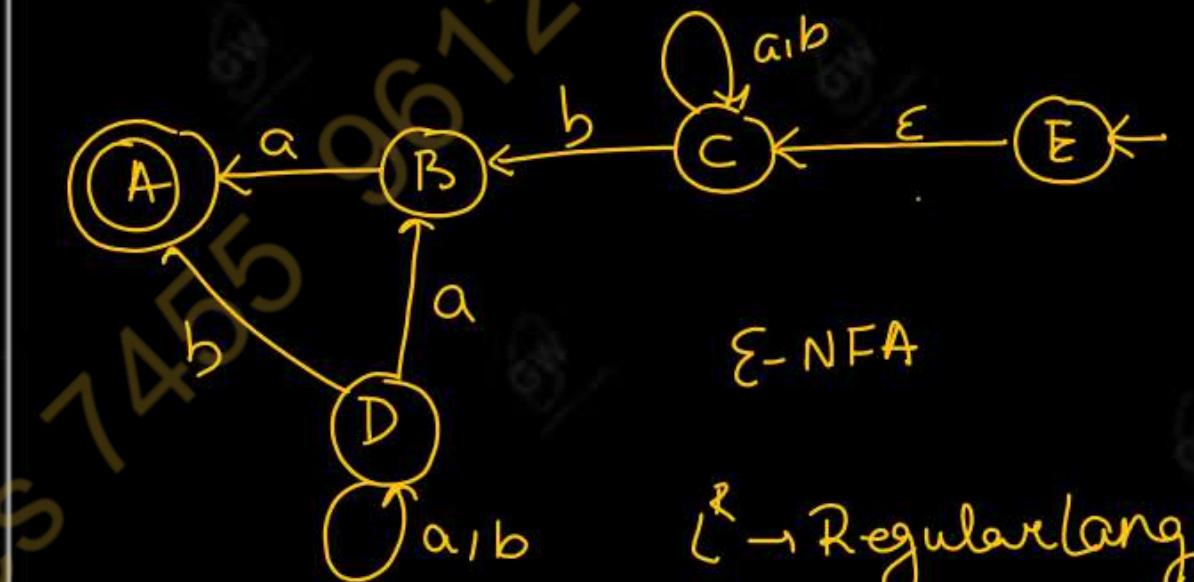
$L_1 = \{ w \mid w \text{ starts with } ab \}$

$L_1 = \{ ab, aba, abaa, \dots, abbabb, \dots \}$



L_1 is Regular

$L^R = \{ ba, ab, ba, \dots \}$



$L^R \rightarrow$ Regular lang

Gateway Classes 7455

6. The Kleene closure of a regular language is also regular

$$L = \{a\}$$

$$\gamma_1 = a$$

$$\gamma_1^* = a^*$$

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



7. The Concatenation of two regular language is also regular

$$L_1 = \{a\}$$

$$L_2 = \{b\}$$

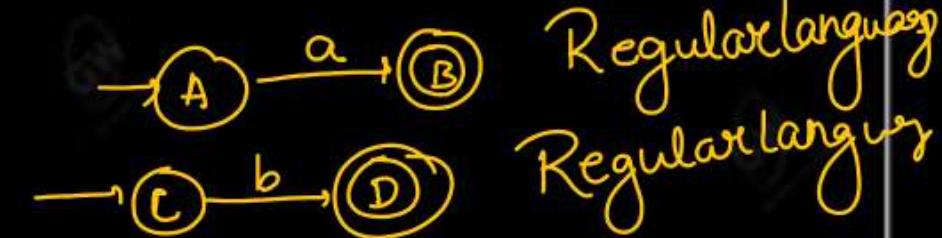
$$\gamma_1 = a$$

$$\gamma_2 = b$$

$$\gamma_1 \cdot \gamma_2 = a \cdot b$$

$$L_3 = \{ab\}$$

L_3 is also regular language



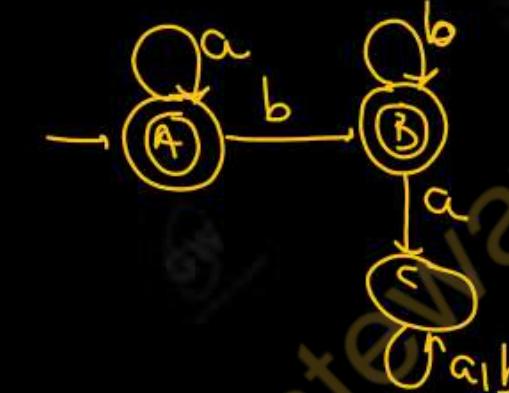
8. A homomorphism (substitution of string for symbol) of a regular language is regular

$$\begin{aligned}
 h: \Sigma &\rightarrow \Gamma^* \\
 \Sigma &= \{a, b\} \\
 \Gamma &= \{0, 1, 2\} \\
 h(a) &= 010 \\
 h(b) &= 102
 \end{aligned}
 \quad
 \left\{
 \begin{aligned}
 h(xy) &= h(x) \cdot h(y) \\
 h(\epsilon) &= \epsilon \\
 h(abba) &= h(a) \cdot h(b) \cdot h(b) \cdot h(a) \\
 &= 010 \quad 102 \quad 102 \quad 010
 \end{aligned}
 \right.$$

$$L_1 = \{a^n b^m \mid n, m \geq 0\}$$

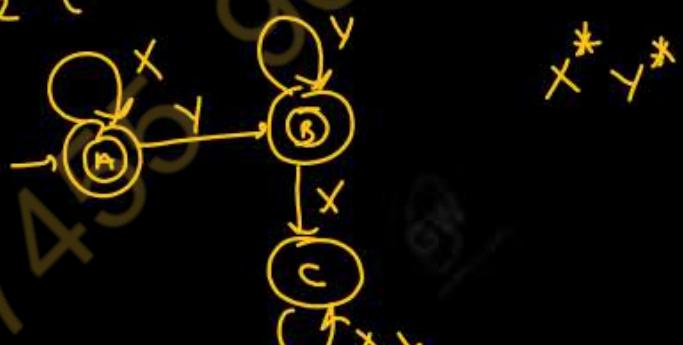
$$a^* b^* - R_E$$

L_1 is Regular



$$\begin{aligned}
 h(a) &= x \\
 h(b) &= y \\
 L_2 &= \{x^n y^m \mid n, m \geq 0\}
 \end{aligned}$$

x^* y^*
 ϵ x y
 x, y
 L_2 is also Regular



8. The inverse homomorphism of a regular language is also regular

$$\Sigma = \{ab\}$$

$$\Gamma^* = \{0, 1, 2\}$$

$$h(a) = 0$$

$$h(b) = 12$$

$$L = \{012120\}$$

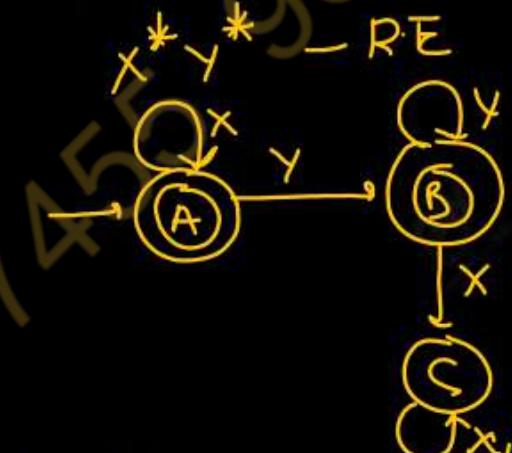
$$h^{-1}(L) = abba$$

h^{-1} is inverse of homomorphism

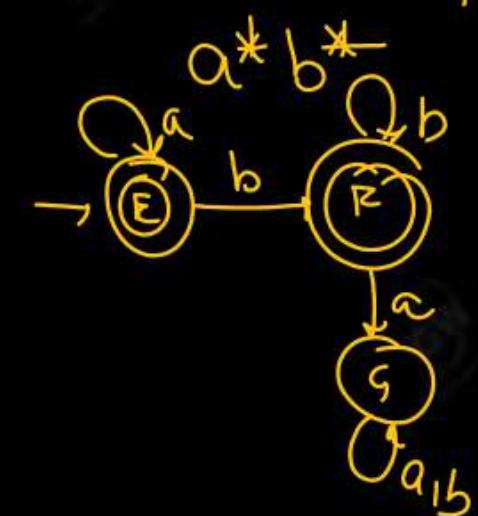
$$h(a) = x$$

$$h(b) = y$$

$$L = \{x^n y^m, n, m \geq 0\}$$



$$h^{-1}(L) = \{a^n b^m, n, m \geq 0\}$$



Regular Expressions and Languages

TODAY's TARGET

- Identities for regular expression

By PRAGYA RAJVANSI

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

Identities for regular expression

1. $\emptyset + r = r$
2. $\emptyset \cdot r = r \cdot \emptyset = \emptyset$
3. $\epsilon \cdot r = r \cdot \epsilon = r$
4. $\epsilon^* = \epsilon$ and $\emptyset^* = \epsilon$ \textcircled{D}

5. $r + r = r$
6. $r^* \cdot r^* = r^*$
7. $r \cdot r^* = r^* \cdot r = r^*$
8. $(r^*)^* = r^*$
9. $\epsilon + r \cdot r^* = r^*$

Identities for regular expression

$$10. (p + q)^* = (p^*.q^*)^* = (p^* + q^*)^*$$

$$11. (p.q)^*.p = p.(q.p)^*$$

$$12. (p+q).r = p.r + q.r \text{ and } r.(p+q) = r.p + r.q$$

Gateway Classes 7455 961284

SOME MORE IMPORTANT PROOF

$$1. \quad r^+ \cup r^* = r^*$$

$$2. \quad r^+ \cap r^* = r^+$$

$$3. \quad (r^*)^+ = (r^*)$$

$$4. \quad (r^+)^* = (r^*)$$

$$5. \quad ((r^*)^+)^*.r^+ = r^+$$

$$6. \quad (a+b)^* = (a^* + b)^*$$

$$7. \quad (a+b)^* = (a + b^*)^*$$

$$8. \quad (a+b)^* \neq (a.b)^*$$

$$9. \quad (a+b)^* \neq (a^*.b)^*$$

$$10. \quad (a+b)^* \neq (a.b^*)^*$$

Proof the following

$$\begin{aligned}
 & \checkmark (1+00*1)+(1+00*1)(0+10*1)*(0+10*1) = (0*1(0+10*1))^* \\
 & (1+00^*1) (\varepsilon + (0+10^*1)^*(0+10^*1)) \\
 & (1+00^*1) (0+10^*1)^* \\
 & (1 \cdot \varepsilon + 00^*1) (0+10^*1)^* \\
 & (\varepsilon + 00^*) \perp (0+10^*1)^* \\
 & \underbrace{\varepsilon + RR^*}_{\varepsilon + RR^*} \\
 & 0^* \perp (0+10^*1)^*
 \end{aligned}$$

$$\begin{aligned}
 & \frac{\varepsilon + 1^*(011)^*}{R} \frac{[1^*(011)^*]^*}{R^*} = (1+011)^* \\
 & [1^*(011)^*]^* \\
 & (P+Q)^* = (P^*Q^*)^* \\
 & - (1+011)^*
 \end{aligned}$$

$$\begin{aligned}
 R &= 1^*(011)^* \\
 R^* &= (1^*(011)^*)^* \\
 \varepsilon + RR^* &= R^* \\
 P &= 1 \\
 P^* &= 1^* \\
 Q &= (011) \\
 Q^* &= (011)^* \\
 (P^*Q^*)^* &= (1^*(011)^*)^*
 \end{aligned}$$

Gateway Classes

Regular Expressions and Languages

TODAY's TARGET

- Arden's Theorem
- Converting finite automata to regular expression

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

It states that-

Let P and Q be two regular expressions over Σ .

➤ If P does not contain a null string ϵ , then-

$R = Q + RP$ has a unique solution i.e. $R = QP^*$

NOTE:

➤ Arden's Theorem is popularly used to convert a given DFA to its regular expression.

Proof

$$\begin{aligned} R &= Q + RP \\ R &= Q + QP^k \cdot P \\ Q(\epsilon + P^k \cdot P) \\ QP^* \end{aligned}$$

this is the solution

$$R = QP^*$$

$$\epsilon + R^* R = R^*$$

$$\begin{aligned} R &= Q + RP \\ &= Q + (Q + RP) \cdot P \\ &= Q + QP + RP^2 \\ &= Q + QP + (Q + RP) \cdot P^2 \\ &= Q + QP + QP^2 + RP^3 \\ &= Q + QP + QP^2 + \dots + QP^n + RP^{n+1} \\ &= Q + QP + QP^2 + \dots + QP^n + QP^* P^{n+1} \\ &= Q(\epsilon + P + P^2 + \dots + P^n + P^* P^{n+1}) \\ &= QP^* \end{aligned}$$

$$R = Q + RP$$

$$R = QP^*$$

this proof this the Unique Solution

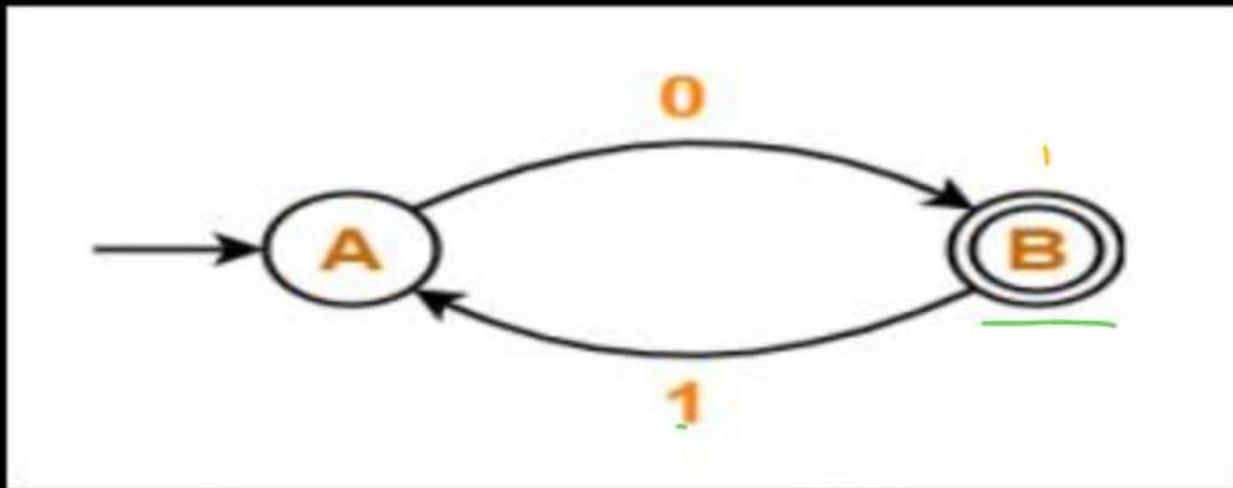
NOTE-

- Arden's Theorem can be used to find a regular expression for both DFA and NFA.
- If there exists multiple final states, then
 - Write a regular expression for each final state separately.
 - Add all the regular expressions to get the final regular expression.

PROCEDURE TO FIND REGULAR EXPRESSION

- Form a equation for each state considering the transitions which comes towards that state.
- Add ' ϵ ' in the equation of initial state.
- Bring final state in the form $R = Q + RP$ to get the required regular expression.

EXAMPLE 1



$$A = \epsilon + B \cdot I \quad \textcircled{1}$$

$$B = A \cdot O \quad \textcircled{2}$$

Put A value in eq²

$$B = (\epsilon + B \cdot I) \cdot O$$

$$B = \epsilon \cdot O + B \cdot I \cdot O$$

$$B = O + B \cdot I \cdot O$$

$$R = O + R \cdot P$$

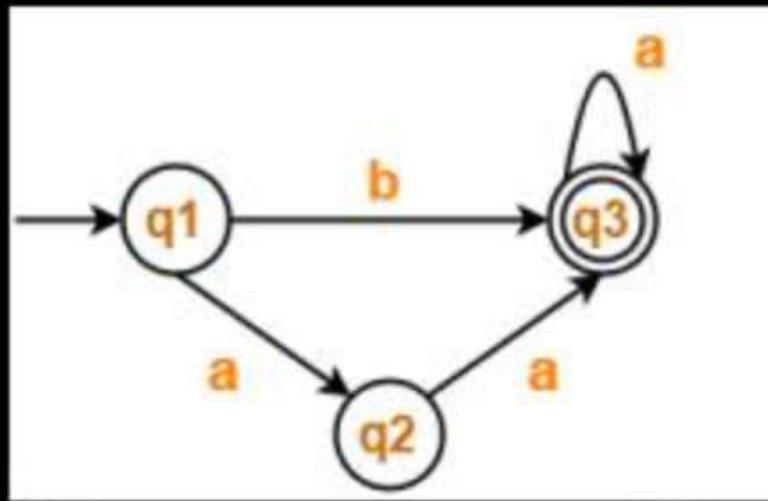
$$R = O + R \cdot P$$

$$O \cdot P^*$$

$$\epsilon \cdot R = R$$

$$|B = O(10)^*|$$

EXAMPLE 2



$$q_1 = \epsilon \xrightarrow{1}$$

$$q_2 = q_1 a \xrightarrow{2}$$

$$q_3 = q_1 \cdot b + q_2 \cdot a + q_3 \cdot a \xrightarrow{3}$$

Put q_1 value in equation 2

$$q_2 = \epsilon \cdot a = a$$

$$\boxed{q_2 = a} \xrightarrow{4}$$

$$R = Q + RP$$

$$R = Q P^*$$

$$q_3 = q_1 \cdot b + q_2 \cdot a + q_3 \cdot a$$

Put q_1 value from eq 1

Put q_2 value from eq 2

$$q_3 = \epsilon \cdot b + a \cdot a + q_3 \cdot a$$

$$b + a \cdot a + q_3 \cdot a$$

$$q_3 = (b + aa) + q_3 \cdot a$$

$$R \quad Q \quad + \quad R \quad P$$

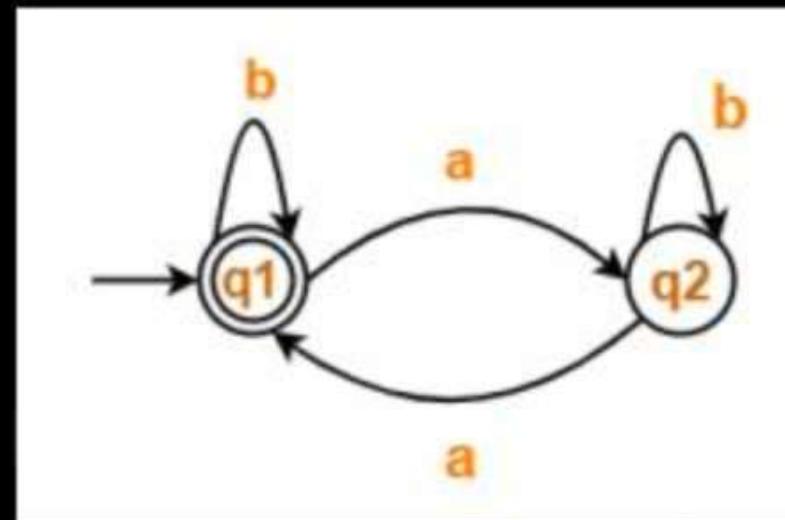
$$q_3 = (b + aa) \cdot a^*$$

$$RE = (b + aa) \cdot a^*$$

$$Q = b + aa$$

$$P = a$$

EXAMPLE 3



$$q_{V1} = \epsilon + q_1 b + q_2 a \quad \text{--- } ①$$

$$q_{V2} = q_1 a + q_2 b \quad \text{--- } ②$$

$$R = \emptyset + R P$$

$$q_2 = q_1 a b^* \quad \text{--- } ③$$

Put q_2 value in eq1

$$q_{V1} = \epsilon + q_1 b + q_2 a$$

$$q_{V1} = \epsilon + q_1 b + q_1 a b^* a$$

$$R = \emptyset + R P$$

$$q_{V1} = \epsilon \cdot (b + a b^* a)^*$$

$$q_{V1} = (b + a b^* a)^*$$

$$RE = (b + a b^* a)^*$$

$$q_{V2} = q_1 a b^*$$

$$(b + a b^*)$$

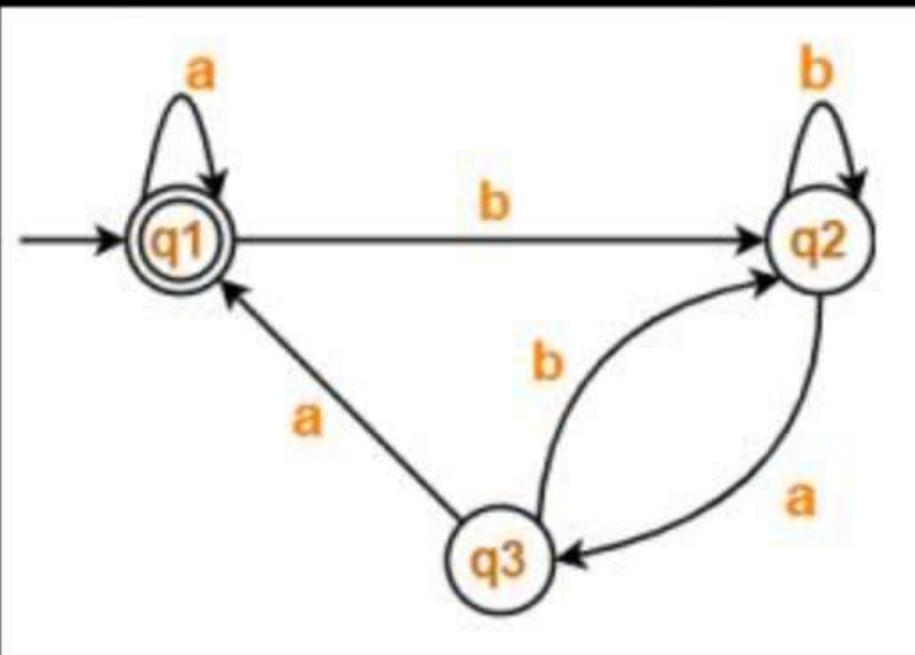
$$a \cdot \epsilon = a$$

$$a \cdot b = ab$$

$$a \cdot b b = ab b$$

$$a b b b = abbb$$

EXAMPLE 4



$$q_1 = \epsilon + q_1 a + q_3 a \quad \text{---} \textcircled{1}$$

$$q_2 = q_1 b + q_2 b + q_3 b \quad \text{---} \textcircled{2}$$

$$q_3 = q_2 a \quad \text{---} \textcircled{3}$$

Put eq₃ in eq₂

$$q_2 = q_1 b + q_2 b + q_3 b$$

$$q_1 b + q_2 b + q_2 a b$$

$$q_2 = q_1 b + q_2 (b + a b)$$

$$R = Q + R P$$

$$q_2 = q_1 b (b + a b)^* \quad \text{---} \textcircled{4}$$

Put equation 4 in eq₃

$$q_3 = q_2 a$$

$$q_3 = q_1 b (b + a b)^* a \quad \text{---} \textcircled{5}$$

Put equation 5 in eq 1

$$q_1 = \epsilon + q_1 a + q_1 b (b + a b)^* a a$$

$$q_1 = \epsilon + (a + b (b + a b)^* a a)$$

$$R = Q + R P$$

$$q_1 = \epsilon \cdot (a + b (b + a b)^* a a)^*$$

$$q_1 = (a + b (b + a b)^* a a)^*$$

RE
 $(a + b (b + a b)^* a a)^*$

Regular Expressions and Languages

TODAY's TARGET

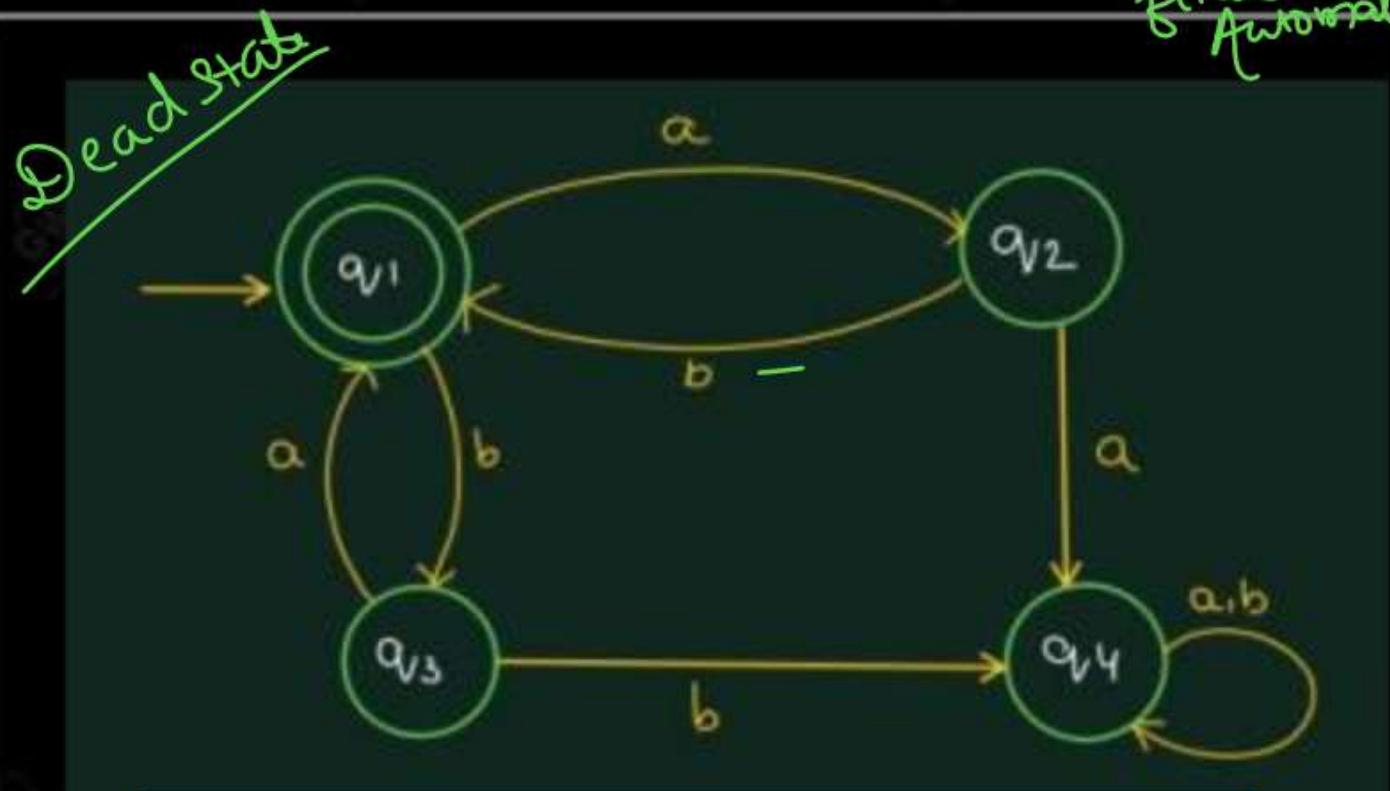
- Converting finite automata to regular expression

By PRAGYA RAJVANSI

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

find out the R.E from the
Given
finite
Automata

EXAMPLE 1



$$\begin{aligned} q_{V1} &= \epsilon + q_{V2}b + q_{V3}a \quad \text{--- (1)} \\ q_{V2} &= q_{V1}a \quad \text{--- (2)} \\ q_{V3} &= q_{V1}b \quad \text{--- (3)} \\ q_{V4} &= q_{V2}a + q_{V3}b + q_{V4}a+b \quad \text{--- (4)} \end{aligned}$$

Put q_{V2} & q_{V3} value in equation 1

$$q_{V1} = \epsilon + q_{V1}ab + q_{V1}ba$$

$$q_{V1} = \epsilon + q_{V1}(ab+ba)$$

~~$$R = \emptyset + R^P$$~~

~~$$q_{V1} = \epsilon \cdot (ab+ba)^*$$~~

~~$$q_{V1} = (ab+ba)^*$$~~

$$\boxed{R.E = (ab+ba)^*}$$

EXAMPLE 2



$$\begin{aligned} q_{V1} &= \epsilon + q_{V1}0 \quad \text{--- (1)} \\ q_{V2} &= q_{V1}1 + q_{V2}1 \quad \text{--- (2)} \\ q_{V3} &= q_{V2}0 + q_{V3}0 + q_{V3}1 \quad \text{--- (3)} \end{aligned}$$

if q_1 is my final State

$$\begin{aligned} q_{V1} &= \epsilon + q_{V1}0 \\ R &= Q + RP \end{aligned}$$

$$q_{V1} = \epsilon \cdot 0^*$$

$$q_{V1} = 0^*$$

$$R \cdot E_1 = 0^*$$

if q_1, q_2 are my both final state

$$q_{V2} = q_{V1}1 + q_{V2}1$$

$$q_{V2} = 0^*1 + q_{V2}1$$

$$R = Q + RP$$

$$q_{V2} = 0^*11^*$$

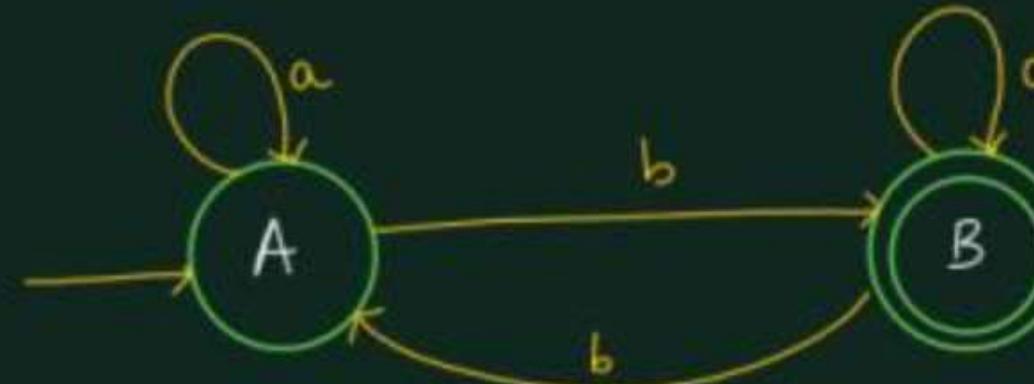
$$RE_2 = 0^*11^*$$

$$\begin{aligned} RE &= RE_1 + RE_2 \\ 0^* + 0^*11^* \\ 0^*(\epsilon + 11^*) \end{aligned}$$

$$0^*1^*$$

$$\begin{aligned} \epsilon + RR^* &= R^* \\ C + R^*R &= R^* \end{aligned}$$

EXAMPLE 3



$$A = \epsilon + Aa + Bb \quad \text{--- } ①$$

$$B = Ab + Ba \quad \text{--- } ②$$

$$R = \emptyset + RP$$

$$B = Ab\alpha^* \quad \text{--- } ④$$

$$A = \epsilon + Aa + Bb$$

$$A = \epsilon + Aa + Aba^*b$$

$$A = \epsilon + A(a + ba^*b)$$

$$R = \emptyset + RP$$

$$A = (\epsilon + a + ba^*b)^*$$

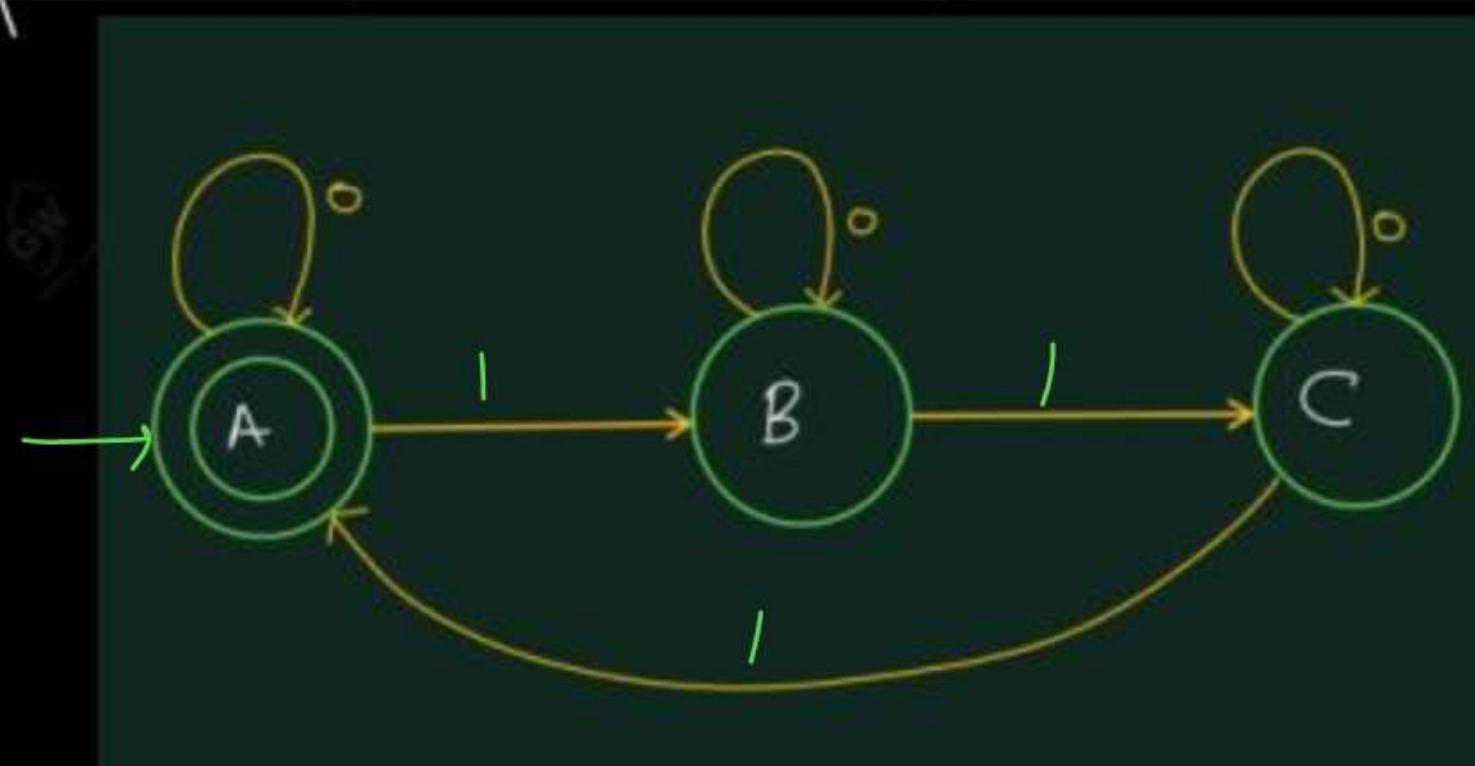
$$A = (a + ba^*b)^* \quad \text{--- } ⑤$$

Put value of A from eq ⑤ to A

$$B = (a + ba^*b)^*ba^*$$

$$RE = (a + ba^*b)^*ba^*$$

EXAMPLE 4



$$A = \epsilon + A \cdot 0 + C \cdot 1 \quad \text{--- } ①$$

$$B = A \cdot 1 + B \cdot 0 \quad \text{--- } ②$$

$$C = B \cdot 1 + C \cdot 0 \quad \text{--- } ③$$

$$R = \emptyset + R_P$$

$\emptyset P^*$

$$B = A \cdot 1 \cdot 0^* \quad \text{--- } ④$$

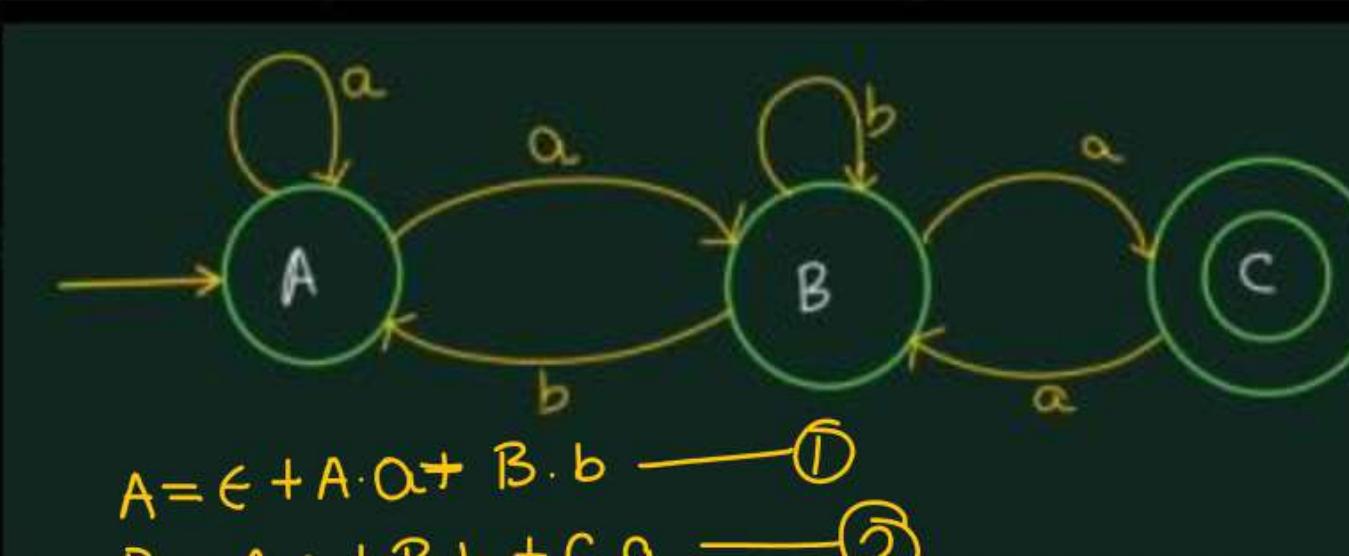
$$C = B \cdot 1 \cdot 0^* \quad \text{--- } ⑤$$

$$\begin{aligned}
 A &= \epsilon + A \cdot 0 + C \cdot 1 \quad \text{--- } ① \\
 &\quad \epsilon + A \cdot 0 + B \cdot 1 \cdot 0^* \cdot 1 \\
 &\quad \epsilon + A \cdot 0 + A \cdot 1 \cdot 0^* \cdot 1 \cdot 0^* \cdot 1 \\
 A &= \epsilon + A(0 + 1 \cdot 0^* \cdot 1 \cdot 0^* \cdot 1) \\
 R &= \emptyset + R_P \\
 A &= \epsilon \cdot (0 + 1 \cdot 0^* \cdot 1 \cdot 0^* \cdot 1)^* \\
 A &= (0 + 1 \cdot 0^* \cdot 1 \cdot 0^* \cdot 1)^* \\
 R_E &= (0 + 1 \cdot 0^* \cdot 1 \cdot 0^* \cdot 1)^*
 \end{aligned}$$

$$\begin{aligned}
 C &= B \cdot 1 \cdot 0^* \\
 B &= A \cdot 1 \cdot 0^*
 \end{aligned}$$

$$\underline{A \cdot 1 \cdot 0^* \cdot 1 \cdot 0^*}$$

EXAMPLE 5



$$A = \epsilon + A \cdot a + B \cdot b \quad \text{---} \textcircled{1}$$

$$B = A \cdot a + B \cdot b + C \cdot a \quad \text{---} \textcircled{2}$$

$$C = B \cdot a \quad \text{---} \textcircled{3}$$

Put eq₃ in eq₂

$$B = A \cdot a + B \cdot b + C \cdot a$$

$$B = A \cdot a + B \cdot b + B \cdot aa$$

$$B = A \cdot a + B(b+aa)$$

$$R = Q + RP$$

$$B = Aa(b+aa)^* \quad \text{---} \textcircled{4}$$

Put eq₄ in eq₁

$$A = \epsilon + A \cdot a + B \cdot b$$

$$A = \epsilon + A \cdot a + Aa(b+aa)^*b$$

$$A = \epsilon + A(a + a(b+aa)^*b)$$

$$A = \epsilon \cdot (a + a(b+aa)^*b)^*$$

$$A = (a + a(b+aa)^*b)^* \quad \text{---} \textcircled{5}$$

Put eq₆ in eq₄

$$B = (a + a(b+aa)^*b)^*a(b+aa)^* \quad \text{---} \textcircled{6}$$

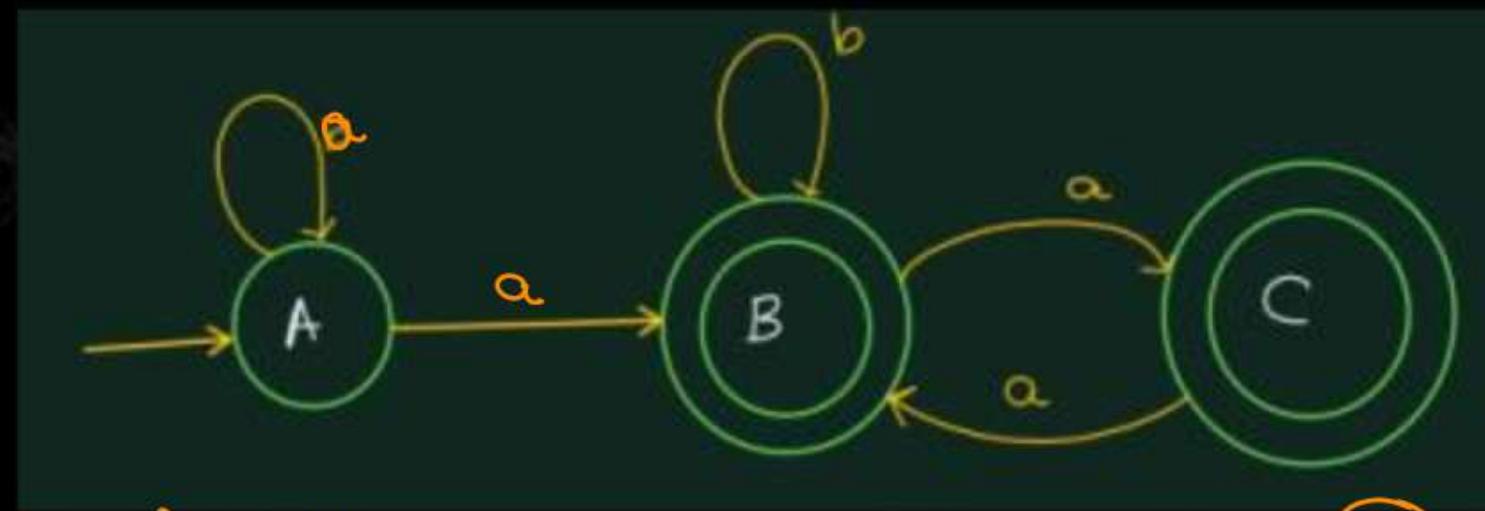
Put eq₇ in eq₃

$$C = B \cdot a$$

$$(a + a(b+aa)^*b)^*a(b+aa)^* \cdot a.$$

$$\downarrow R \cdot E$$

EXAMPLE 6



$$R = Q + RP$$

$$A = \epsilon + A \cdot a \quad \text{---} \textcircled{1}$$

$$B = A \cdot a + B \cdot b + C \cdot a \quad \text{---} \textcircled{2}$$

$$C = B \cdot a \quad \text{---} \textcircled{3}$$

Put eq3 in eq2

$$B = A \cdot a + B \cdot b + B \cdot a a \rightarrow B = A \cdot a + B(b+aa)$$

$$R = Q + RP$$

$$B = A \cdot a (b+aa)^* \quad \text{---} \textcircled{6}$$

Put eq5 in eq6

$$B = a^* a (b+aa)^* \quad \text{---} \textcircled{7}$$

Put equation 7 in eq3

$$C = B \cdot a$$

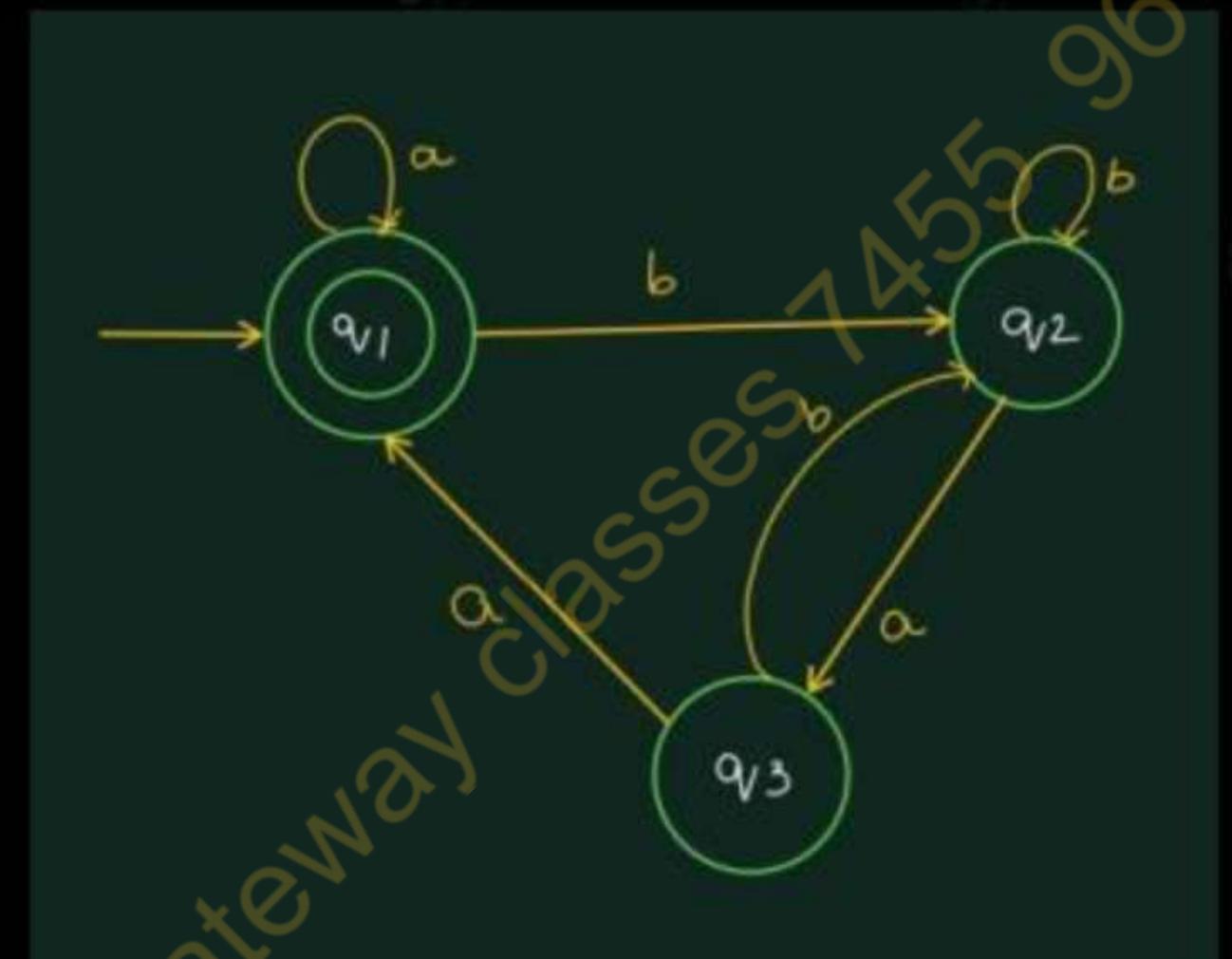
$$\boxed{C = a^* a (b+aa)^* a} \quad \text{---} \textcircled{8}$$

$$R.E = B + C$$

$$a^* a (b+aa)^* + a^* a (b+aa)^* a$$

$$= a^* a (b+aa)^* (\epsilon + a)$$

Find the regular expression for the following



Regular Expressions and Languages

TODAY's TARGET

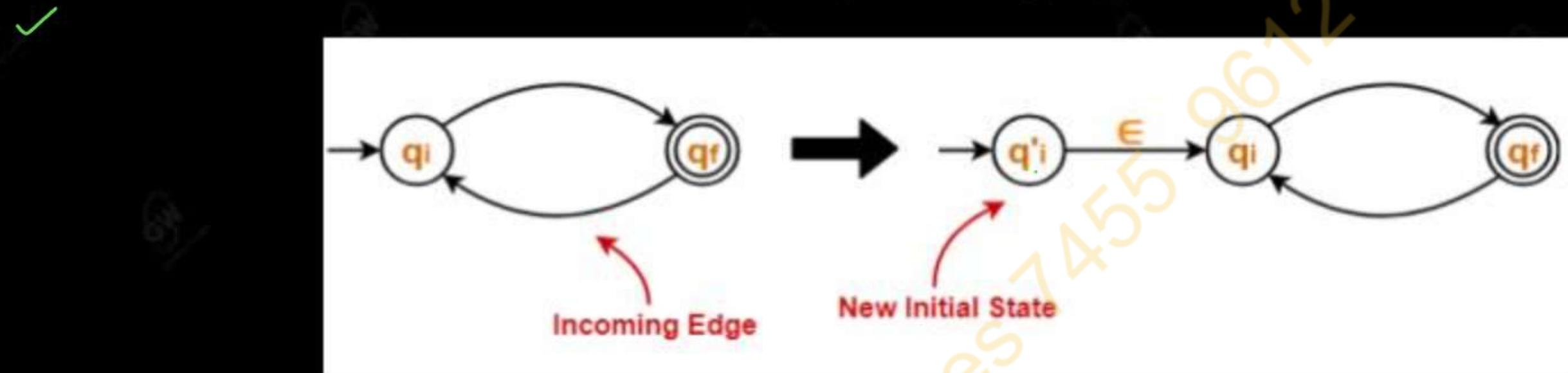
- State elimination method
- Converting finite automata to regular expression

Using state elimination method

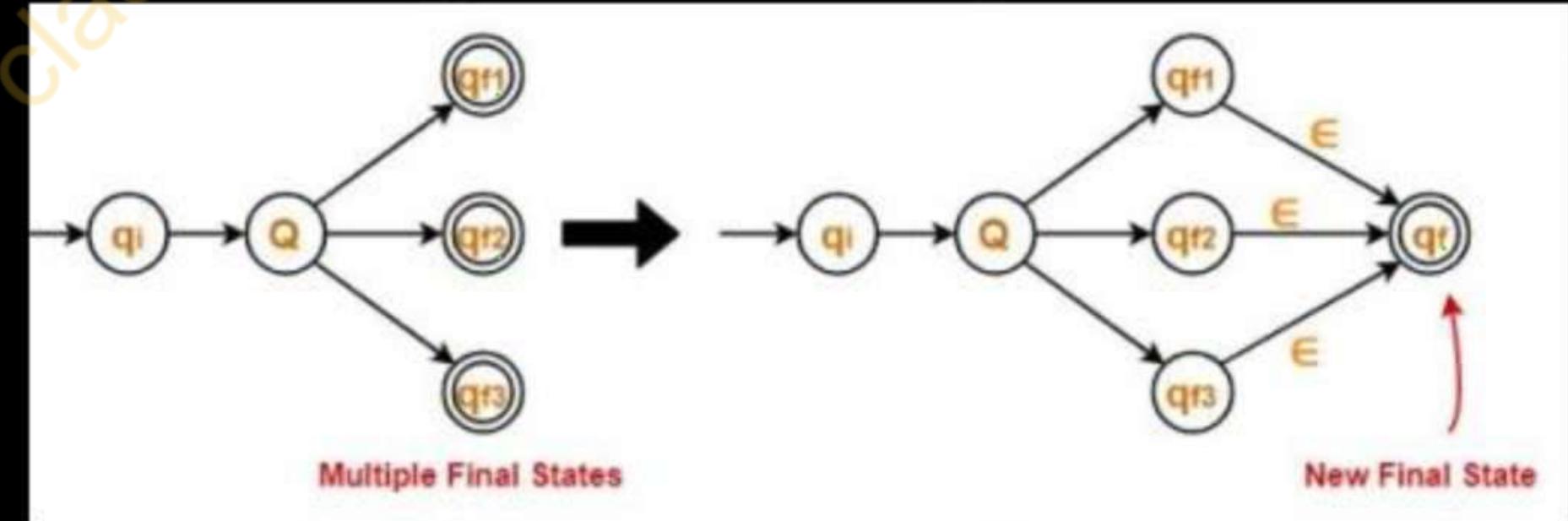
By PRAGYA RAJVANSHI
B.Tech, M.Tech(C.S.E)

State elimination method

1. The initial state of the DFA must not have any incoming edge.

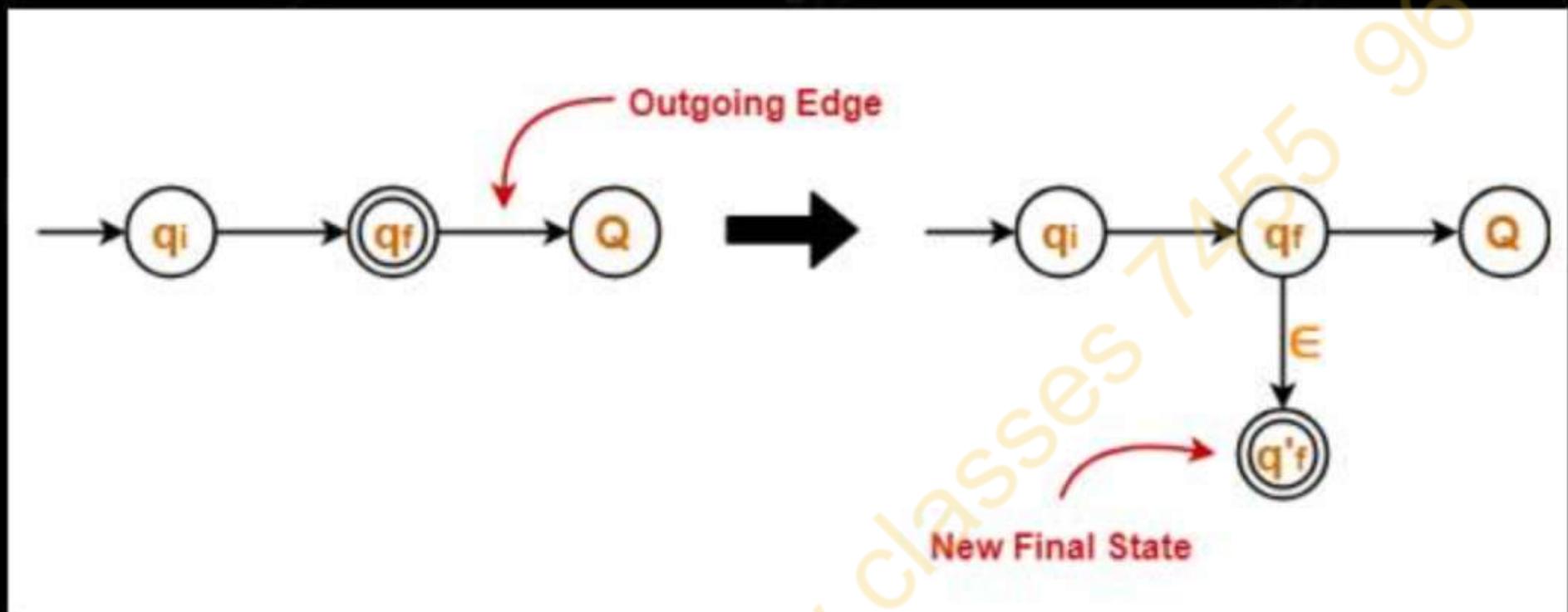


2. There must exist only one final state in the DFA.

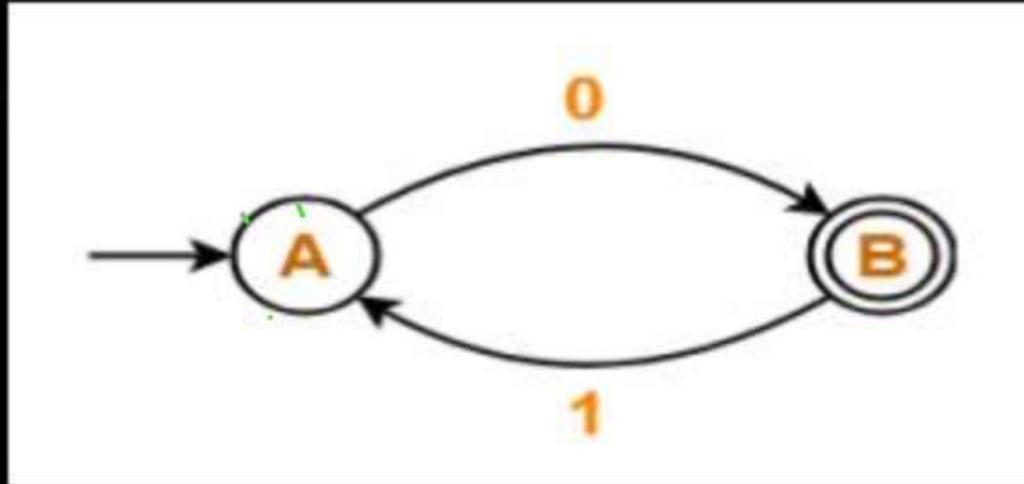


State elimination method

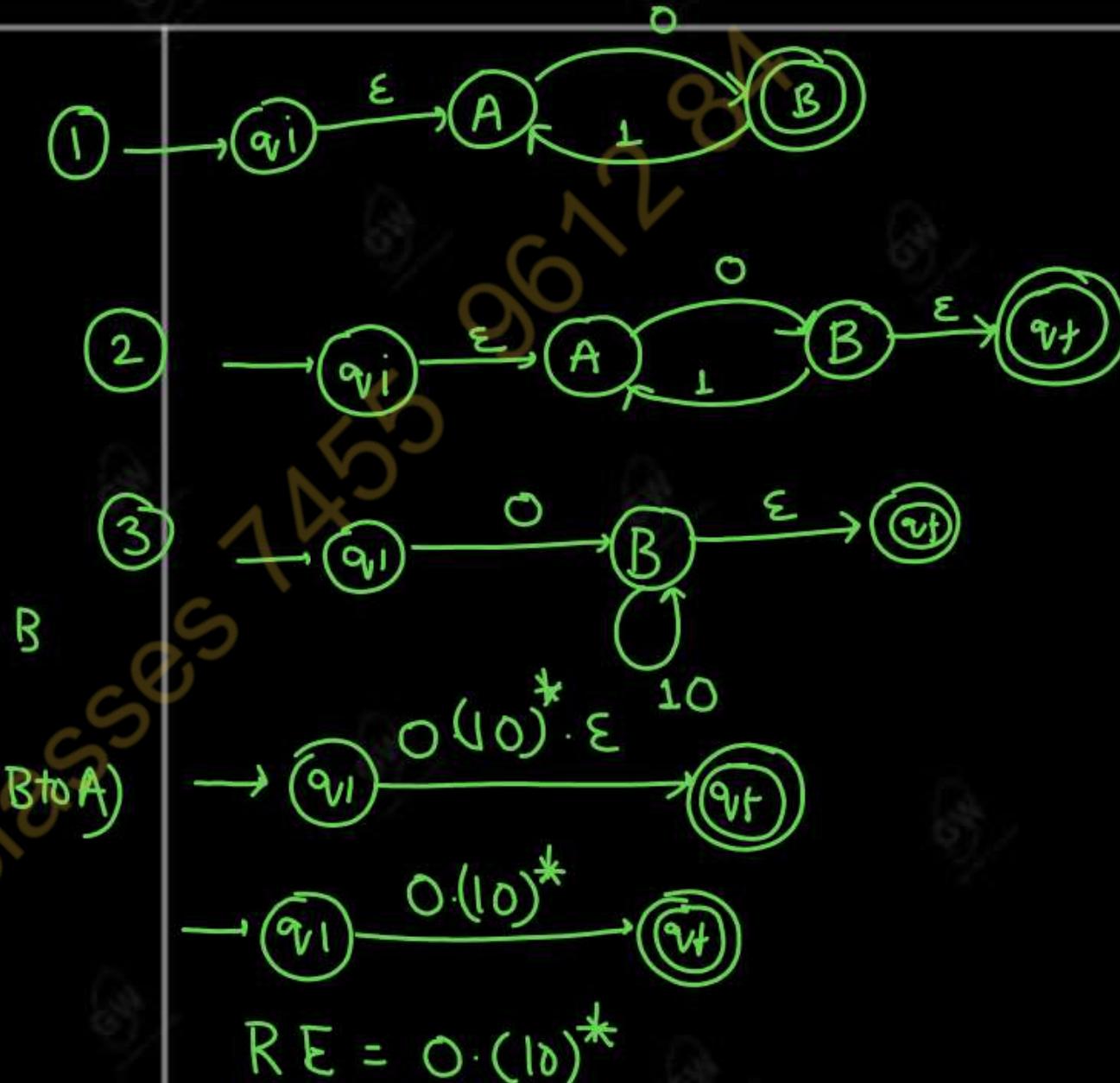
3. The final state of the DFA must not have any outgoing edge



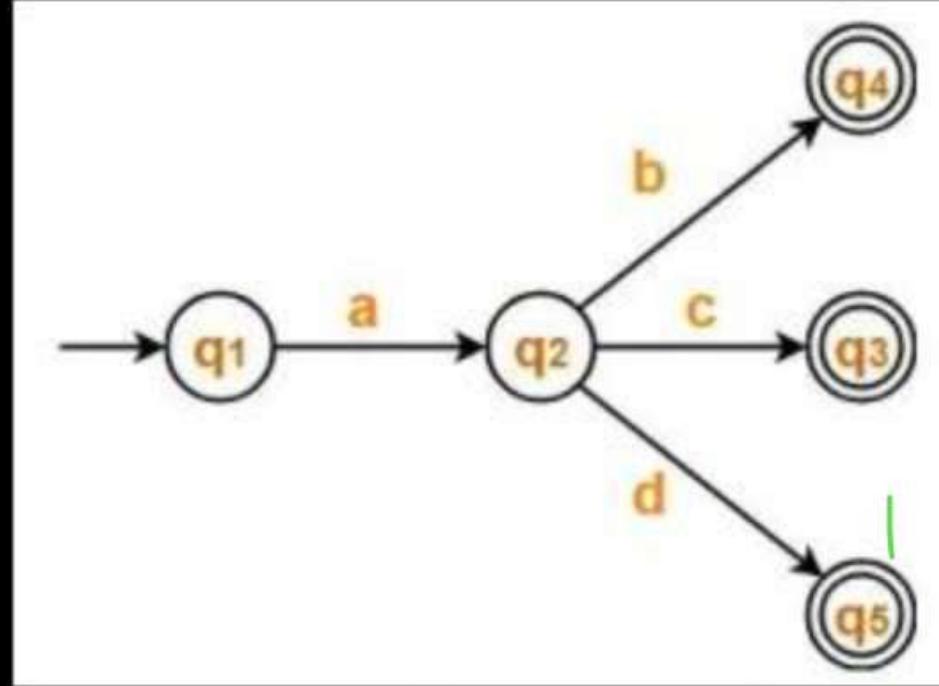
EXAMPLE 1



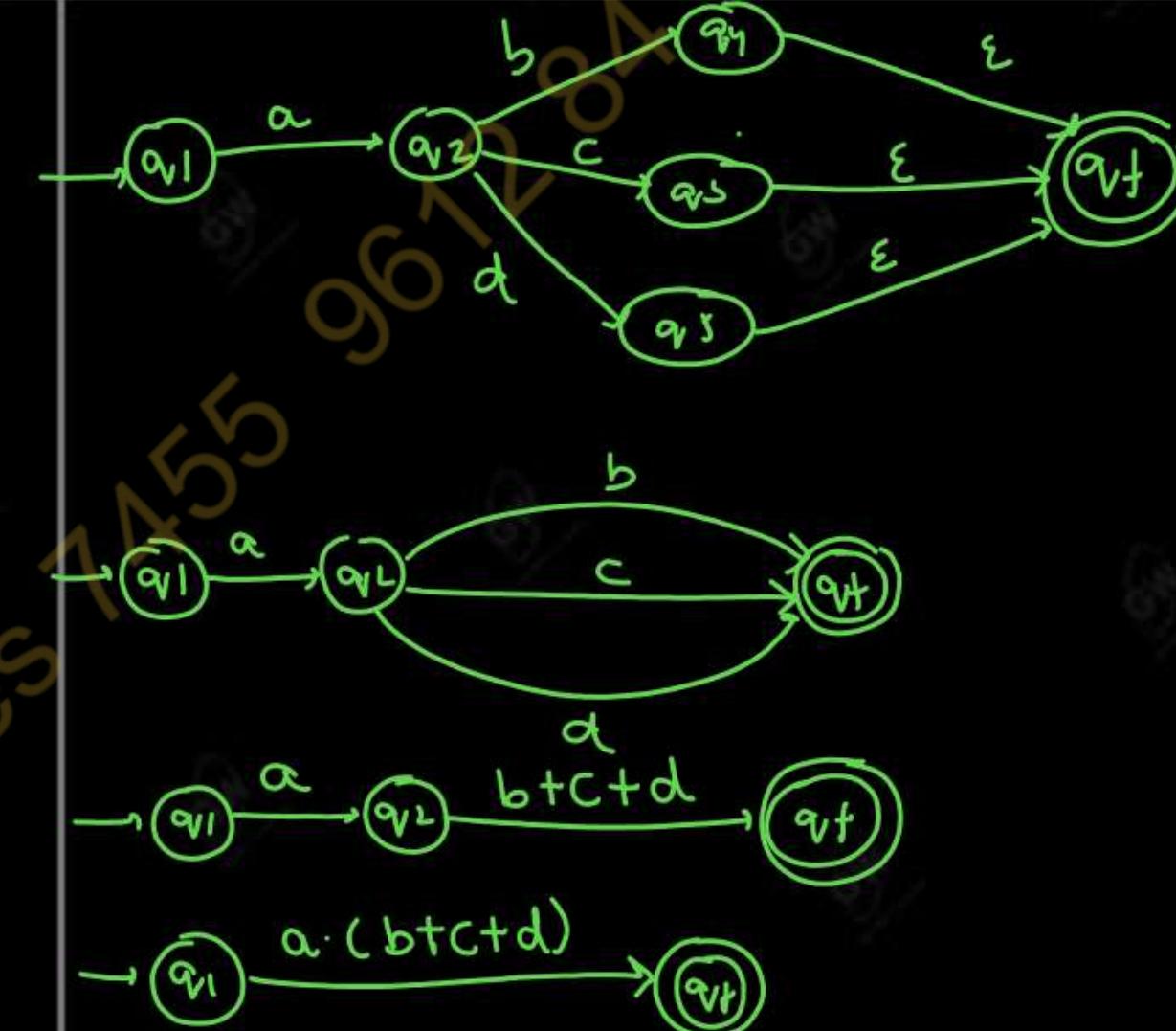
- ① there is an incoming (initial state) (A to B)
- ② Single final state ✓
- ③ there is an outgoing edge from final state (B to A)



EXAMPLE 2

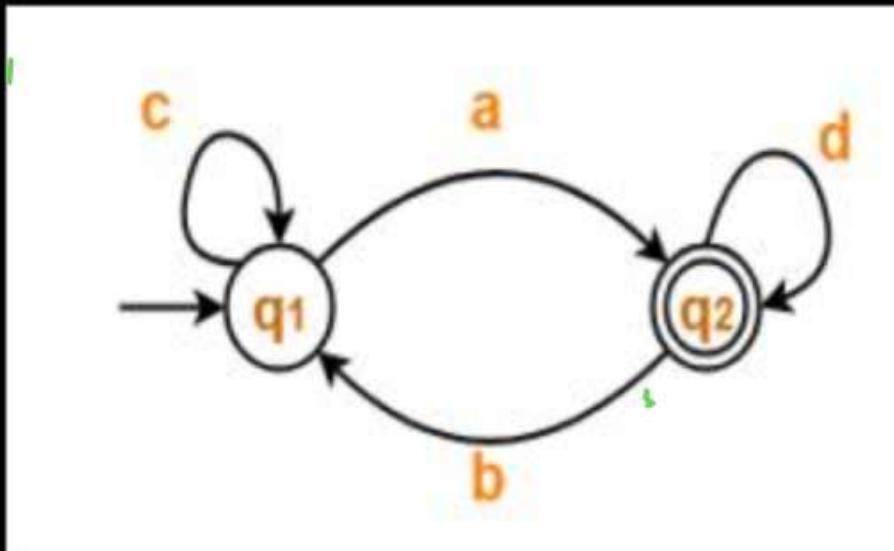


- NO incoming edge on initial state ✓
- Multiple final state
- NO outgoing edge from final state

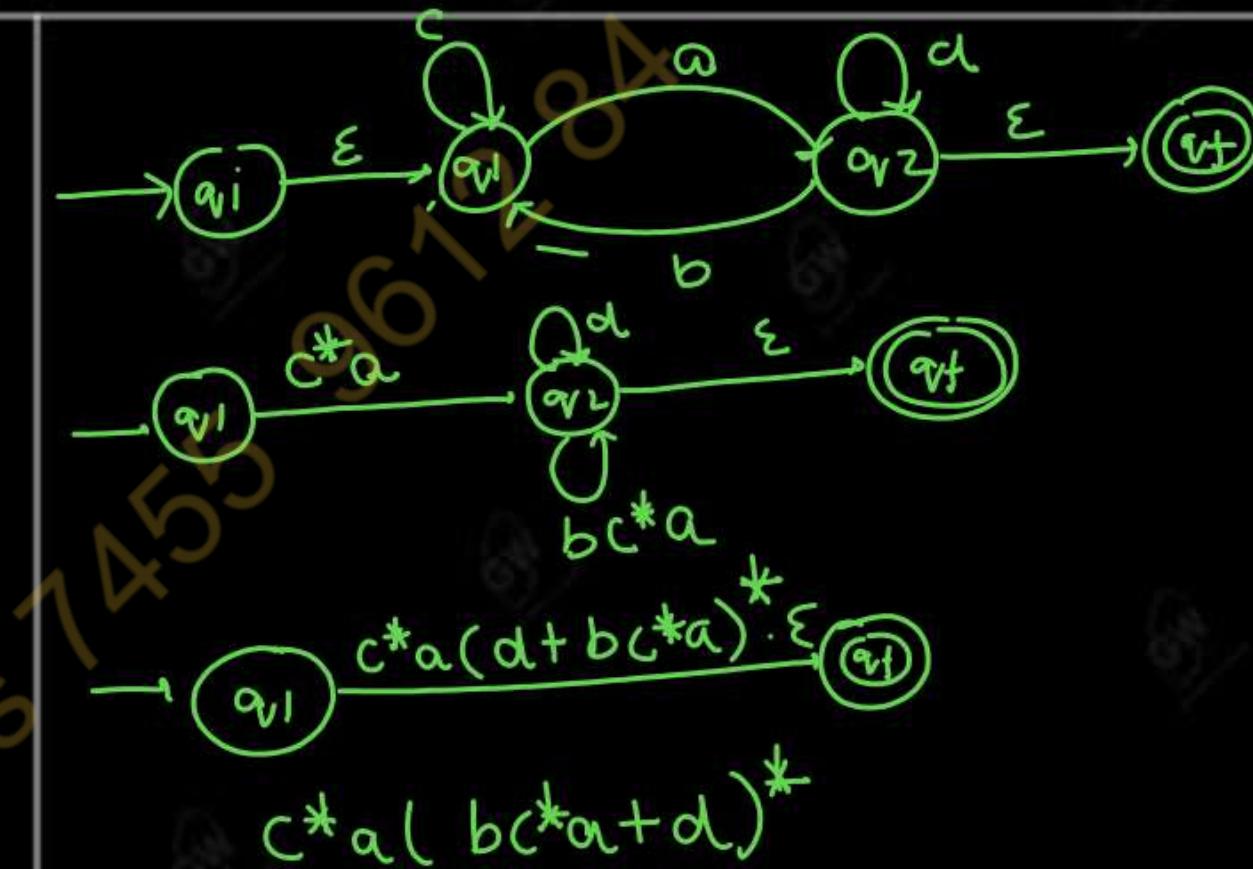


$$R.E. : a \cdot (b + c + d)$$

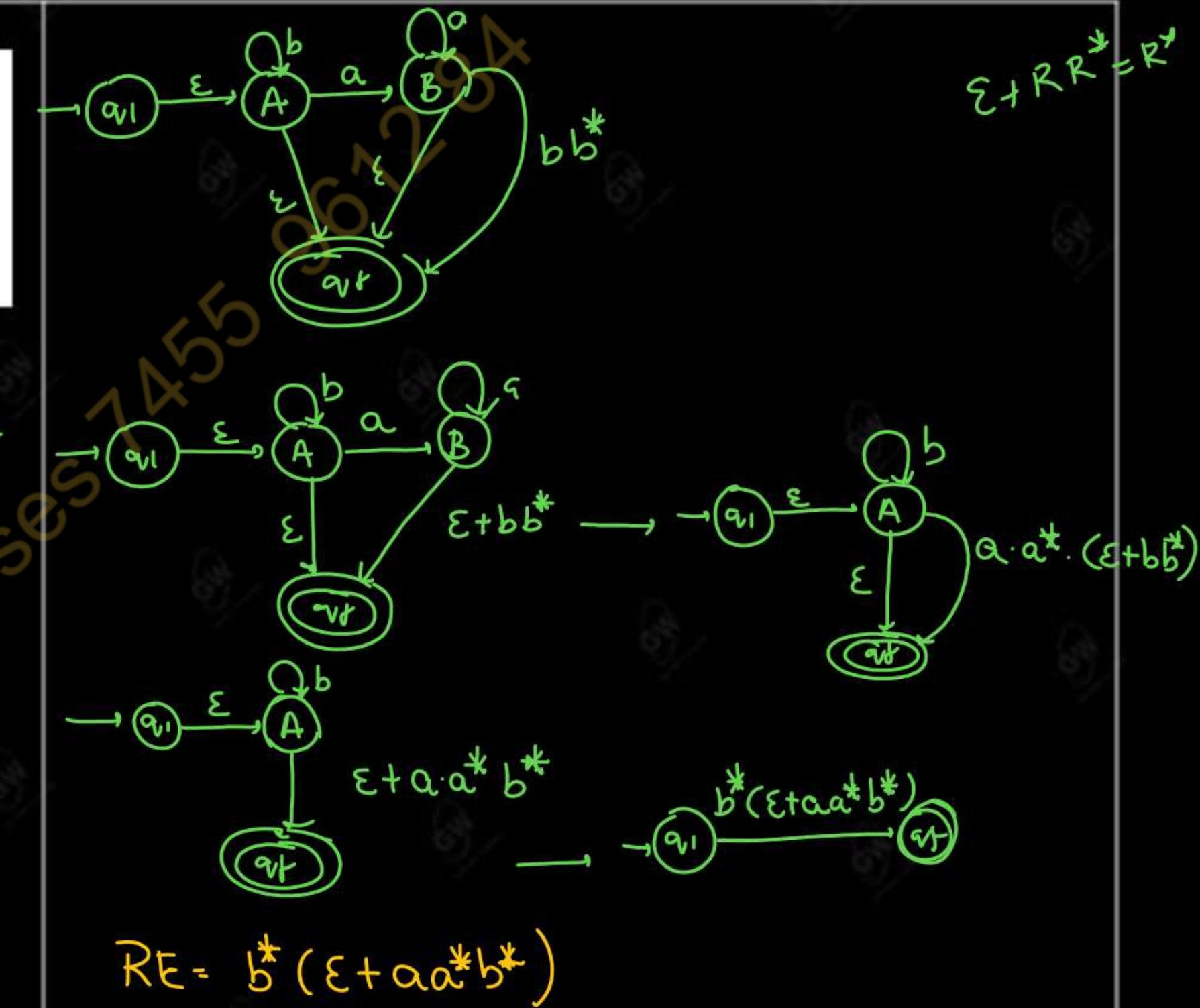
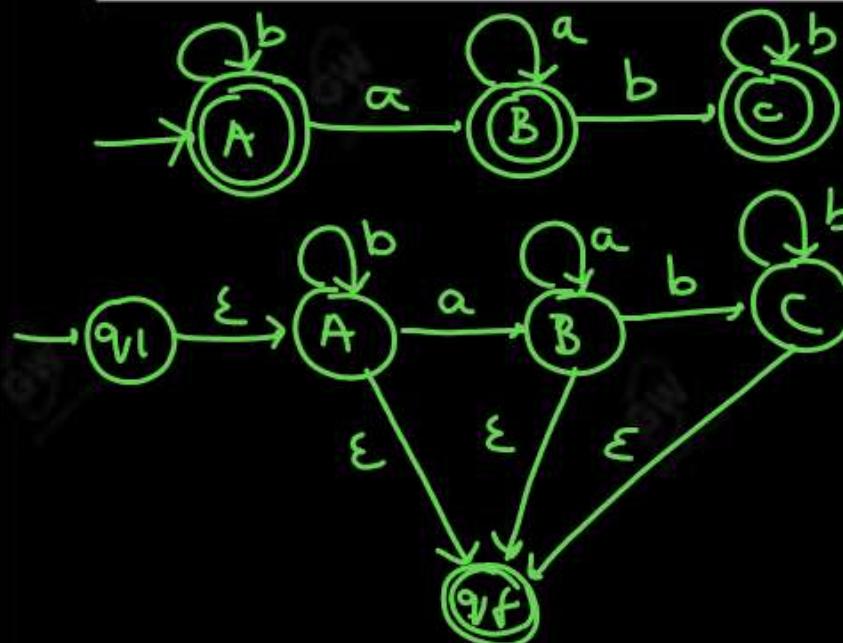
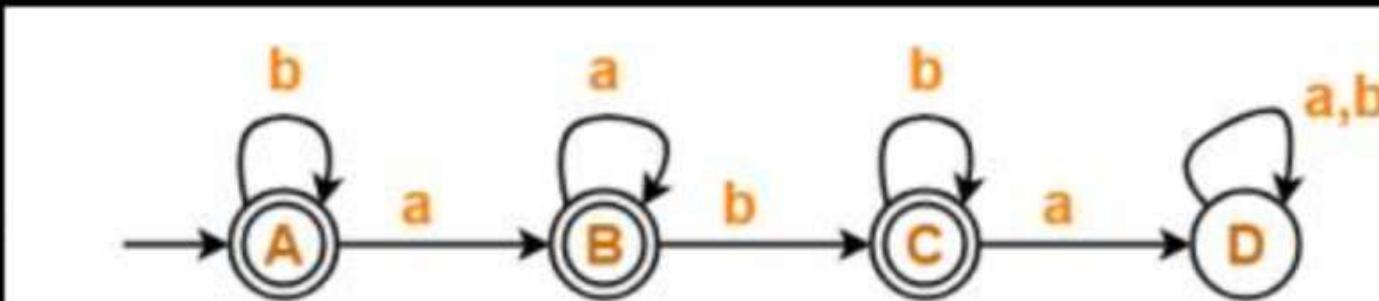
EXAMPLE 3



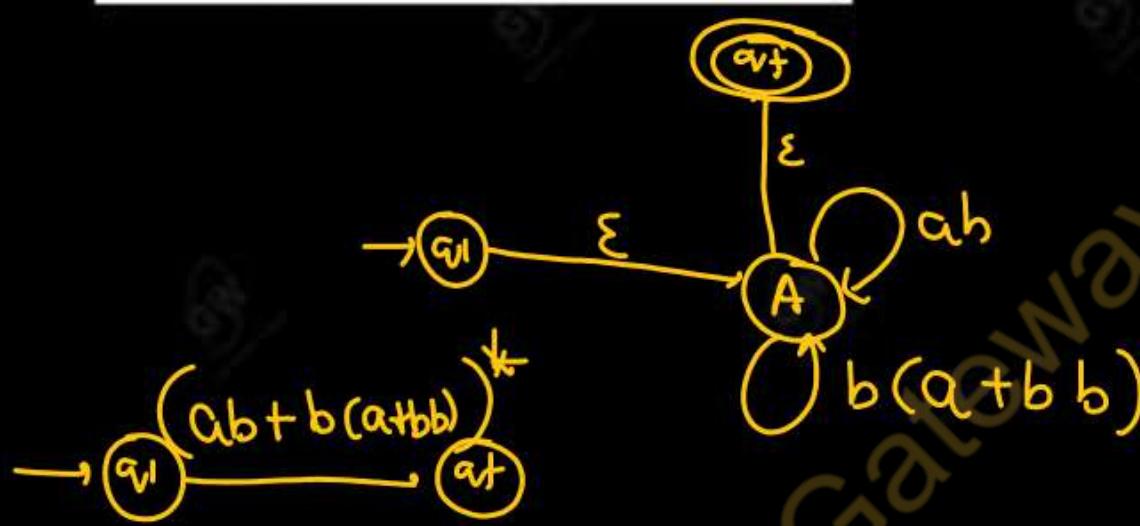
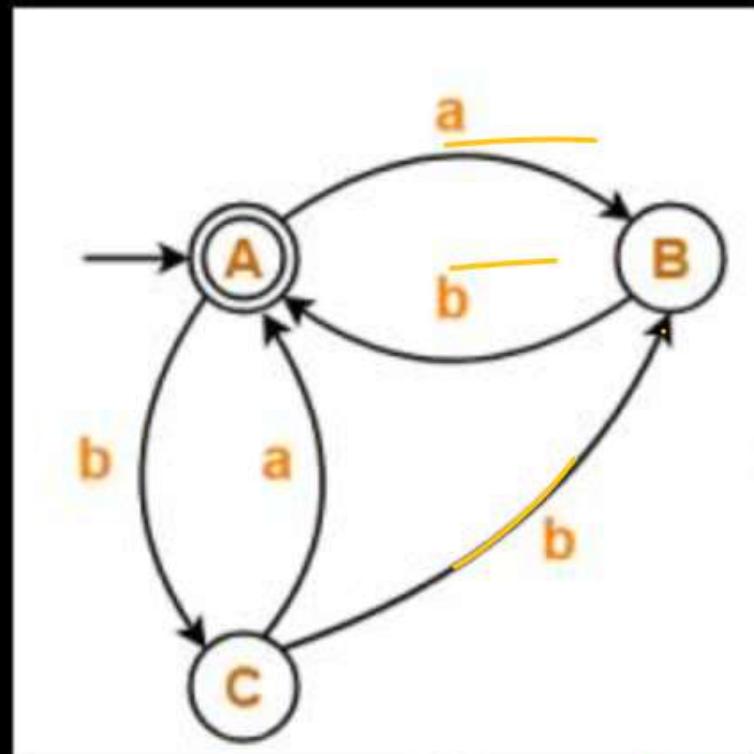
- Incoming in start state
- Single final state
- Outing edge from final state



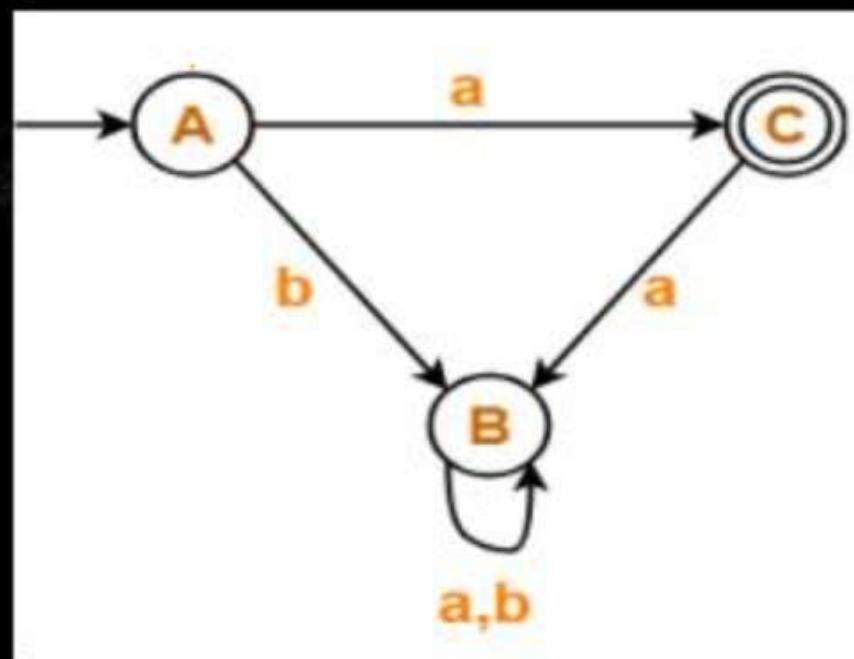
EXAMPLE 4



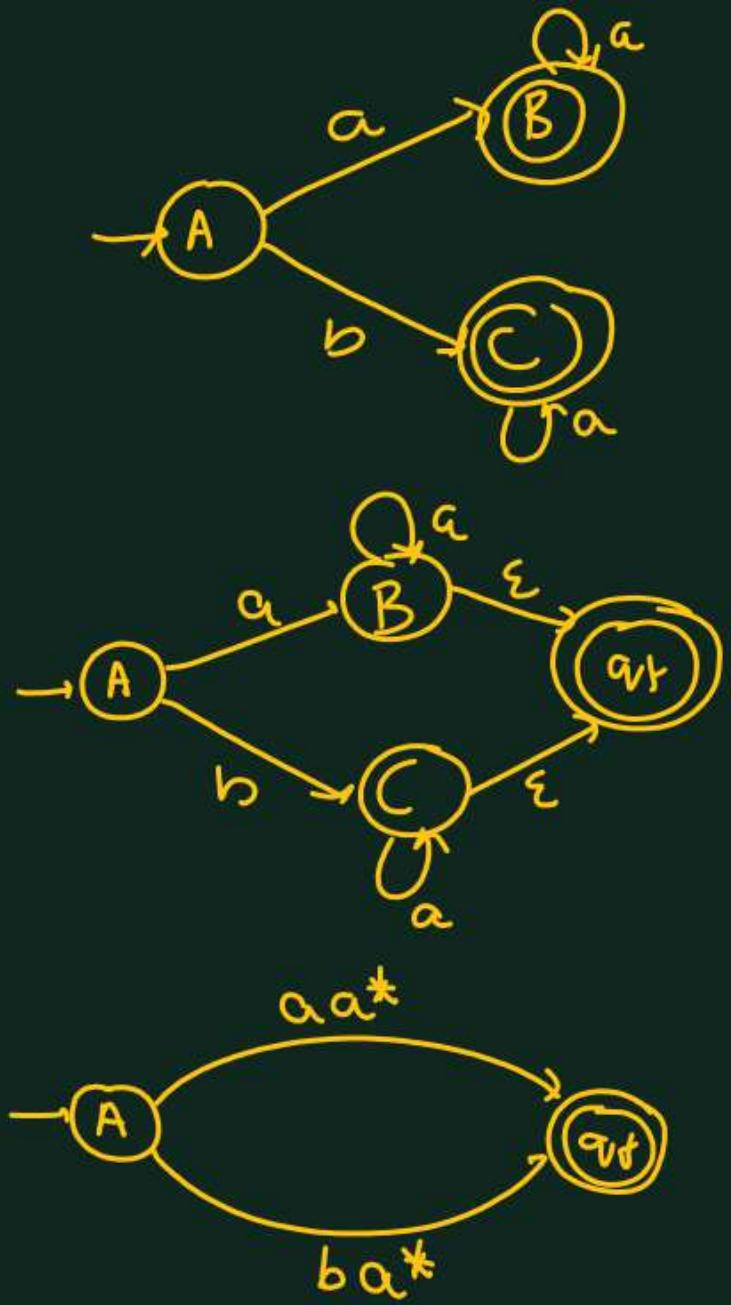
EXAMPLE 5



EXAMPLE 6



$$R.E = \alpha$$



$$(aa^* + ba^*)$$



Regular Expressions and Languages

TODAY's TARGET

- pumping lemma

By PRAGYA RAJVANSI

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

The language accepted by **Finite AUTOMATA** is known as **Regular Language**. Pumping Lemma is used to prove that a Language is not Regular. It cannot be used to prove that a language is Regular.

What is Pumping Lemma?

The term Pumping Lemma is made up of two words:-

Pumping: The word pumping refers to generating many input strings by pushing a symbol in an input string repeatedly.

Lemma: The word Lemma refers to the intermediate theorem in a proof.

There are two Pumping Lemmas, that are defined for

- Regular Languages
- Context-Free Languages

Pumping Lemma For Regular Languages

Theorem: If A is a Regular Language, then A has a Pumping Length 'P' such that any string 'S' where $|S| \geq P$ may be divided into three parts $S = xyz$ such that the following conditions must be true:

1.) $xy^iz \in A$ for every $i \geq 0$

2.) $|y| > 0$

3.) $|xy| \leq P$

In simple words, if a string y is 'pumped' or insert any number of times, the resultant string still remains in A

$L = \{a^n b^n \mid n \geq 1\}$ is not a regular language

Let L is Regular language

$L = \{ab, aabb, aaabbb, \dots\}$

$P=4$

$|S| > P$

$S = aabb$

$|S| = 4$

$|S| > P \quad |y| \geq 4 - \text{True}$

$S = XYZ$

$\begin{matrix} aabb \\ \downarrow \\ x \quad y \quad z \end{matrix}$

$y = a \quad |y| = 1 \quad |y| > 0$
 $xy = aa \quad |xy| = 2 \quad |xy| > 0$

$xy^iz \in L \quad i \geq 0 \quad 2 \leq 4 = T$

$a(a)^i(b b)$

$i=0 \quad a \cdot \epsilon \cdot bb = abb \notin L$

$i=1 \quad a \cdot abb \in L$

$i=2 \quad a \cdot a \cdot bb \notin L$

$S: ab$

$|S| = 2$

$|S| > P$

$2 > 4 - \text{False}$

}

$S = aabb$

if a string y is pumped number of time string will not remain in language

this not Regular Language

Pumping Lemma For Regular Languages

$L = \{a^n b^{2n} \mid n \geq 1\}$ is not a regular language

let L is a Regular Language

$L = \{abb, aabbbb, aaaabb bbb bbb\}$

$P=6$ (let)

$$S = abb$$

$$|S|=3$$

$$|S| > P$$

$$3 > 6 - \text{False}$$

$$S = aabb bbb$$

$$|S|=6$$

$$|S| \geq 1$$

$$6 \geq 6 \top$$

$$S = \underbrace{aab}_{x} \underbrace{bb}_{y} \underbrace{b}_{z}$$

$$\begin{aligned} \textcircled{1} |y| &> 0 & y &= bb \\ &|y|=2 & 2 &> 0 \top \end{aligned}$$

$$\begin{aligned} \textcircled{2} xy &= aabb b \\ |xy| &= |aabbb| = 5 \end{aligned}$$

$$5 \leq P \top$$

$$\begin{aligned} S &= aab (bb)^i b \\ i=0 & aab \in L = abb \in L \\ i=1 & aabb \in L \\ i=2 & aabbabb \notin L \end{aligned}$$

this is not a Regular language

case II

$$S = \underbrace{aa}_{x} \underbrace{bb}_{y} \underbrace{bb}_{z}$$

$$\textcircled{1} |y| > 0 \quad 3 > 0 \top$$

$$\textcircled{2} |xy| \leq P \quad 4 \leq 6 \top$$

$$\textcircled{3} S = xy^i z \in L \quad i \geq 0$$

$$S = a (abb)^i bb$$

$$i=0 \quad a \in L = abb \in L$$

$$i=1 \quad aabb \in L$$

$$i=2 \quad aabbabb \notin L$$

$$\begin{aligned} x &= a & |y|=3 \\ y &= abb & |xy|=4 \\ z &= bb \end{aligned}$$

from these two cases
we proof this is not
a Regular language

Pumping Lemma For Regular Languages

$L = \{a^n \mid n \text{ is prime}\}$ is not a regular language

$n = 2, 3, 5, 7, \dots$

let L is a regular language

$L = \{aa, aaa, aaaaa, \dots\}$

$p=3$

$s = aaaaa$

$|s|=5$

$|s| \geq p \quad s \geq 3$

$s = \overbrace{aaaaa}^s \quad x = a \quad y = a \quad z = aaa$

$s = xyz \in L \quad i > 0$

$s = a(a)^i a a$

$i=0 \quad a \cdot \epsilon \cdot a a = a a a \notin L$

This is not Regular

Case 2

$s = \overbrace{aa}^x \overbrace{aaa}^y \overbrace{a}^z$

①

$$\begin{aligned}|y| &> 0 \\ |a| &> 0 \\ i &> 0\end{aligned}$$

$\overbrace{aa}^x \overbrace{aaa}^y \overbrace{a}^z$

$x = aa$

$y = a$

$z = aa$

②

$$|xy| \leq p$$

$$|aaa| \leq 3$$

$$3 \leq 3 \quad T$$

$s = xy^i z \in L \quad i > 0$

$s = a a (a)^i a a$

$i=0 \quad a a \cdot \epsilon \cdot a a \quad a a a a \notin L$

$i=1 \quad a a a a a \in L$

$i=2 \quad a a a a a a \notin L$

This is not
a Regular
One

Pumping Lemma For Regular Languages

$L = \{a^{n^2} \mid n \geq 0\}$ is not a regular language

$$L = \{a^0, a^1, a^2, a^3 - a^3 - a^4 - \dots\}$$

$$L = \{\epsilon, a, a^1 - a^1 - a^2 - \dots\}$$

Let L is a Regular Language

$$P = 3$$

$$S = aaaa$$

$$|S| \geq P$$

$$|aaaa| \geq 3$$

$$4 \geq 3$$

+ true

$$S = \overbrace{aaa}^x \underbrace{aa}_{y} \overbrace{a}^z$$

$$|Y| > 0 \quad |a| > 0 \quad L > 0 \quad T$$

$$|XY| \leq P \quad |aa| \leq P \quad 2 \leq 3 \quad T$$

$$S = XY^i Z \in L \quad i \geq 0$$

$$S = a(a)a$$

$$i=0 \quad a \cdot \epsilon \cdot a \notin L \quad aaa \notin L$$

this not a regular language

Language

$$S = \overbrace{aaa}^x \underbrace{aa}_{y} \overbrace{a}^z$$

①

$$|Y| > 0 \quad |a| > 0 \quad L > 0 \quad T$$

②

$$|XY| \leq P \quad |aa| \leq P \quad 3 \leq 3 \quad T$$

$$S = XY^i Z$$

$$aa(a)a$$

$$i=0 \quad aa \cdot a = aaa \notin L$$

$$i=1 \quad aaaa \in L$$

$$i=2 \quad aaaaa \notin L$$

from these two case we confirmed

that L is not a regular language

$L = \{0^{3x} 1^{2y} \mid x > y > 0\}$ is not a regular language

$$L = \{O^{3*2} 1^{2*1}, O^{3*3} 1^{2*2}, \dots, O^5 1^2, O^9 1^4, \dots\}$$

$$P = \emptyset$$

$$g = 000000 \mid$$

$$|S| \geq f$$

1000000 || | > €

$8 \geq 6$ True

$$S = \overbrace{0000000}^x \overbrace{000000}^y \overbrace{000000}^z$$

1

$$|Y| > 0$$

10 | Page

$$|xy| < p$$

$$|10000| \leq 6$$

$$S = xy^iz \in L \quad i > 0$$

$$S = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}$$

$i = 0$ 000. € 001

0000011

1-2 000000011€L

this not a Regular lang

this not a Regular lang

CAS

$$S = \{0, 0, 0, 0, 0\} \quad | \quad \}$$

$|\gamma| > 0$ || 00000 | 5 > 0 T

$$|xy| \leq p \quad |0000001| \leq 6$$

S - 00 (00001)

$i = 0 \quad 00 \in I = 001 \notin L$

This is not Regular

Pumping Lemma For Regular Languages

$L\{yy, y \text{ belongs to } \{0,1\}^*\}$ is not a regular language

$$L = \{00, 11, 0101, 1010, \dots\}$$

Let L is a Regular Language

$$P = 4$$

$$S = 0101$$

$$|S| > P$$

$$|0101| > P$$

$$4 > 4$$

$$S = \underset{x}{0} \underset{y}{10} \underset{z}{1}$$

$$|y| > 0 \quad |1| > 0 \quad |1| > 0 \quad T$$

$$|xy| \leq P \quad |01| \leq 4 \quad 2 \leq 4 \quad T$$

$$S = xy^i z \in L \quad i > 0$$

$$0(1)^i 01$$

$$i=0 \quad 0 \in 01 = 00 \notin L$$

$$i=1 \quad 0101 \in L$$

$$i=2 \quad 01101 \notin L$$

$$S = \underset{x}{0} \underset{y}{10} \underset{z}{1}$$

$|y| > 6$
 $|10| > 0$
 $2 > 0$
 T

$S = 0 \underset{i=0}{(10)^i} 1$
 $i=0 \quad 0 \in 1 = 01 \notin L$
 $|xy| \leq P$
 $|010| \leq 4$
 $3 \leq 4$
 T

$|y| > 0$
 $|0| > 0$
 $1 > 0$ true
 $|xy| \leq P$
 $|010| \leq 4$
 $3 \leq 4$ T

this is not
Regular

$$\underset{x}{0} \underset{y}{10} \underset{z}{1}$$

$$|y| > 0 \quad 1 > 0$$

$$|xy| \leq P$$

$$|010| \leq 4$$

$$3 \leq 4$$

$$S = 01(0)^i 1$$

$$i=0 \quad 01 \in 1 = 011 \notin L$$

Regular Expressions and Languages

TODAY's TARGET

- Recursive definition for a regular expression
- Convert language to regular expression
- Pumping lemma question

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

Recursive definition for a regular expression

A Regular Expression can be recursively defined as follows -

- ϵ is a Regular Expression indicates the language containing an empty string. ($L(\epsilon) = \{\epsilon\}$)
- ϕ is a Regular Expression denoting an empty language. ($L(\phi) = \{\}$)
- x is a Regular Expression where $L = \{x\}$
- If X is a Regular Expression denoting the language $L(X)$ and Y is a Regular Expression denoting the language $L(Y)$, then
 - $X + Y$ is a Regular Expression corresponding to the language $L(X) \cup L(Y)$ where $L(X+Y) = L(X) \cup L(Y)$.
 - $X \cdot Y$ is a Regular Expression corresponding to the language $L(X) \cdot L(Y)$ where $L(X \cdot Y) = L(X) \cdot L(Y)$
 - R^* is a Regular Expression corresponding to the language $L(R^*)$ where $L(R^*) = (L(R))^*$

If we apply any of the rules several times from 1 to 4, they are Regular Expressions.

$$R = ab$$

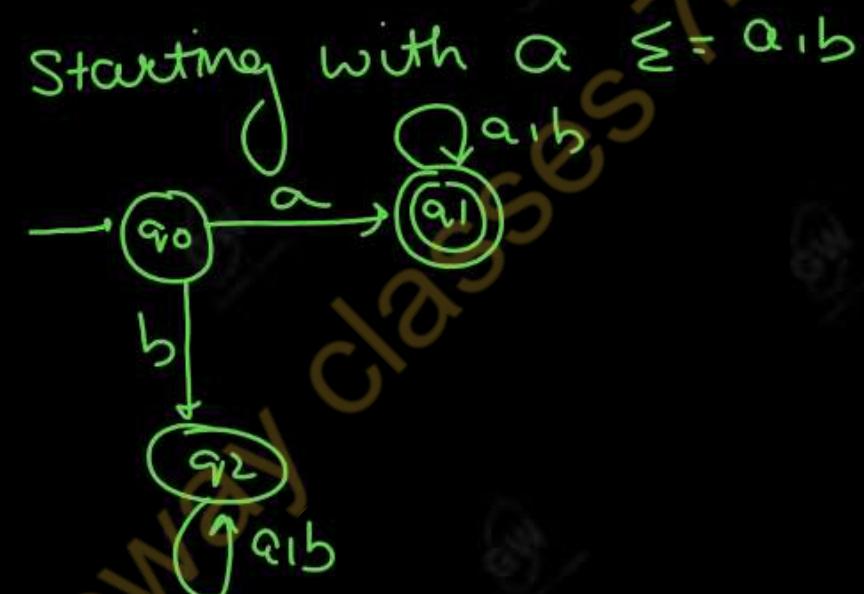
$$R^* = (ab)^*$$

Example of regular expression

$(0 + 10^*)$	$L = \{ 0, 1, 10, 100, 1000, 10000, \dots \}$
(0^*10^*)	$L = \{ 1, 01, 10, 010, 0010, \dots \}$
$(0 + \varepsilon)(1 + \varepsilon)$	$L = \{ \varepsilon, 0, 1, 01 \}$
$(a+b)^*$	Set of strings of a's and b's of any length including the null string. So $L = \{ \varepsilon, a, b, aa, ab, bb, ba, aaa, \dots \}$
$(a+b)^*abb$	Set of strings of a's and b's ending with the string abb. So $L = \{ abb, aabb, babb, aaabb, ababb, \dots \}$
$(11)^*$	Set consisting of even number of 1's including empty string, So $L = \{ \varepsilon, 11, 1111, 111111, \dots \}$
$(aa)^*(bb)^*b$	Set of strings consisting of even number of a's followed by odd number of b's, so $L = \{ b, aab, aabbb, aabbabb, aaaab, aaaabbb, \dots \}$
$(aa + ab + ba + bb)^*$	String of a's and b's of even length can be obtained by concatenating any combination of the strings aa, ab, ba and bb including null, so $L = \{ aa, ab, ba, bb, aaab, aaba, \dots \}$

transition diagram/ transition graph

- **Transition diagram** is a special kind of flowchart for language analysis. In transition diagram the boxes of flowchart are drawn as circle and called as states. States are connected by arrows called as edges.
The label or weight on edge indicates the input character that can appear after that state.
- It is a graph which consists of series of state and there is successful path begin at start state end at the final state



1. Write the regular expression for the language starting and ending with a and having any combination of b's in between.

$$a b^* a$$

2. Write the regular expression for the language starting with a but not having consecutive b's.

$$(a+ab)^*$$

3. Write the regular expression for the language that does not end with 01

$$\epsilon + 0 + 1 + (0+1)^* (00+10+11)$$

4. Write the regular expression for the language having a string which should have at least one 0 and at least one 1

$$(0+1)^* 1 (0+1)^* 0 (0+1)^* + (0+1)^* 0 (0+1)^* 1 (0+1)^*$$

5. Write the regular expression for the language L over $\Sigma = \{0, 1\}$ such that all the string do not contain the substring 01

$$1^* 0^*$$

6. Write the regular expression for the language containing the string in which every 0 is immediately followed by 11.

$$(1 + 011)^* = ((1^* + (011)^*))^*$$

$$(1^* . (011)^*)^*$$

Pumping lemma

APPLICATION OF PUMPING LEMMA

- to prove that a particular language is non-regular

Let L is Regular Language

$$L = \{a^1, a^8, a^{27}, a^{64} \dots\}$$

$$\text{Let } p = 7$$

$$S = a \quad | \quad |S| = 1$$

$$|S| \geq p \quad |S| > p$$

$$1 > 7 \quad 8 > 7$$

(False)

case 3

$$S = \underbrace{aaa}_{x} \underbrace{aaa}_{y} \underbrace{aaa}_{z}$$

$$1. |y| > 0 \quad |y| = 3$$

$$2. |xy| \leq p \quad |aaa| \leq 7$$

$$3. |xy|^i z \in L \quad i > 0$$

$$aa(aaa)^i(aaa)$$

$$i=0 \quad aa \epsilon . aa = a^5 \notin L$$

$$i=1 \quad aaaa \epsilon . aaaa = a^8 \notin L$$

$$i=2 \quad aaaaaaa \epsilon . aaaa = a^9 \notin L$$

this is not Regular lang.

(7 marks)

Prove the language $L = \{a^n \mid n \text{ is a perfect cube}\}$ is not a regular language?

Case 2

$$S = \underbrace{aaa}_{x} \underbrace{aaa}_{y} \underbrace{aaa}_{z}$$

$$1. |y| > 0 \quad |y| = 3$$

$$2. |xy| \leq p \quad |aaa| \leq 7$$

$$3. |xy|^i z \in L \quad i > 0$$

$$aaa(aa)^i(aa)$$

$$i=0 \quad aaa \epsilon . aa = a^5 \notin L$$

$$i=1 \quad aaaa \epsilon . aaaa = a^8 \notin L$$

$$i=2 \quad aaaaaaa \epsilon . aaaa = a^9 \notin L$$

this is not Regular Lang.

$$S = \underbrace{aaa}_{x} \underbrace{aaa}_{y} \underbrace{aaa}_{z}$$

$$1. |y| > 0 \quad |y| = 3$$

$$2. |xy| \leq p \quad |aaa| \leq 7$$

$$3. |xy|^i z \in L \quad i > 0$$

$$3. S = xy^i z \in L \quad i > 0$$

$$aaa(aa)^i(aa)$$

$$i=0 \quad aaa \epsilon . aa = a^5 \notin L$$

$$i=1 \quad aaaa \epsilon . aaaa = a^8 \notin L$$

$$i=2 \quad aaaaaaa \epsilon . aaaa = a^9 \notin L$$

Pumping lemma

Prove the language $L = \{a^n \mid n \text{ is a perfect square}\}$ is not a regular language

Let L is Regular language

$$L = \{a^1, a^4, a^9, a^{16}, \dots\} \\ P = 11$$

$$\begin{array}{c|c|c} S = a & S = a^4 & S = a^9 \\ |S| = 1 & |S| = 4 & |S| = 9 \\ |S| > P & 4 > 11 & 9 > 11 \\ i > 11 & F & T \end{array}$$

$$S = \underbrace{aaaa}_{x} \underbrace{aaaa}_{y} \underbrace{a}_{z}$$

① $|y| > 0$ $|xy| \leq P$
 $|aaaa| > 0$ $|aaaaaaa| \leq 11$
 $4 > 0$ $8 \leq 11$ T
T

$$S = xy^i z \in L \quad i > 0$$

aaaa (aaaa)ⁱ a

$$i=0 \quad a^4 \cdot \epsilon \cdot a = a^5 \notin L$$

this is not Regular.

$$\begin{array}{c} \cancel{a^0}^{e^2} \\ S = \underbrace{aaa}_{x} \underbrace{aaaa}_{y} \underbrace{a}_{z} \end{array}$$

$|y| > 0 \rightarrow |aaaaa| > 0$
 $5 > 0 \rightarrow T$
 $|xy| \leq P$
 $|aaaaaaaa| \leq 11 \quad 8 \leq 11 \quad T$

$$S = xy^i z \in L \quad i > 0$$

aaaa (aaaa)ⁱ a

$i=0 \quad aaaa \quad a^4 \in L$

$i=1 \quad aaaaaaaaaa \quad a^9 \in L$

$i=2 \quad a^3 a^{10} a^1 = a^{14} \notin L$

this is not Regular.

$$\begin{array}{c} \cancel{a^0}^e \\ S = \underbrace{aa}_{x} \underbrace{aa}_{y} \underbrace{aaaaa}_{z} \end{array}$$

$|y| > 0$
 $|aa| > 0 \quad 2 > 0 \quad T$

$|xy| \leq P$
 $|aaaa| \leq 11 \quad 4 \leq 11 \quad T$

$$S = xy^i z \in L \quad i > 0$$

aa (aa)ⁱ aaaaa

$i=0 \quad a^2 a^5 = a^7 \notin L$

this is not Regular.

Regular expression and language

TODAY s' TARGET

- Kleene theorem and proof
- Equivalence of two finite automata
- Pigeon hole principle
- Proof of pumping lemma theorem

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

➤ If a language can be expressed by finite automata , transition graph or regular expression, then it can also be expressed by other two as well

➤ It may be noted that theorem is proved by proving the three following parts-

➤ **KLEENE THEOREM PART 1**

If a language is accepted by finite automata then it must be accepted by transition graph

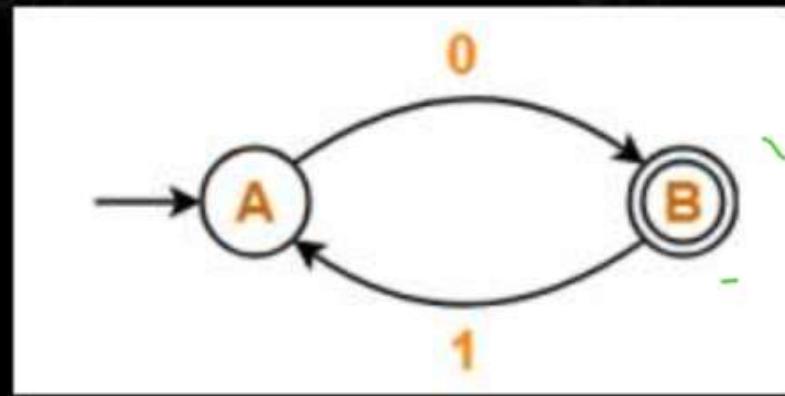
PROOF: every finite automata is considered to be the transition graph then nothing to prove

➤ **KLEENE THEOREM PART 2**

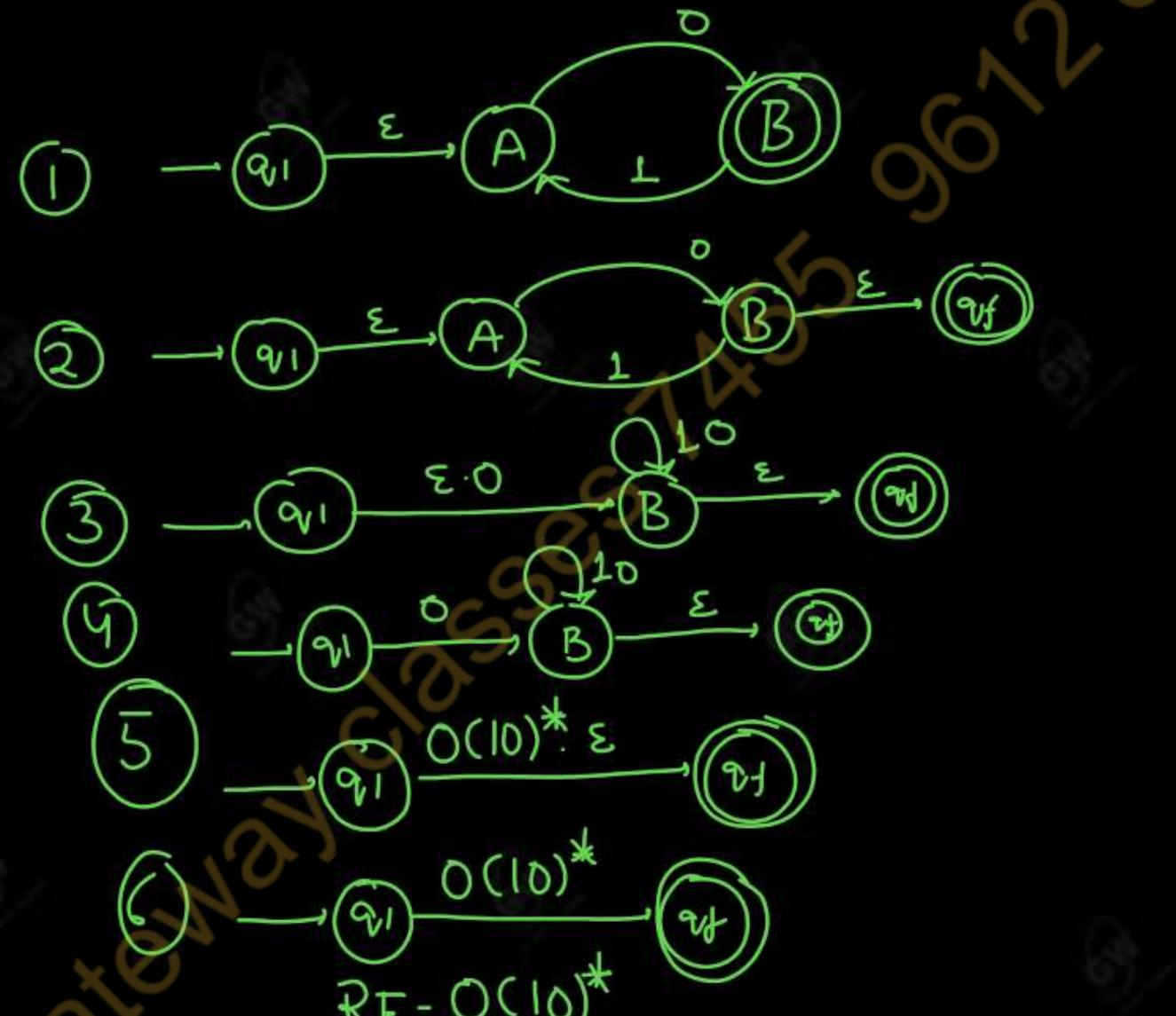
If a language is accepted by transition graph then it must be expressed by regular expression.

Kleene's theorem

PROOF



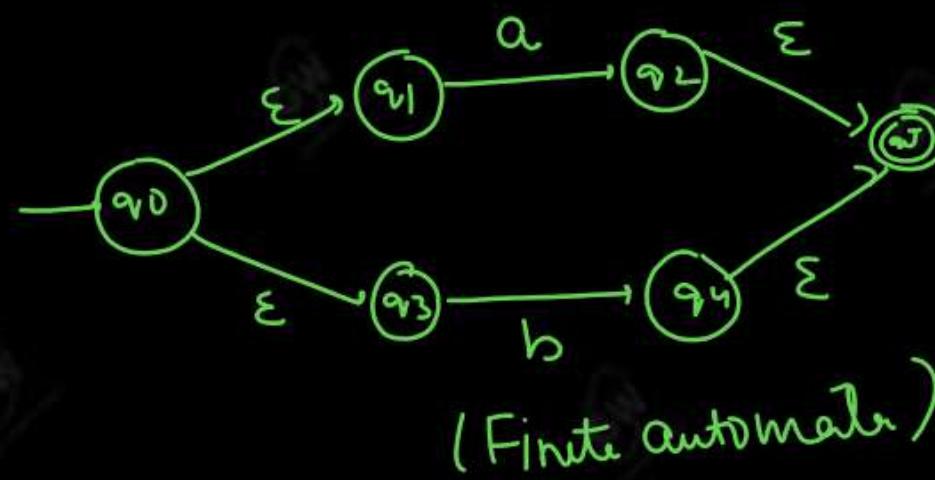
- ① Single final state
- ② No incoming edge to the initial state
- ③ No outgoing edge from final state



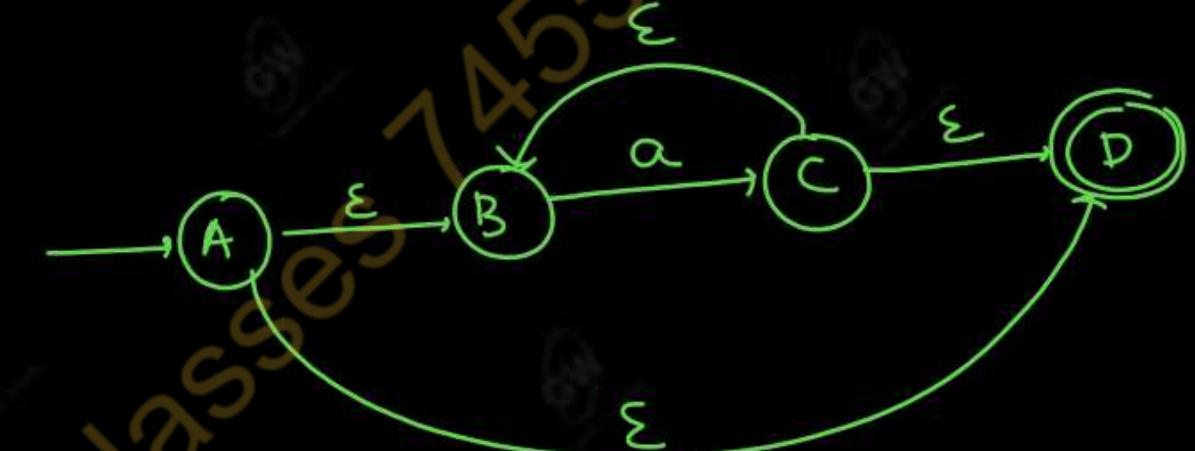
KLEENE THEORM PART 3

If a language is expressed by a regular expression then it can be accepted by a finite automata.

$$R.E = a+b$$



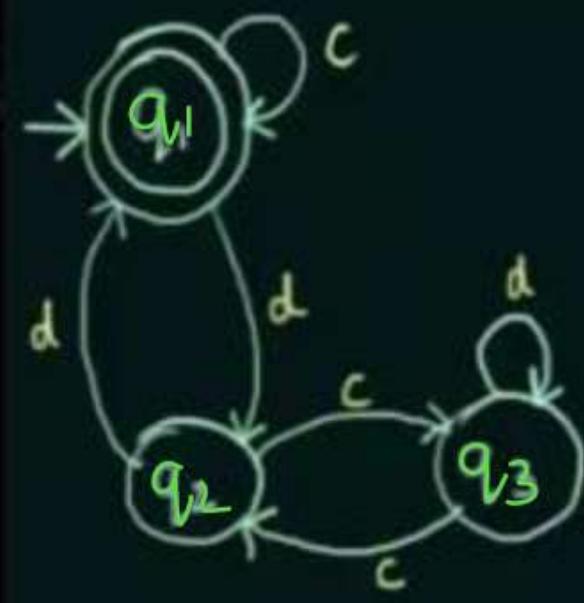
$$a^* = \{\epsilon, a, aa, aaa, \dots\}$$



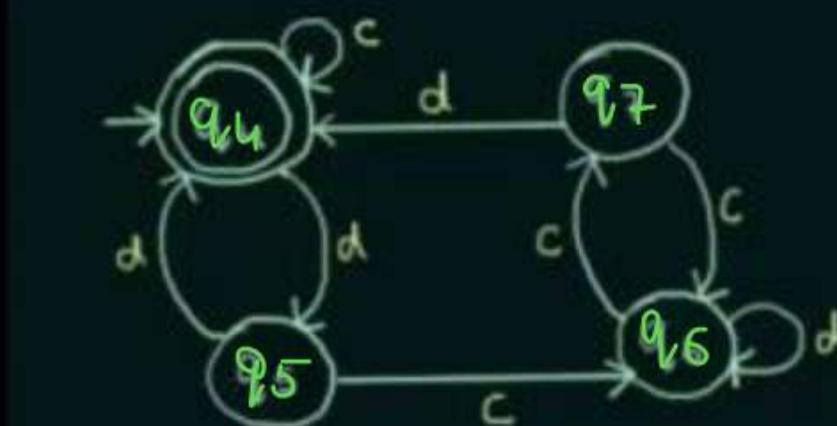
Equivalence of two finite automata

1. For any pair of states $\{q_i, q_j\}$ the transition for input $a \in \Sigma$ is defined by $\{q_a, q_b\}$ where $\delta(q_i, a) = q_a$ and $\delta(q_j, a) = q_b$. The two automata are not equivalent if for a pair $\{q_a, q_b\}$ one is intermediate state and other is final state.
2. If initial state is final state of one automation then in second automation also initial state must be the final state for them to be equivalent.

Equivalence of two finite automata



A



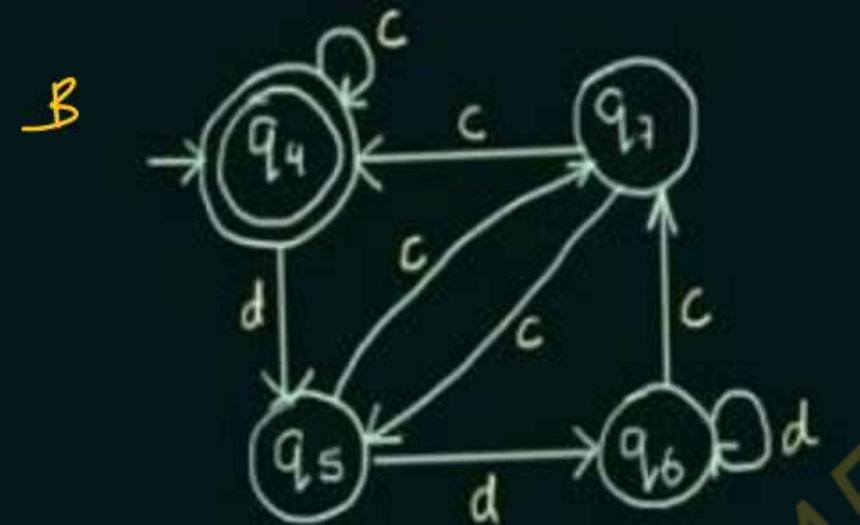
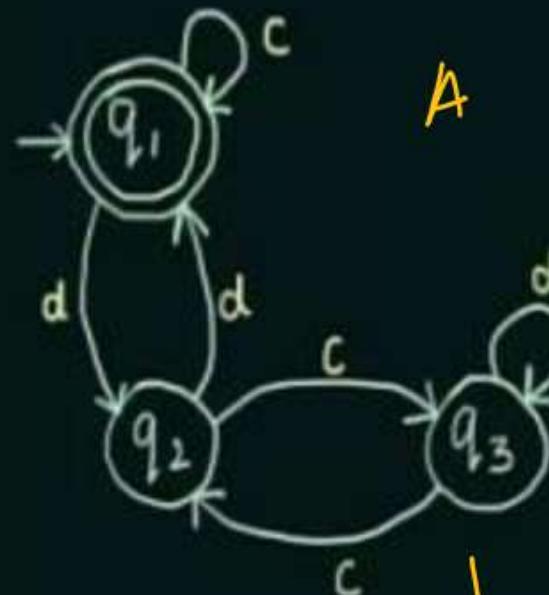
B

Initial state of automata A is q_1 & final state is q_1
 Initial state of automata B is q_4 & final state is q_4 \Rightarrow condⁿ

		c	d
$\{q_1, q_4\}$	c	$\{q_1, q_4\}$ FS FS	$\{q_2, q_5\}$ IS IS
	d	$\{q_1, q_4\}$ FS FS	$\{q_2, q_5\}$ IS IS
$\{q_2, q_5\}$	c	$\{q_3, q_6\}$ IS IS	$\{q_1, q_4\}$ FS FS
	d	$\{q_3, q_6\}$ IS IS	$\{q_3, q_6\}$ IS IS
$\{q_3, q_6\}$	c	$\{q_2, q_7\}$ IS IS	$\{q_3, q_6\}$ IS IS
	d	$\{q_3, q_6\}$ IS IS	$\{q_1, q_4\}$ FS FS

above two condition
are satisfied so we
can say that two automata
are equivalent to each
other.

Equivalence of two finite automata

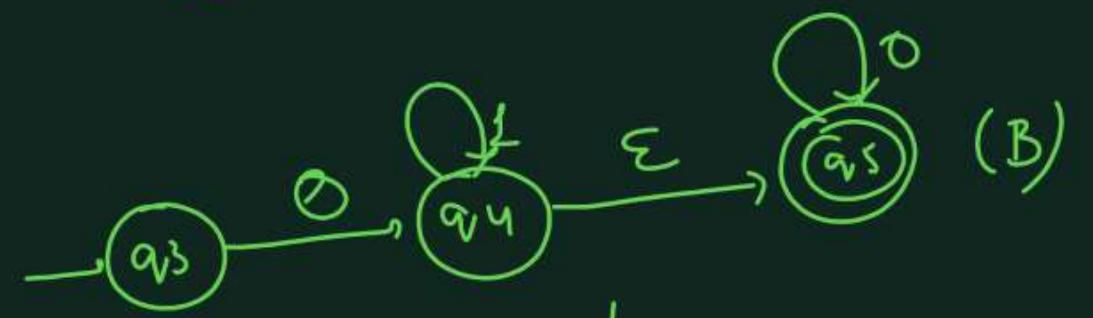
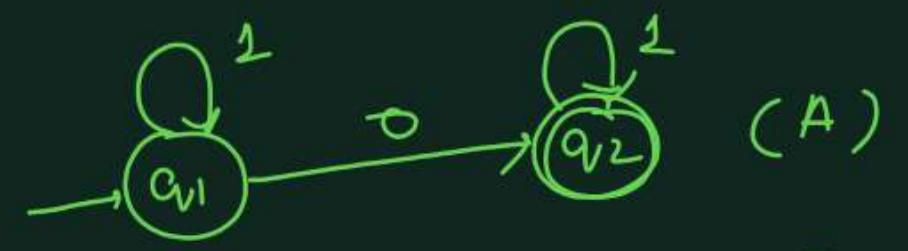


If initial & final state is same in one automata then the initial & final state is also same in another automata

$A \rightarrow q_1 \rightarrow$ initial & final
 $B \rightarrow q_4 \rightarrow$ initial & final

	c	d
$\{q_1, q_4\}$	$\{q_1, q_4\}$ FS FS	$\{q_2, q_5\}$ IS IS
$\{q_2, q_5\}$	q_3, q_7 IS IS	q_1, q_6 FS IS

→ this prove that A & B are equivalence.



$\{q_{v1}, q_{v3}\}$	$\{q_{v2}, q_{v4}\}$	$\{q_{v1}\}$
FS	IS	IS

this not equivalence

Gateway Classes 7455 9612 84

Pigeon hole principle

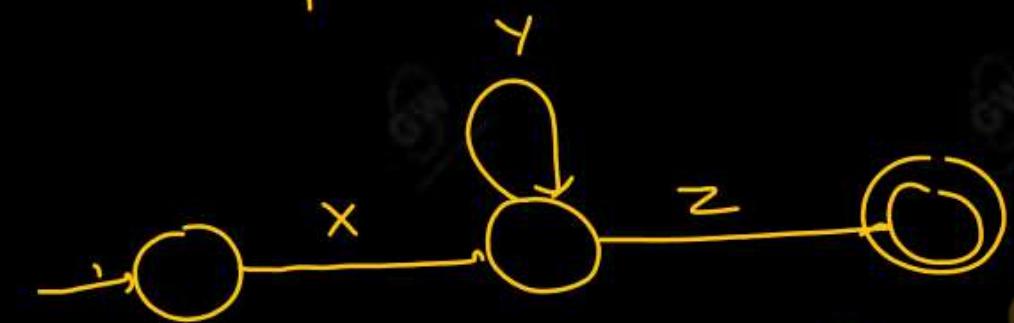
- If n Pigeon are assigned to m Pigeonhole then atleast one Pigeon hole contain two or more Pigeon $m < n$
- If one pigeon is assigned to each Pigeonhole then $n-m$ pigeons are left without assigned to any Pigeon hole

Gateway Classes 7455 961084

Proof of pumping lemma theorem



n states (DFA)
 $s = w_1 w_2 \dots w_p$
 $p \geq n$
 $n=3$



$$xz \in L$$

$$xy \in L$$

$$xy^2 \in L$$

$$xy^3 \in L$$

$$\vdots$$

$$s = xy^i z \in L \quad i \geq 0$$

$|y| > 0 \rightarrow$
this a DFA y never be ϵ
 $|xy| \leq p$

Thank You

961284
455
Gateway Classes



Gateway Classes



Full Courses Available in App

AKTU B.Tech I- Year : All Branches

AKTU B.Tech II- Year

Branches :

- 1. CS IT & Allied**
- 2. EC & Allied**
- 3. ME & Allied**
- 4. EE & Allied**

Download App Now



**Full
Courses**

- V. Lectures
- Pdf Notes
- AKTU PYQs
- DPP