AKTU

B. Tech II-Year

Gateway Classes

GW
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

AKTU B. Tech II-Year

Gateway Classes

BCS402 Theory of Automata and Formal Languages

Unit-2

Introduction to Regular Expressions and Languages

Syallbus

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

Regular language

- 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

- $L=\{a^m b^n \ m,n>=0\}$ yes (able to learn order)
  × No comparison, infinite
- $L=\{a^p b^q c^r \ p,q,r>=0\}$ yes   Order, No comparison, infinite
- $L=\{a^m b^n \ 1<=m<=500, 1<=n<=1000\}$ yes {language is finite, able to learn order}
- $L=\{a^n b^n \ n>=0\}$ no{ able to do counting / comparison ,infinite}
- $L=\{a^n b^n \ 1<=n<=10\}$ yes finite, comp
- $L=\{a^n b^n \ 1<=n<= 2^{29 \text{ prime number}}\}$ yes { comparison, finite}
  Counting
- $L=\{a^n b^n \ 1<=n<= 2^{|gate|}\}$ yes

Regular Expression

- A way of representing regular language
- Expression of string and operator
- *Kleene closure [a*] $\{\epsilon, a, aa, aaa \ldots\} \rightarrow \{a^0, a^1, a^2, a^3, a^4 \ldots\}$
- + positive closure[a+] $\{a, aa, aaa, aaaa \ldots\}$ $\{a^1, a^2, a^3, a^4 \ldots\}$
- . Concatenation[a.b] cde fi cde.fi
- + union[a+b] a+b $\{a, b\}$
- (a+b).(a+b) $(a+b).(a+b) = (a+b)^2$ – double length all string
  aa ab ba bb
- (a+b)* $= \{(a+b)^0, (a+b)^1, (a+b)^2, (a+b)^3 \}$
  $\{\epsilon, a, b, aa, ab, ba, bb, aaa, aab, aba \ldots\}$

Find out the language for the following regular expression

$a.\epsilon = a$

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. $r = a^+$, $L(r) = \{a, aa, aaa \ldots\}$
7. $r = a^*$, $L(r) = \{\epsilon, a, aa, aaa \ldots\}$
8. $r = (a + ba).(b+a)$, $L(r) = \{ab, aa, bab, baa\}$
9. $r = \Phi$, $L(r) = \phi \; \{ \}$
10. $r = \varepsilon$, $L(r) = \{\epsilon\}$
11. $(a+b).bb$ $\{abb, bbb\}$

12. $r = (a+b)^*$, $L(r) = \{\epsilon, a, b, ab, ba, bb \ldots\}$
13. $r = (a+b)^*(a+b)$, $L(r) = (a+b)^+ = \{a, b, aa, ba \ldots\}$
14. $r = a^*.a^*$, $L(r) = a^* = \{\epsilon, a, aa, aaa \ldots\}$
15. $r = (ab)^*$, $L(r) = \{\epsilon, ab, abab, ababab \ldots\}$
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} → $ab.(a+b)^*$
2. L={ start with bba} $bba(a+b)^*$
3. L={ end with abb} $(a+b)^*.abb$
4. L={ contain a substring aab} $(a+b)^*.aab.(a+b)^*$
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} $(a+b).(a+b).(a+b)$
9. L={|w|>=3} $(a+b).(a+b).(a+b).(a+b)^*$
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. $L=\{ |w|_a=2\}$ $b^* a b^* a b^*$
2. $L=\{ |w|_a>=2\}$ $(a+b)^* a (a+b)^* a (a+b)^*$
3. $L=\{ |w|_a<=2\}$ $b^* + b^* a b^* + b^* a b^* a b^*$ | $b^* (a+\epsilon) b^* (a+\epsilon) b^*$
4. 3^{rd} symbol from the left end is b $(a+b)^2 b (a+b)^*$
5. 28 the symbol from the right end is a
   $(a+b)^* a (a+b)^{27}$
6. $|w|=0$ mod3 $[(a+b)^3]^*$
7. $|w|=2$mod3 $(a+b)^2 [(a+b)^3]^*$
8. $|w|_b=0$ mod2 $a^* (a^* b a^* b a^*)^*$ |
   $a^* + (a^* b a^* b a^*)^*$
9. $|w|_a= 1$ mod3 $b^* a b^* (b^* a b^* a b^* a b^*)^*$
10. $|w|_b=2$ mod3

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

Find out the regular expression for the following regular language

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

1. Language accepting the string of length 2

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

2. Language accepting the string of length at least 2

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

3. Language accepting the string of length at most 2

$\longrightarrow \epsilon + a + b + (a+b).(a+b)$

$\longrightarrow \epsilon + a + b + (a+b)^2$

$\longrightarrow (\epsilon + a + b).(\epsilon + a + b)$

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

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 $(0+1)^* 00 (0+1)^*$

3. Having two consecutive zero or one

$(0+1)^* 00 (0+1)^* + (0+1)^* 11 (0+1)^*$

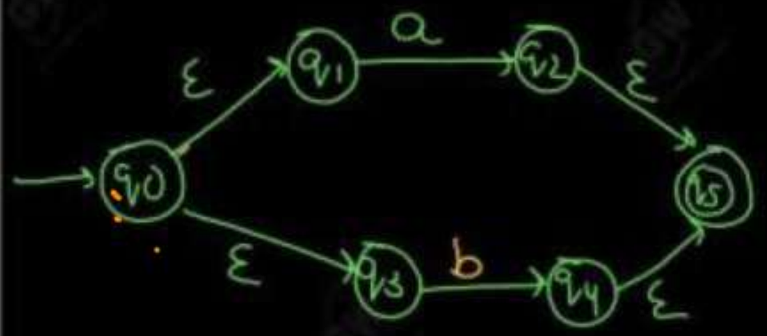
Convert regular expression to epsilon NFA

Φ ,ε are primitive regular expression

Φ

ε

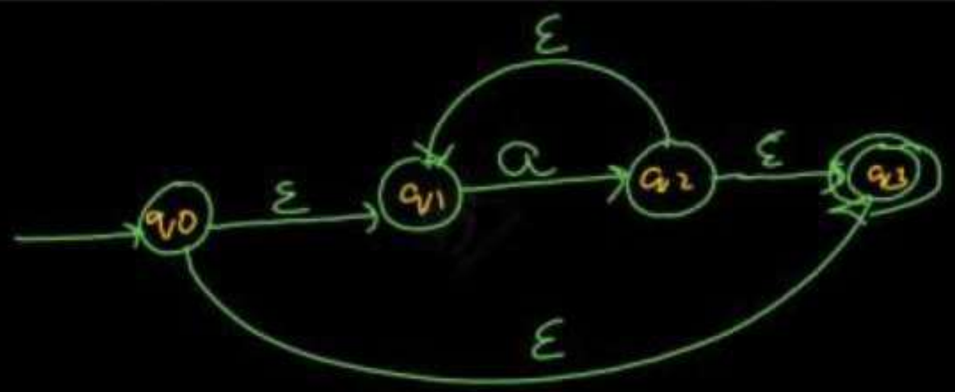
a+b



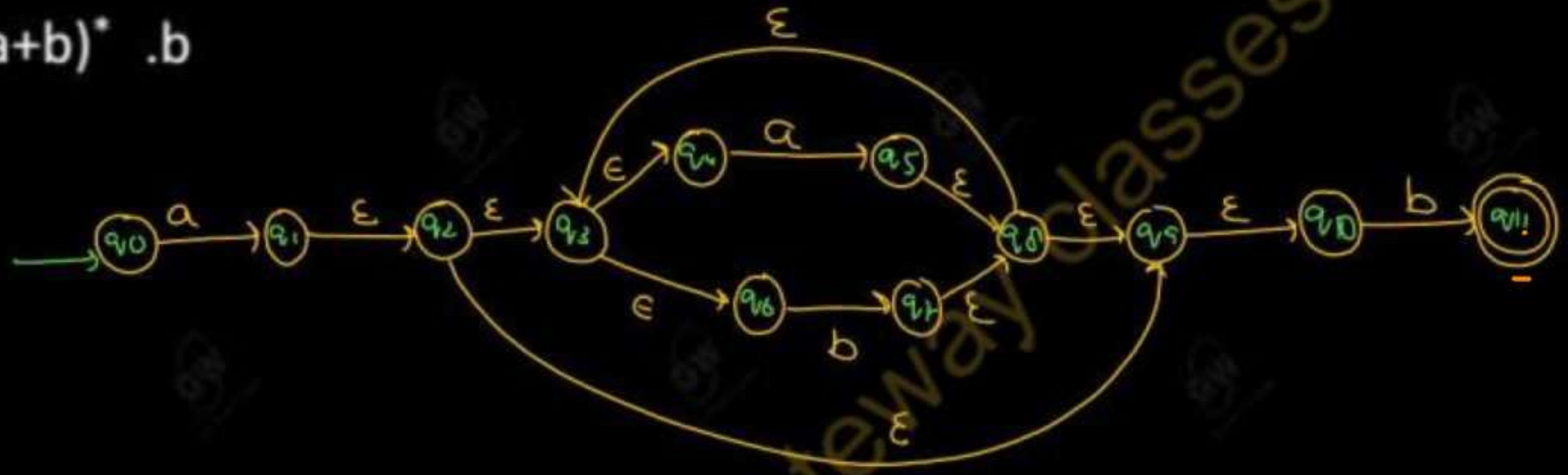
a.b

Convert regular expression to epsilon NFA

a^*



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

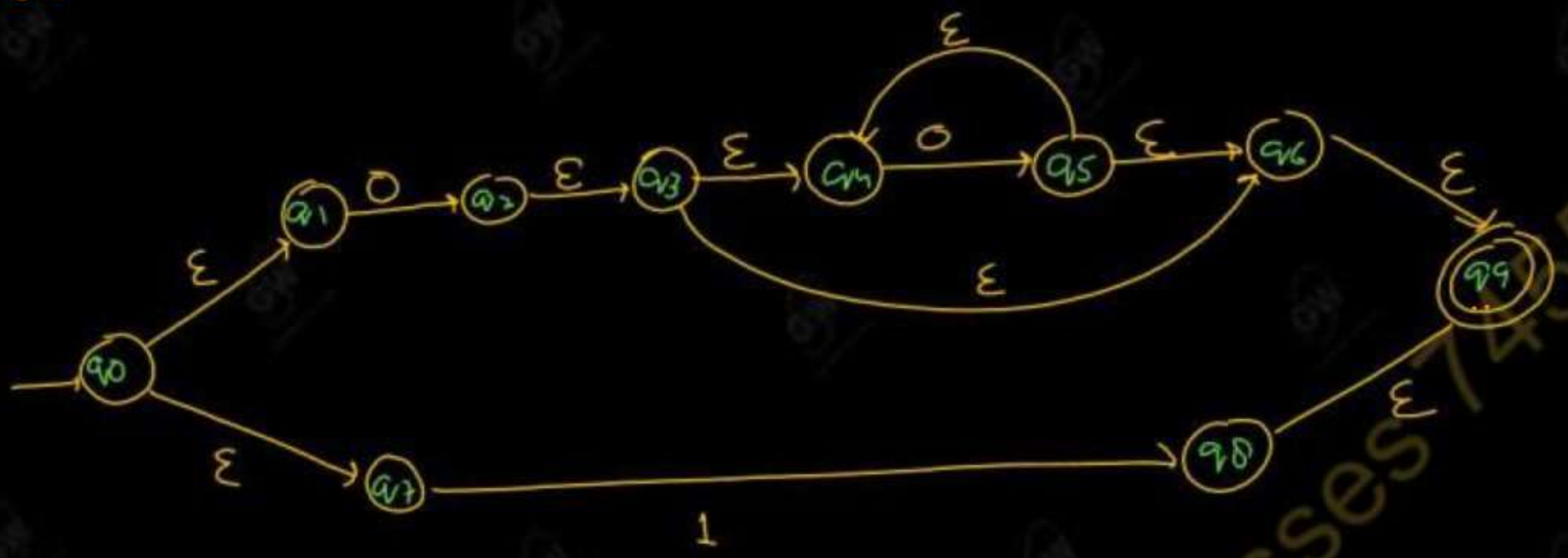


Convert regular expression to epsilon NFA

GW

$\underline{0}0^* + 1$

$a + b$



Convert the following Regular Expression into DFA (Finite Automata)

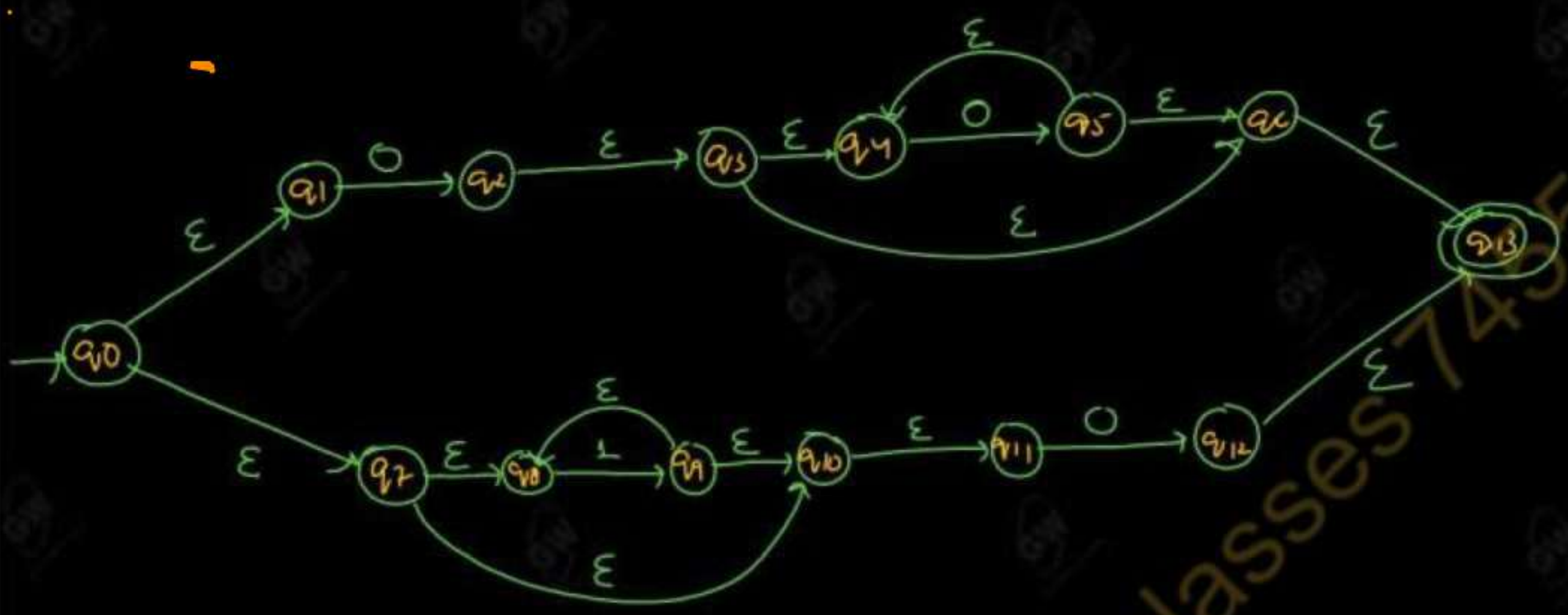
a^*

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

- → ε NFA
- → ε-NFA to DFA
- → Minimization of DFA

Convert regular expression to epsilon NFA

$00^* + 1^*0$



Convert regular expression to epsilon NFA

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

Convert regular expression to epsilon NFA

$(0+1^*)^* 2^* 3^*$

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

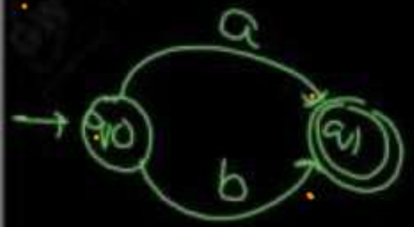
Convert regular expression to NFA

a

ε

a* {ε, a, aa, aaa ----}

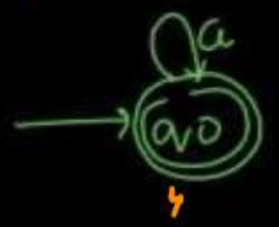
a+b



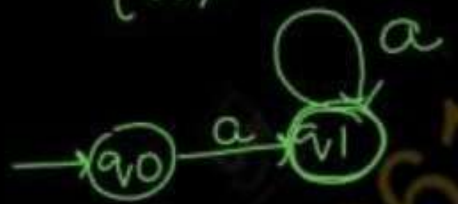
a.b

Convert regular expression to NFA

a* $\{\epsilon, a, aa$



a+ $\{a, aa, aaa - -$



(a+b)*

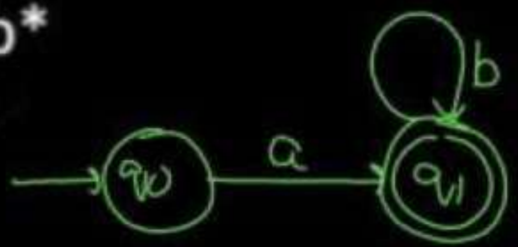
(a.b)*

Convert regular expression to NFA

a^*b

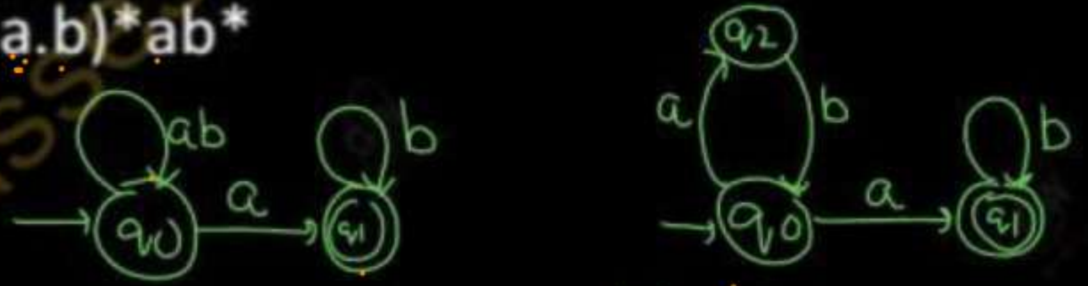
$a^*b(a+b)^*$

ab^*



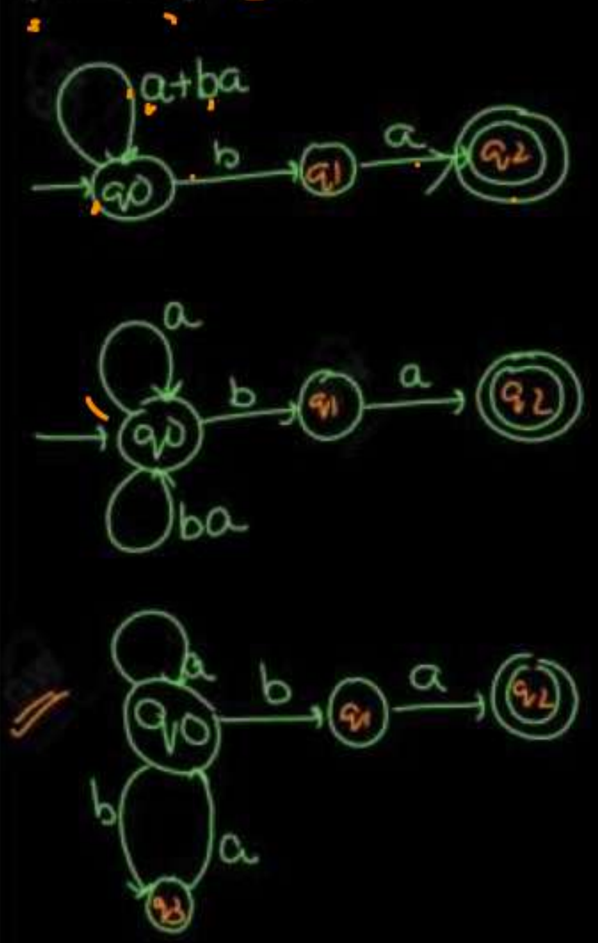
$(ab)^*$	$a\,b^*$
ε	ε → a
ab	b
abab	bbb
ababab	bbb

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



Convert regular expression to NFA

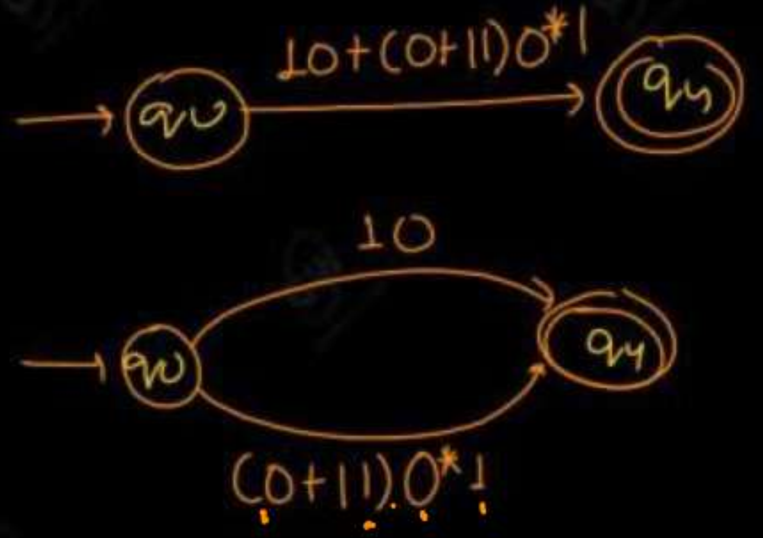
(a+ba)*ba



(aa+aaa)*

Convert regular expression to NFA

10+(0+11)0*1

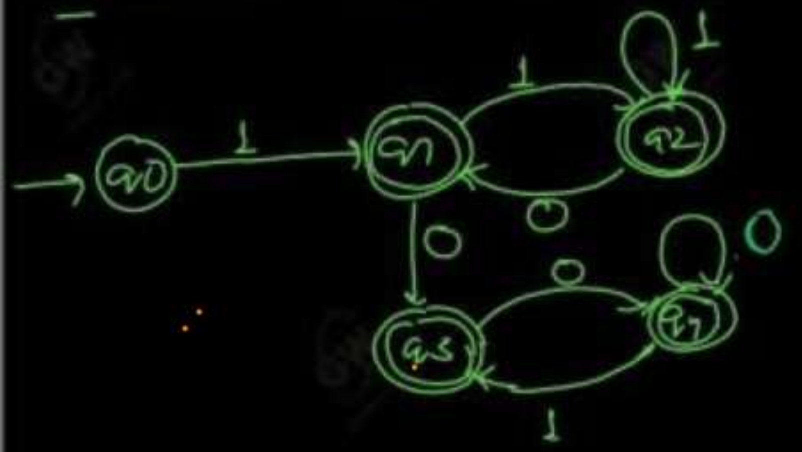


Convert regular expression to NFA

1(1*01*01*)*

Convert regular expression to NFA

$1(1+10)^*+10(0+01)^*$



CLOSURE PROPERTIES OF REGULAR LANGUAGE

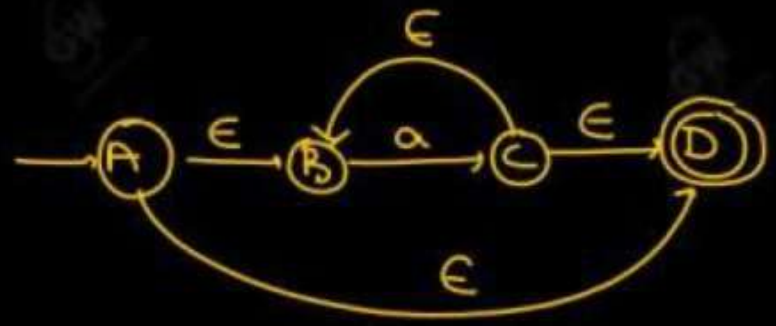
1 The union of two regular language is regular

$\Sigma \cdot \{a, b\}$ $\Sigma^* = \{\epsilon, a, b, ab, ba, aa, ---\}$

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

R.E → a^* (r_1)

ε-NFA

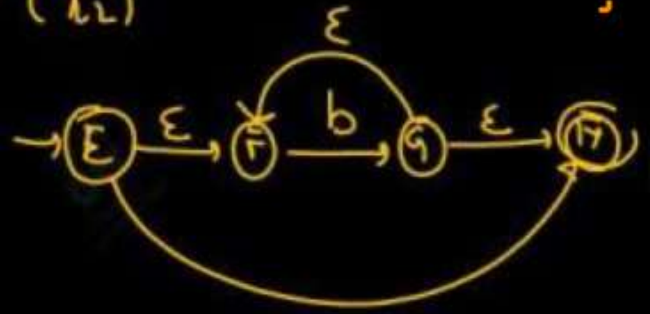


L1 is regular language b/c we can make ε-NFA/R.E

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

R.E= b^* (r_2)

ε-NFA



L2 is regular languag

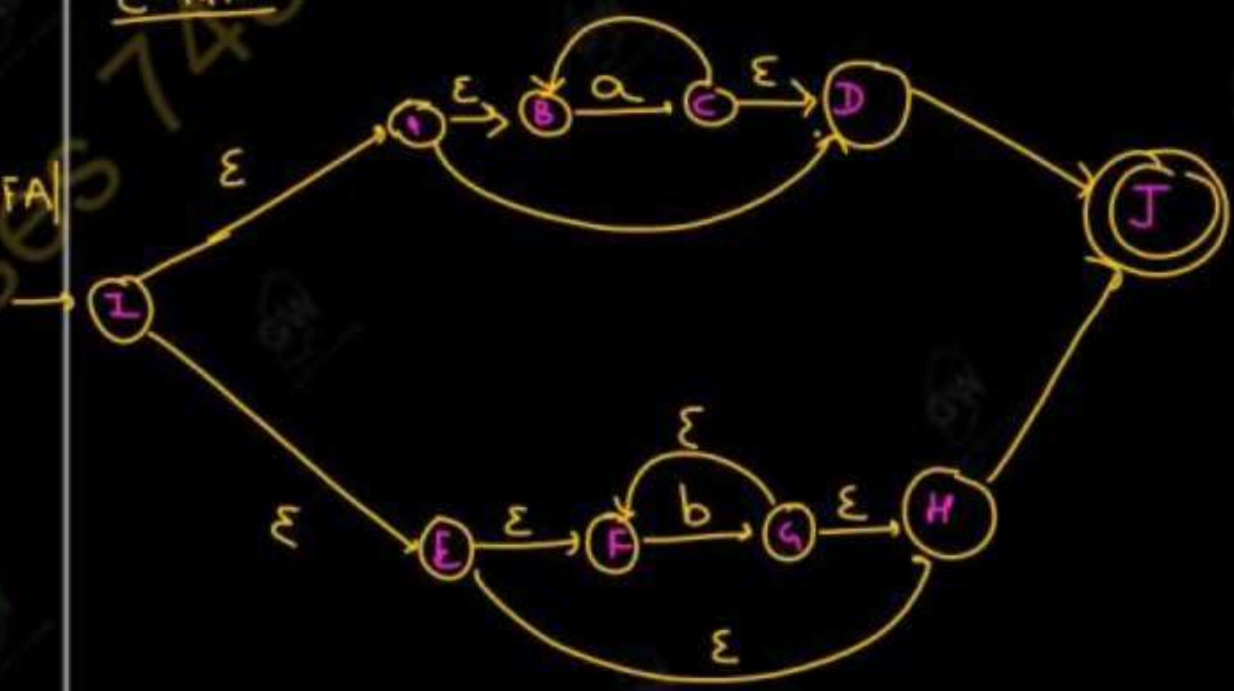
$L_1 \cup L_2$ = Regular

$L_1 \cup L_2 = \{r_1 + r_2\}$

$a^* + b^*$ (r_3)

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

ε-NFA



→ union of two Regular Lang is Regular

$r_3 = r_1 + r_2$

↑

Regular

2. The complement of the regular language is also regular

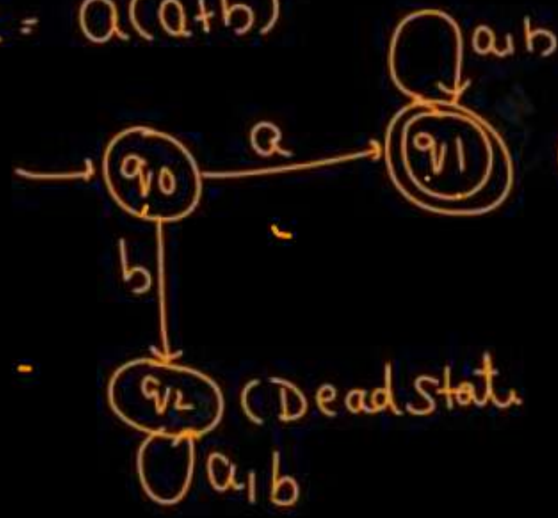
$\Sigma = \{a, b\}$

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

$L_1 = \{a, ab, aab, abaa, aa ---\}$

$L_1 = \{W \mid W$ is start with $a\}$

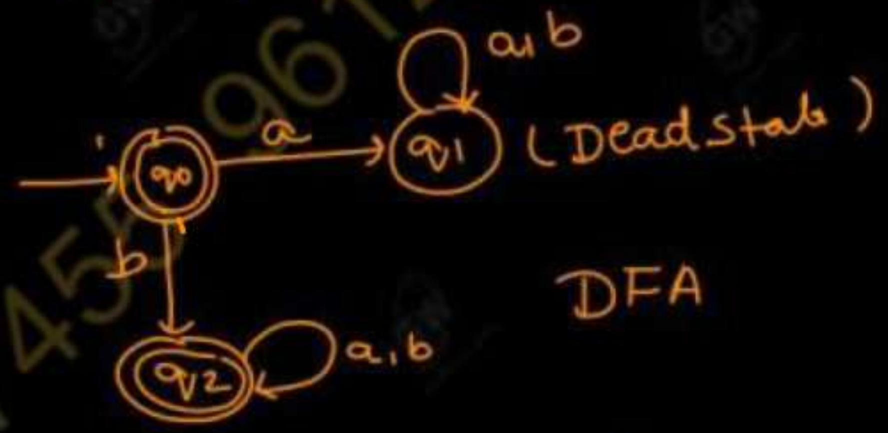
R.E $= a(a+b)^*$



L_1 is regular language

$\overline{L_1} = \{\epsilon, b, ba, bb, baba ---\}$

$\overline{L_1} = \{W \mid W$ is not start with $a\}$



DFA

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

Complement of Regular language is Regular.

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

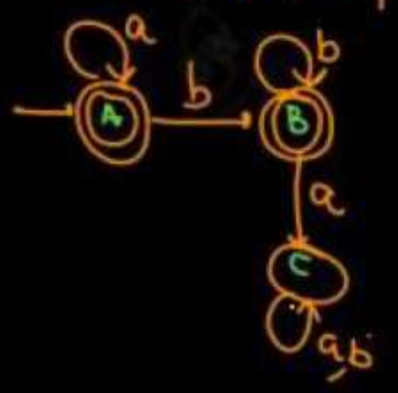
Reg → Regular

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

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

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

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



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

L_1 is Regular

$L_2 = \{a^n \mid n \geq 0\}$

$L_2 = \{\epsilon, a, aa, aaa \ldots\}$

$R.E = a^*$

L_2 is regular language

$L_1 \cap L_2 \ \{a^n \mid n \geq 0\}$

$R.E = a^*$

The intersection of two Regular lang is Regular

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 $\cap$ $\overline{\text{Regular}}$

$\therefore$ the complement of Regular lang is Regular

$\rightarrow$ Regular $\cap$ Regular

$\rightarrow$ Regular

$\Sigma = \{a\}$

$L_1 = \{\epsilon, a, aa, aaa, aaaa \text{ ------ } \ldots\}$

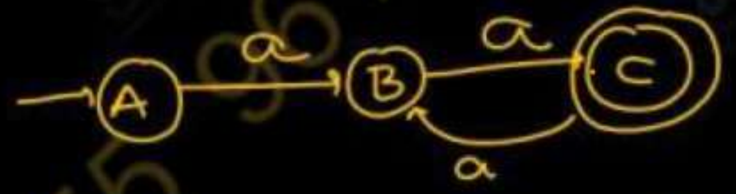
R.E $= a^*$

L_1 = Regular language

$L_2 = \{aa, aaaa, aaaaaa \text{ - - - - } \}$

$(aa)^+ \rightarrow$ R.E

L_2 = Regular language



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

ϵ a, aaa, aaaaa, aaaaaaa

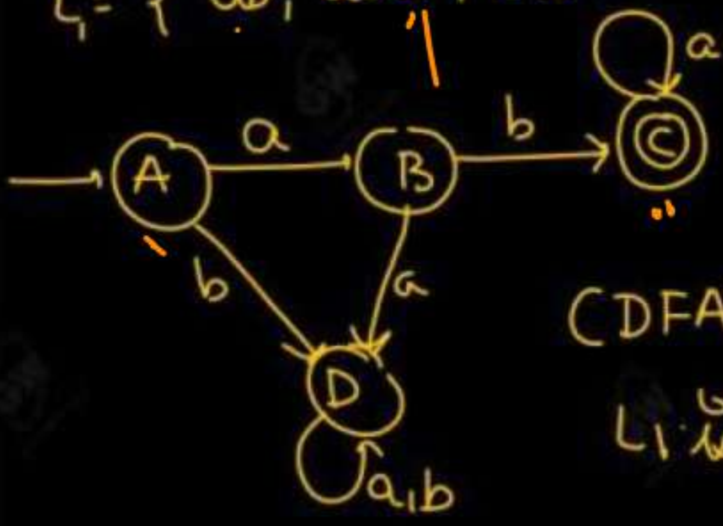
$L_1 - L_2$ - Regular lang

5. The reversal of a regular language is also regular

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

$L_1 = \{ W \mid W$ is start with $ab \}$

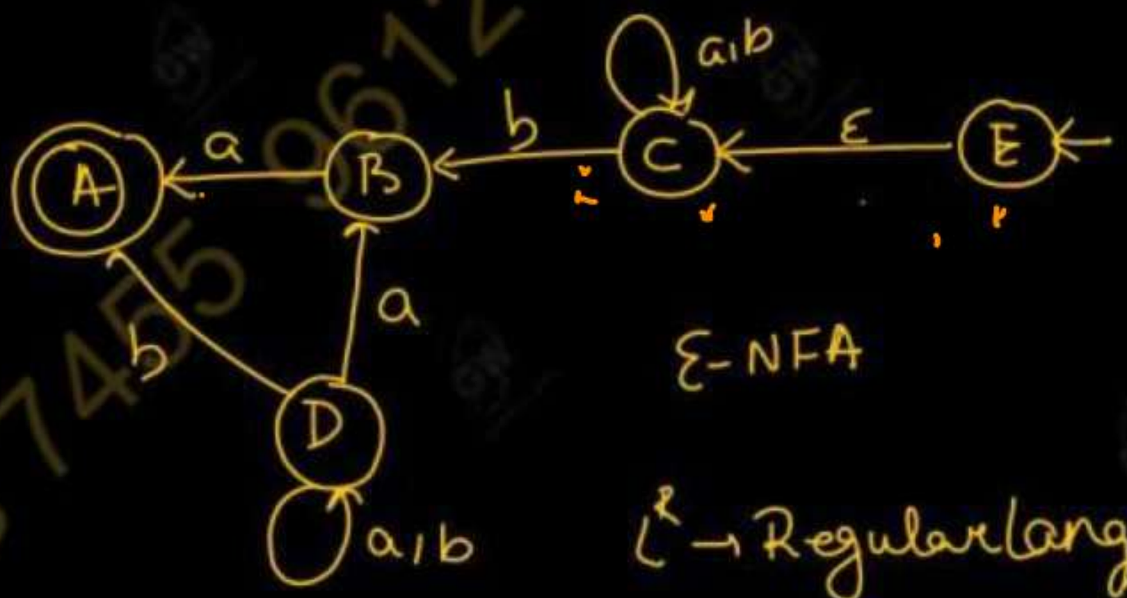
$L_1 = \{ ab, aba, abaa \ldots abbb - - - \}$



(DFA)

L_1 is Regular

$L^R = \{ ba, aba, ba \ldots \}$



ε-NFA

$L^R \rightarrow$ Regular lang

CLOSURE PROPERTIES OF REGULAR LANGUAGE

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

$L = \{a\}$ $\quad r_1 = a$

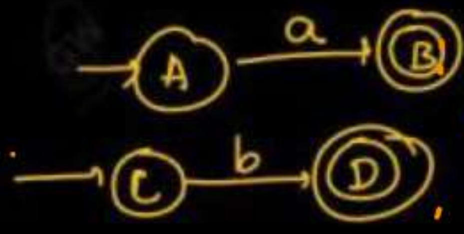
$r_1^* = a^*$

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

7. The Concatenation of two regular language is also regular

$L_1 = \{a\}.$ $\quad r_1 = a.$

$L_2 = \{b\}$ $\quad r_2 = b.$



Regular language,

Regular language

$r_1 \cdot r_2 = a \cdot b$

$L_3 = \{ab\}$

L_3 is also regular language

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

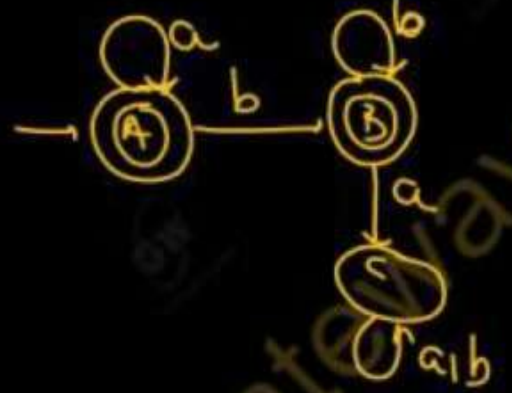
$h : \Sigma \to \Gamma^*$ Tau

$\Sigma = \{a, b\}$
$\Gamma = \{0, 1, 2\}$
$h(a) = 010$
$h(b) = 102$

$h(xyz) = h(x) \cdot h(y) \cdot h(z)$
$h(\varepsilon) = \varepsilon$

$h(abba)$
$h(a) \cdot h(b) \cdot h(b) \cdot h(a)$
$010 \quad 102 \quad 102 \quad 010$

$L_1 = \{a^n b^m \mid n, m \geq 0\}$
$a^* b^*$ – R.E
L is Regular



$h(a) = x$
$h(b) = y$

$L_2 = \{x^n y^m \mid n, m \geq 0\}$

$x^* y^*$

L2 is also Regular

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

$\Sigma = \{a b\}$

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

$h(a) = 0$

$h(b) = 12$

$L = \{0\ 12\ 12\ 0\}$

$h^{-1}(L)$ a b b a

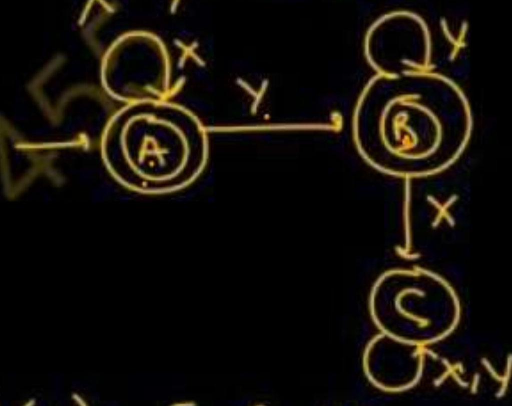
h^{-1} is inverse of homomorphism

$h(a) = x$

$h(b) = y$

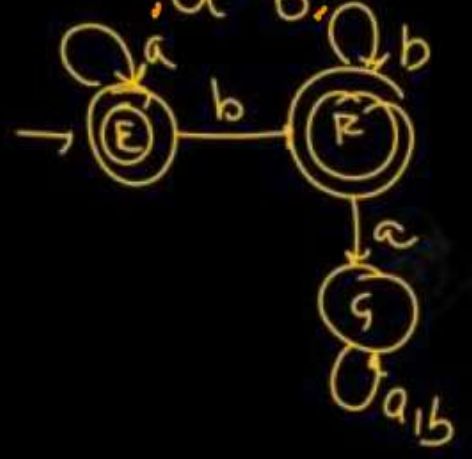
$L = \{x^n y^m, n,m \geq 0\}$

$x^* y^*$ — RE



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

$a^* b^*$



Identities for regular expression

1. $\emptyset + r = r$
2. $\emptyset.r = r.\emptyset = \emptyset$
3. $\in.r = r.\in = r$
4. $\in^* = \in$ and $\emptyset^* = \in$

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

Identities for regular expression

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

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

12. $(p+ q).r= p.r+ q.r$ and $r.(p+q) = r.p + r.q$

SOME MORE IMPORTANT PROOF

1. $r^+ \cup r^* = r^*$
2. $r^+ \cap r^* = r^+$
3. $(r^*)^+ = (r^*)$
4. $(r^+)^* = (r^*)$
5. $((r^*)^+)^* . r^+ = r^+$

6 $(a+b)^* = (a^* + b)^*$

7 $(a+b)^* = (a + b^*)^*$

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

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

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

Proof the following

$(1+00^*1)+(1+00^*1)(0+10^*1)^*(0+10^*1)=(0^*1(0+10^*1)^*)$

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

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

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

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

$\underbrace{(\epsilon+00^*)}_{\epsilon+RR^*}1\ (0+10^*1)^*$

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

$\epsilon+\underbrace{1^*(011)^*}_{R}\underbrace{[1^*(011)^*]^*}_{R^*}=(1+011)^*$

$[1^*(011)^*]^*$

$(P+Q)^*=(P^*Q^*)^*$

$=(1+011)^*$

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

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

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

$P=1$

$P^*=1^*$

$Q=(011)$

$Q^*\ (011)^*$

$(P^*Q^*)^*\ (1^*(011)^*)^*$

Arden's theorm

It states that-

Let P and Q be two regular expressions over Σ.

- If P does not contain a null string $\in$, 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

$R = Q + RP$
$R = Q + QP^*P$
$Q(\epsilon + P^*P)$
QP^*
this is the solution

$R = QP^*$

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

$R = Q + RP$
$Q + (Q + RP)P$
$Q + QP + RP^2$
$Q + QP + (Q + RP)P^2$
$Q + QP + QP^2 + RP^3$
$Q + QP + QP^2 + \cdots QP^n + RP^{n+1}$
$Q + QP + QP^2 + \cdots + QP^n + QP^*P^{n+1}$
$Q(\epsilon + P + P^2 + \cdots + P^n + P^*P^{n+1})$
QP^*
this proof this the unique solution

$R = Q + RP$

$\therefore R = QP^*$

NOTE-	PROCEDURE TO FIND REGULAR EXPRESSION

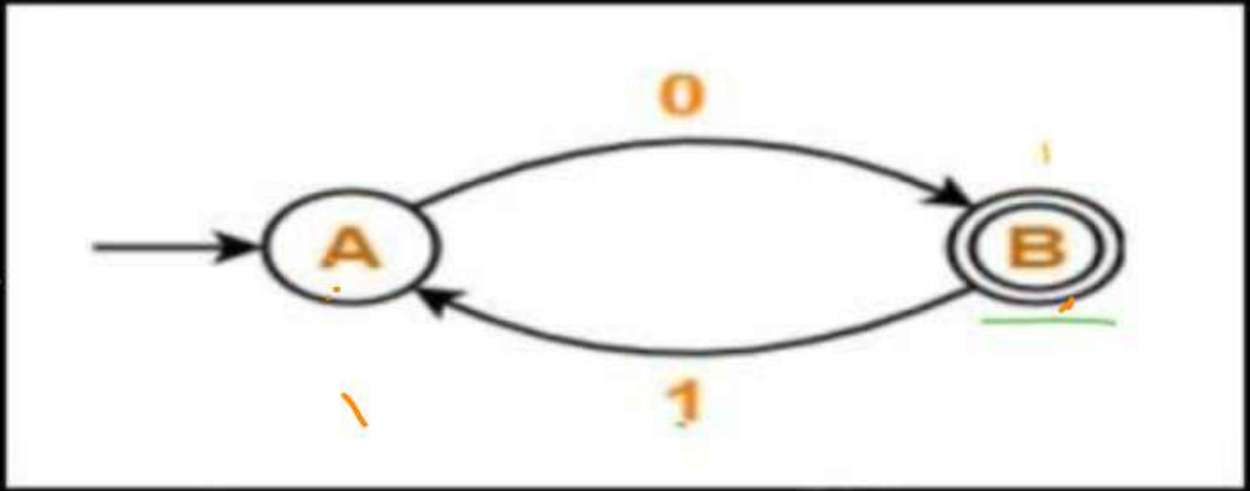
NOTE-

- 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



$A = \epsilon + B \cdot 1$ —— ①

$B = A \cdot 0$ —— ②

Put A value in eq²

$B = (\epsilon + B \cdot 1) \cdot 0$

$B = \epsilon \cdot 0 + B \cdot 1 \cdot 0$

$B = 0 + B \cdot 1 \cdot 0$

$R \quad Q + R \; \overline{P}$

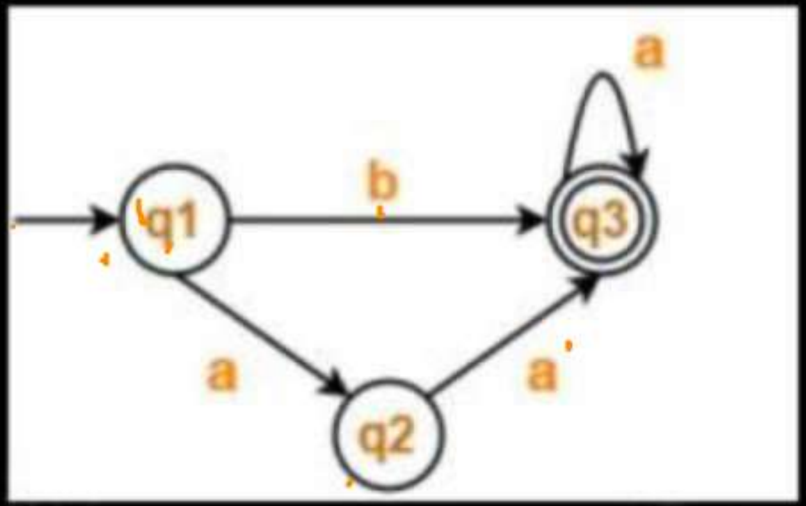
$R = Q + RP$

QP^*

$\epsilon \cdot R = R$

$\boxed{B = 0(10)^*}$

EXAMPLE 2



$R = Q + RP$

$R = QP^*$

$q1 = \epsilon$ ——— ①

$q2 = q1\,a$ ——— ②

$q3 = q1.b + q2.a + q3.a$ ——— ③

Put q1 value in equation 2

$q2 = \epsilon.a = a$

$\boxed{q2 = a}$ ——— ④

$q3 = q1.b + q2.a + q3.a$

Put q1 value from eq 1

Put q2 value from eq 2

$q3 = \epsilon.b + a.a + q3.a$

$b + a.a + q3.a$

$q3 = (b+aa) + q3.a$

R Q + R P

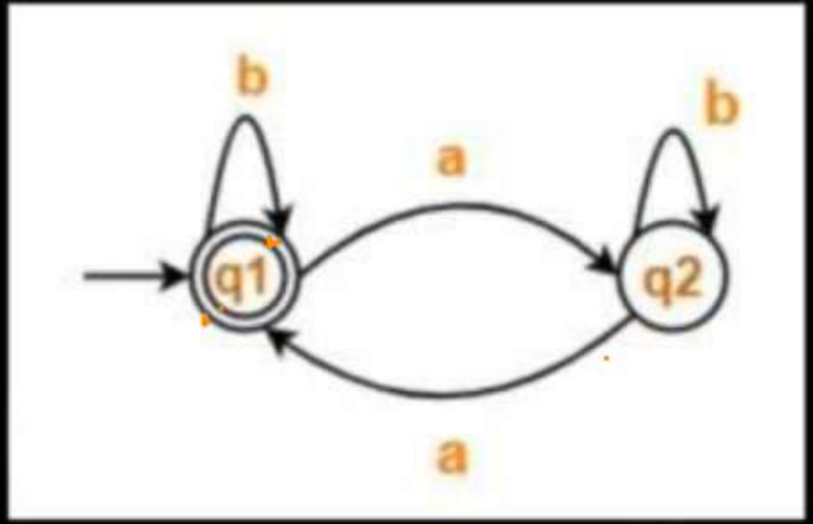
$Q = b + aa$

$P = a$

$q3 = (b+aa).a^*$

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

EXAMPLE 3



$q_1 = \epsilon + q_1 b + q_2 a$ —— (1)

$q_2 = q_1 a + q_2 b$ —— (2)

$R = Q + RP$

$q_2 = q_1 a b^*$ —— (3)

Put q_2 value in eq 1

$q_1 = \epsilon + q_1 b + q_2 a$

$\epsilon + q_1 b + q_1 a b^* a$

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

$R = Q + RP$

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

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

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

$b + ab^*$

$a.\epsilon = a$

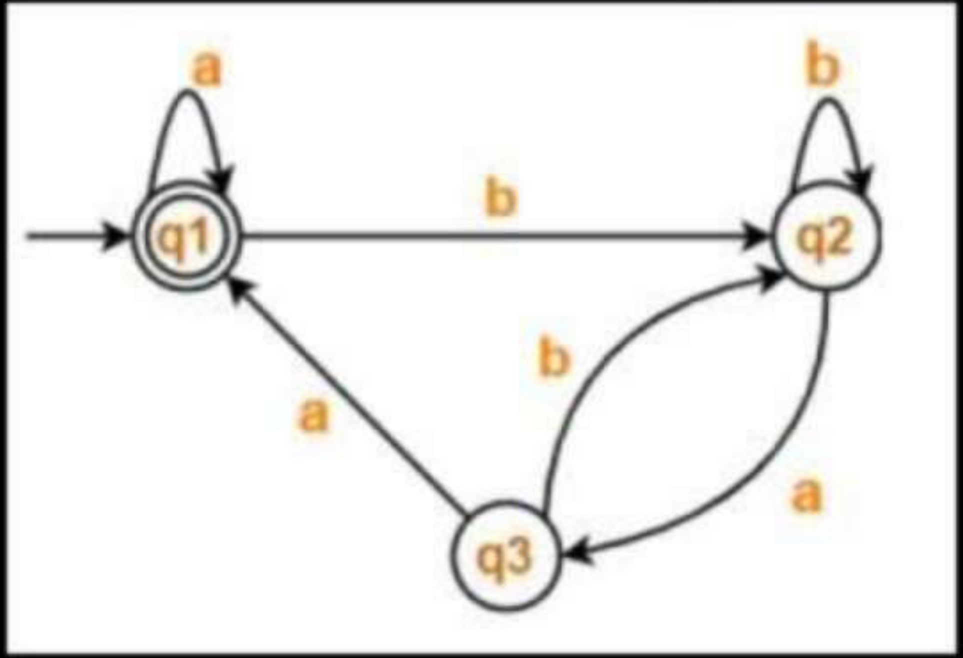
$a.b = ab$

$a.bb = abb$

$a\ b\ ab\ abbb$

$q_2 = q_1 a b^*$

EXAMPLE 4



$q_1 = \epsilon + q_1 a + q_3 a$ —— (1)

$q_2 = q_1 b + q_2 b + q_3 b$ —— (2)

$q_3 = q_2 a$ —— (3)

Put eq 3 in eq 2

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

$q_1 b + q_2 b + q_2 ab$

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

$R = Q + R P$

$q_2 = q_1 b (b+ab)^*$ —— (4)

Put equation 4 in eq 3

$q_3 = q_2 a$

$q_3 = q_1 b (b+ab)^* a$ —— (5)

Put equation 5 in eq (1)

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

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

$R = Q + R P$

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

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

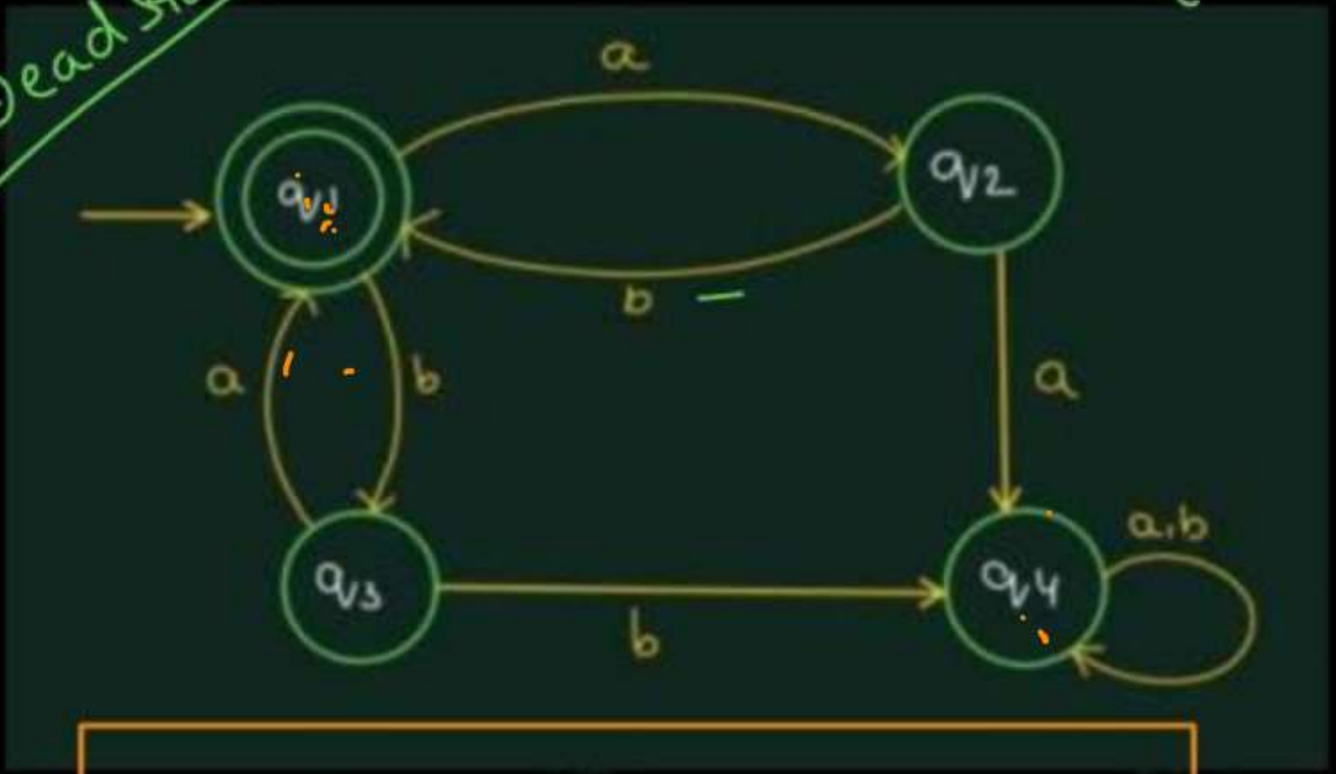
RE

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

EXAMPLE 1

find out the R.E from the Given finite Automata

Dead state



$q_1 = \epsilon + q_2 b + q_3 a$ —— (1)

$q_2 = q_1 a$ —— (2)

$q_3 = q_1 b$ —— (3)

$q_4 = q_2 a + q_3 b + q_4 a + q_4 b$ —— (4)

Put q_2 & q_3 value in equation 1

$q_1 = \epsilon + q_1 ab + q_1 ba$

$q_1 = \epsilon + q_1(ab + ba)$

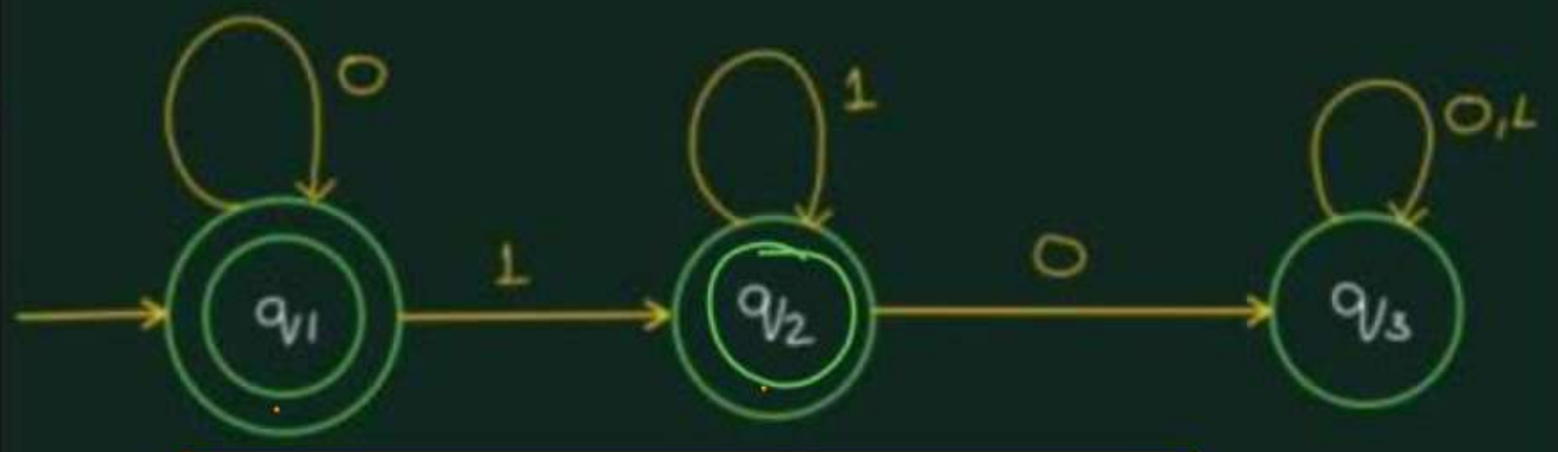
$R = Q + R\,P$

$q_1 = \epsilon\,(ab+ba)^*$

$q_1 = (ab+ba)^*$

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

EXAMPLE 2



$q_1 = \epsilon + q_1 0$ — ①

$q_2 = q_1 \cdot 1 + q_2 \cdot 1$ — ②

$q_3 = q_2 \cdot 0 + q_3 0 + q_3 \cdot 1$ — ③

if q_1 is my final state

$q_1 = \epsilon + q_1 . 0$

$R = Q + RP$

$q_1 = \epsilon . 0^*$

$\boxed{q_1 = 0^*} \longrightarrow \boxed{R.E_1 = 0^*}$

if q_1 q_2 are my both final state

$q_2 = q_1 \cdot 1 + q_2 \cdot 1$

$q_2 = 0^* 1 + q_2 \cdot 1$

$R = Q + RP$

$q_2 = 0^* 1 1^*$

$\boxed{RE_2 = 0^* 1 1^*}$

$\boxed{\begin{array}{l} \epsilon + RR^* = R^* \\ \epsilon + R^*R = R^* \end{array}}$

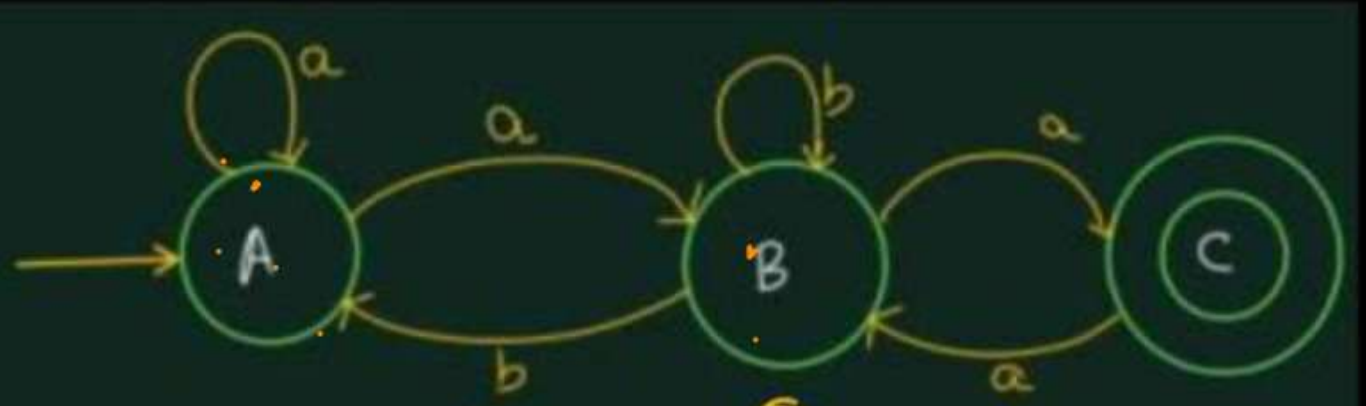
$R.E = RE_1 + R.E_2$

$0^* + 0^* 1 1^*$

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

$\boxed{0^* 1^*}$

EXAMPLE 5



$A = \epsilon + A \cdot a + B \cdot b$ —— ①

$B = A \cdot a + B \cdot b + C \cdot a$ —— ②

$C = B \cdot a$ —— ③

Put eq3 in eq2

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

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

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

$R = Q + RP$

$B = Aa(b+aa)^*$ —— ④

Put eq4 in eq①

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

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

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

$A = \epsilon + A(a + a(b+aa)^* b)^*$ —— ⑤

$A = \epsilon \cdot (a + a(b+aa)^* b)^*$ —— ⑥

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

Put eq6 in eq4

$B = (a + a(b+aa)^* b)^* a(b+aa)^*$ —— ⑦

Put eq7 in eq3

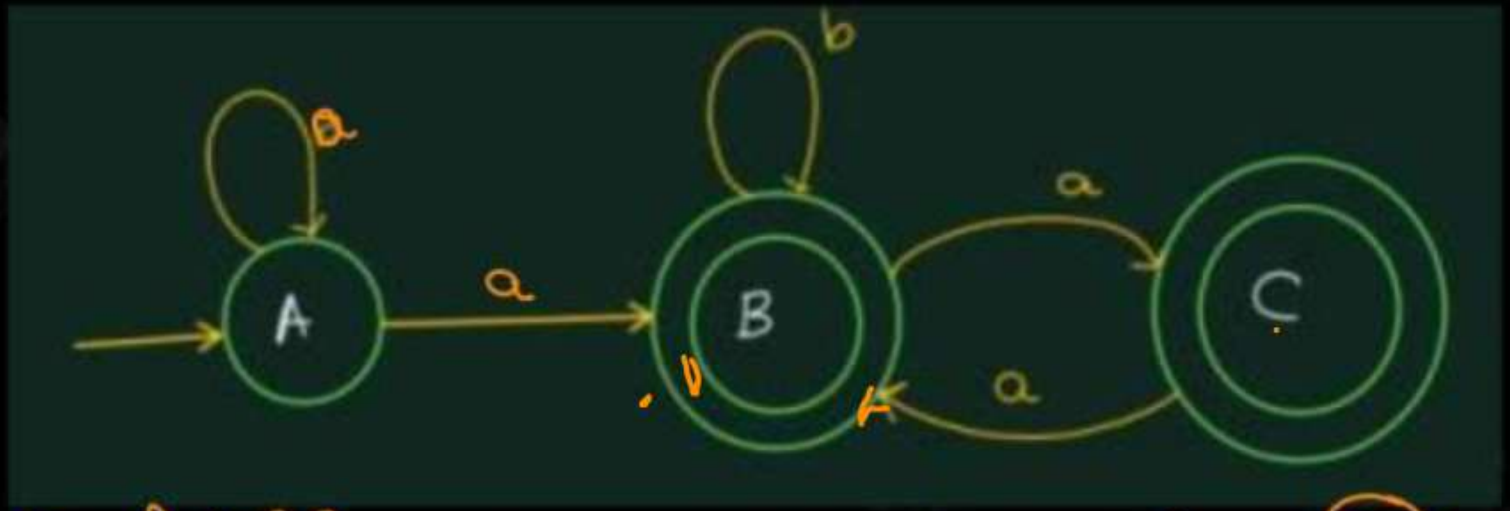
$C = B \, a$

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

$\downarrow$

$R \cdot E$

EXAMPLE 6



$R = Q + RP$

$A = \epsilon + A \cdot a$ —— ①

$B = A \cdot a + B \cdot b + C \cdot a$ —— ②

$C = B \cdot a$ —— ③

Put eq3 in eq2

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

$R = Q + RP$

$A = \epsilon \cdot a^*$ $\quad A = a^*$ —— ⑤

$B = Aa(b+aa)^*$ —— ⑥

Put eq5 in eq6

$B = a^* a(b+aa)^*$ —— ⑦

Put equation 7 in eq3

$C = B\,a$

$\boxed{C = a^* a(b+aa)^* a}$ —— ⑧

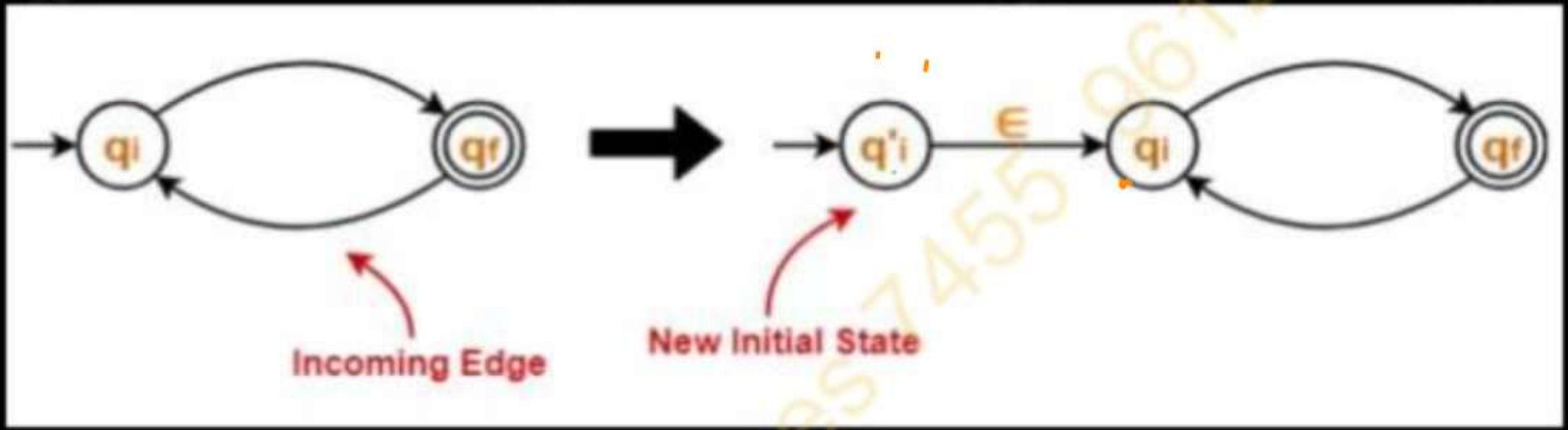
$R.E = B + C$

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

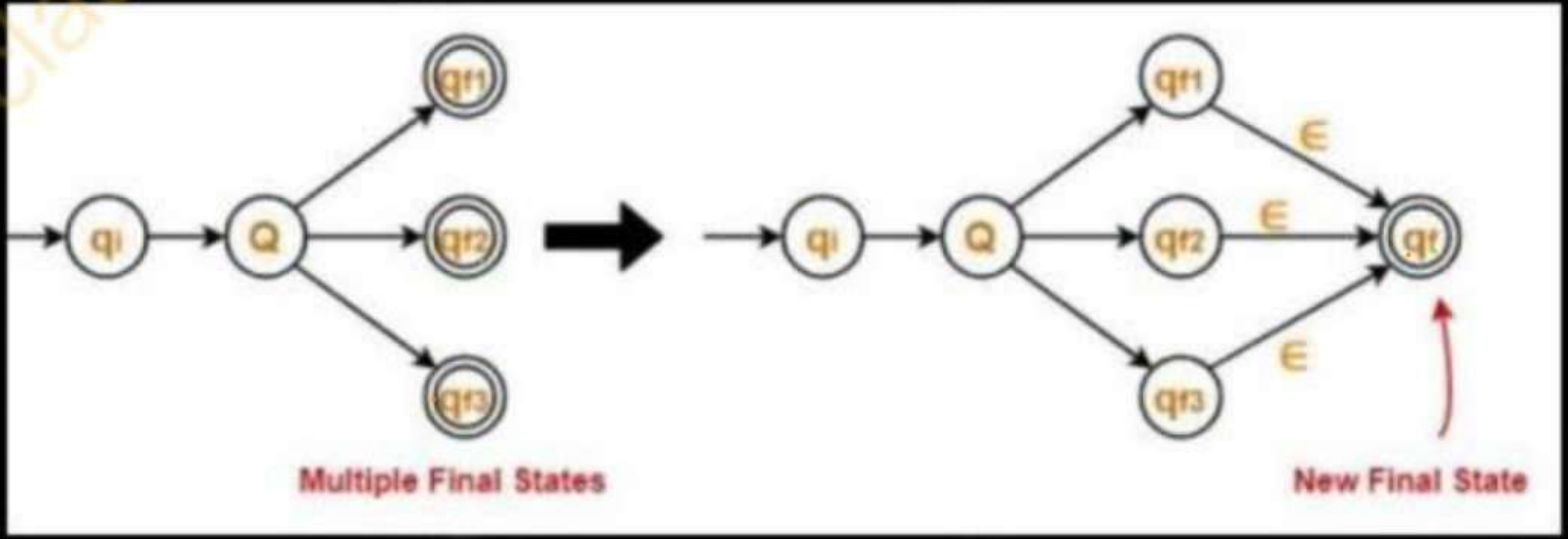
$= a^* a(b+aa)^* (\epsilon + a)$

State elimination method

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

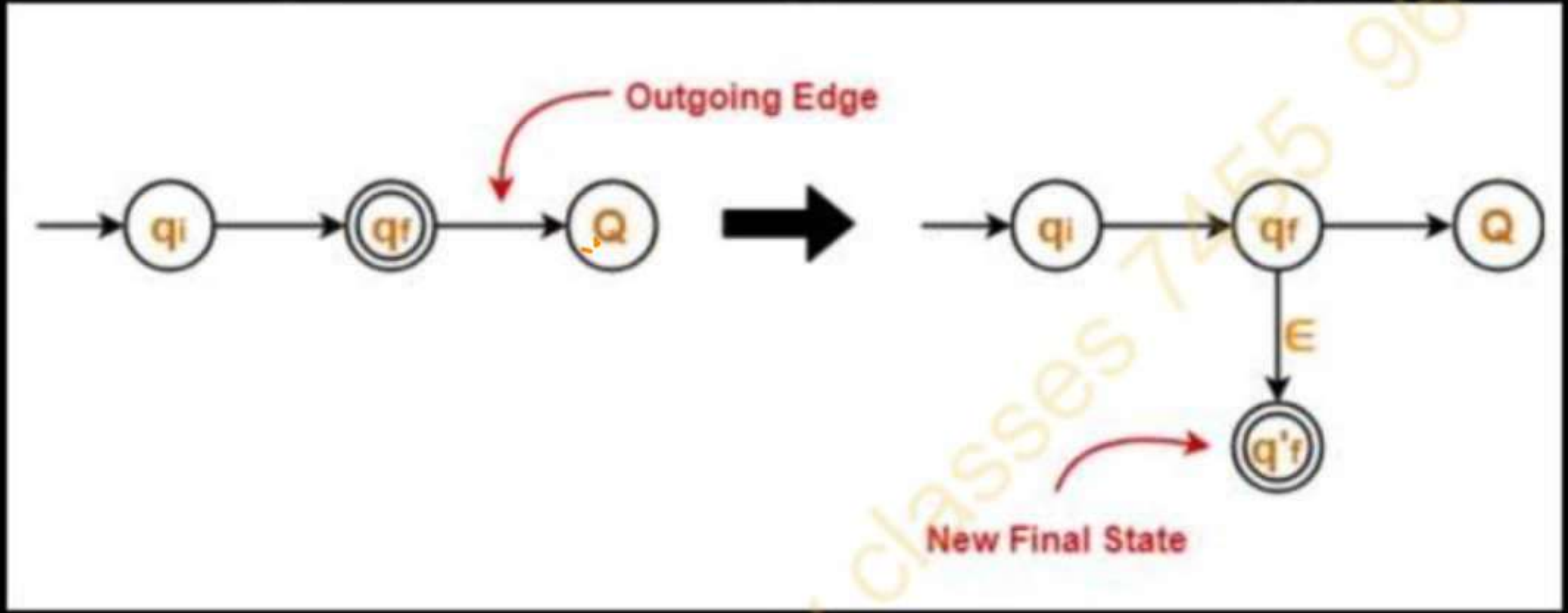


2. There must exist only one final state in the DFA.

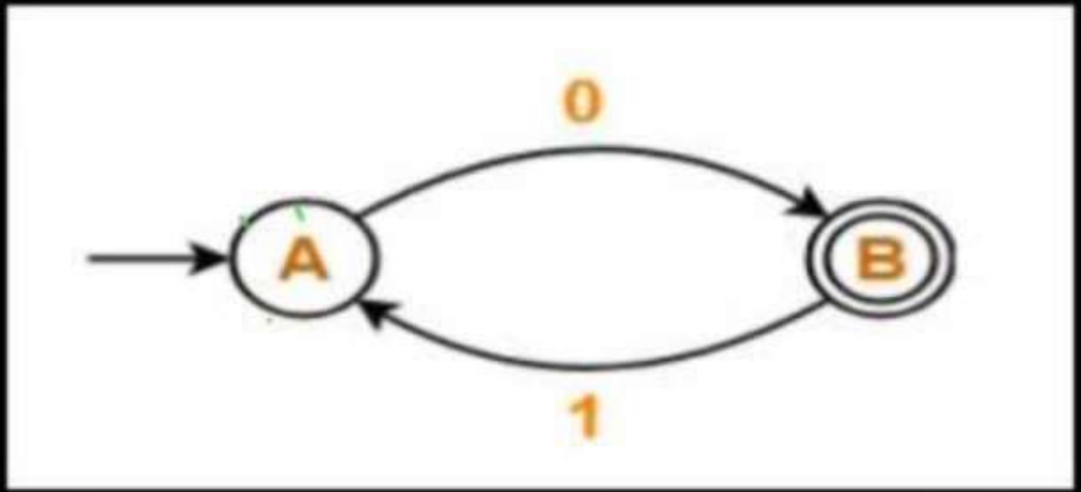


State elimination method

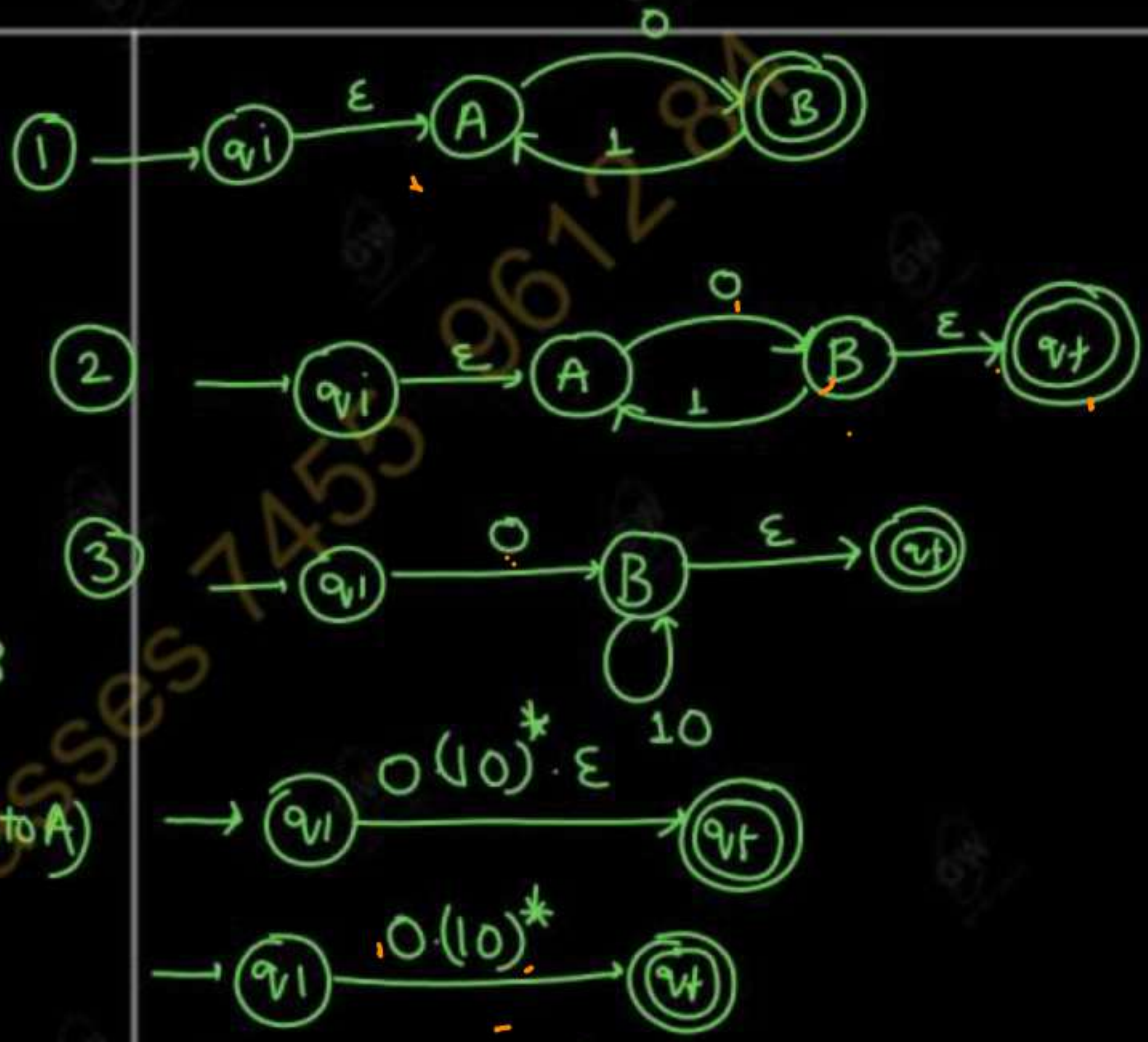
3. The final state of the DFA must not have any outgoing edge



EXAMPLE 1

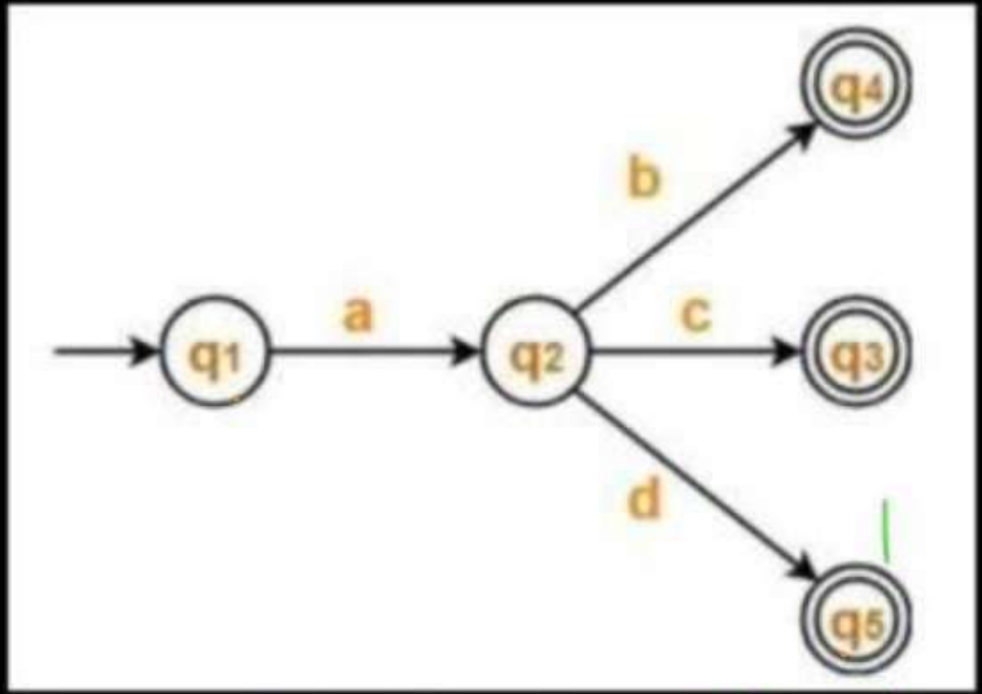


① there is an incoming (initial state) (A to B)
② Single final state ✓
③ there is an Outgoing edge from final state (B to A)

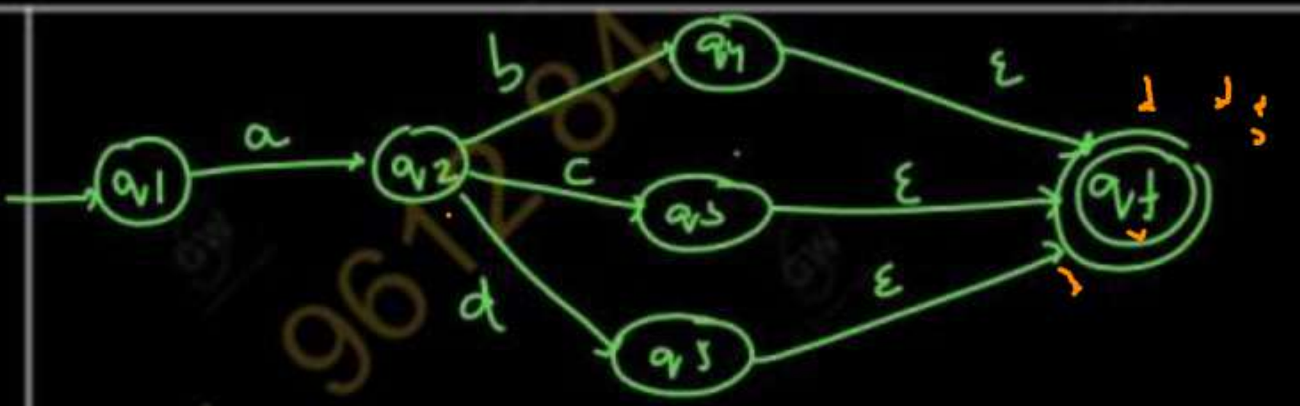


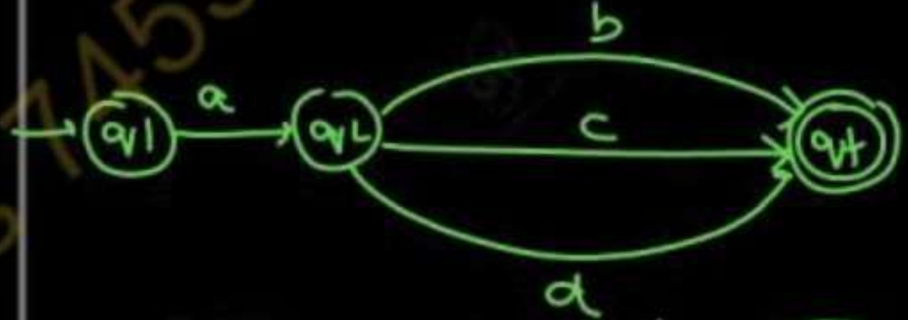
RE = 0·(10)*

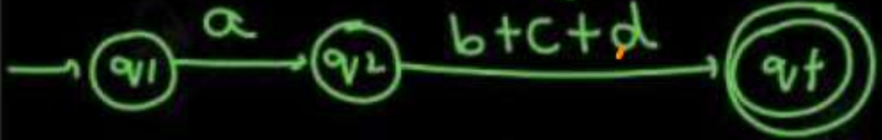
EXAMPLE 2

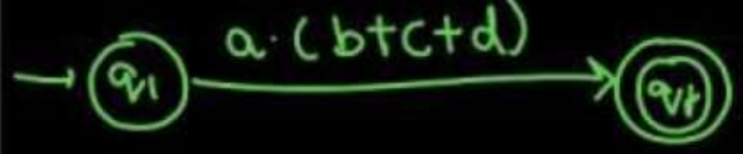


→ No incoming edge on intial stat. ✓
→ Multiple final stat.
→ No out going edge from finalstal



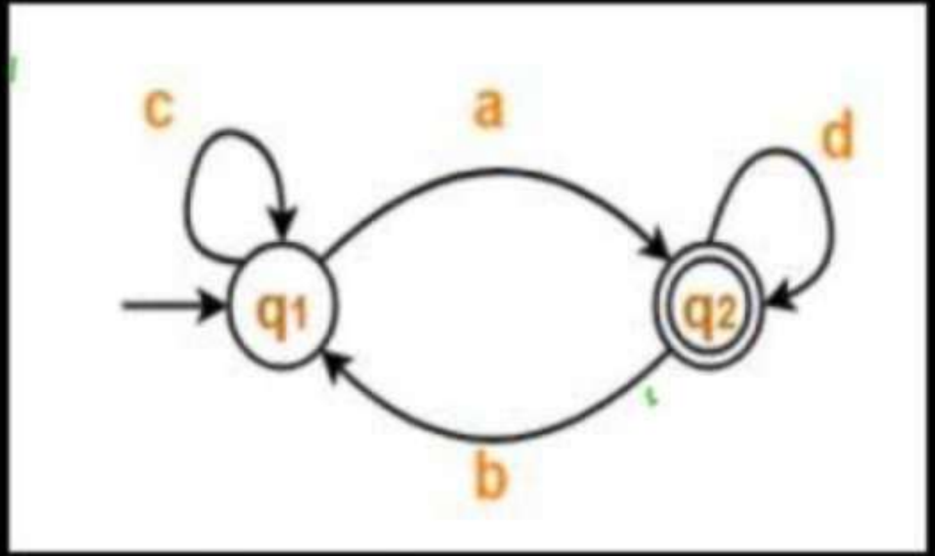




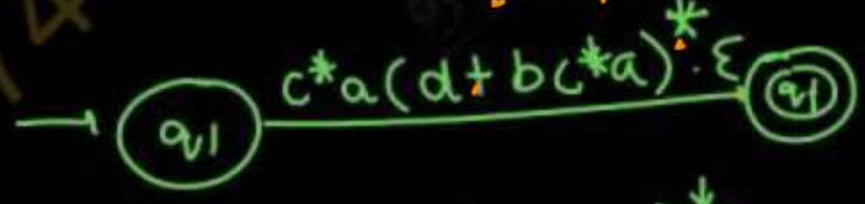


R.E. a.(b+c+d)

EXAMPLE 3

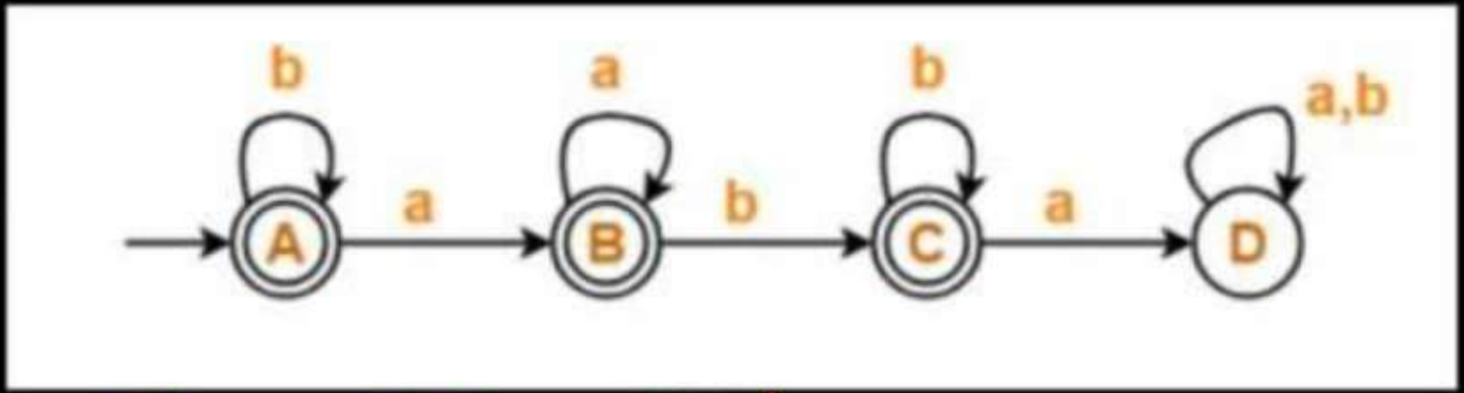


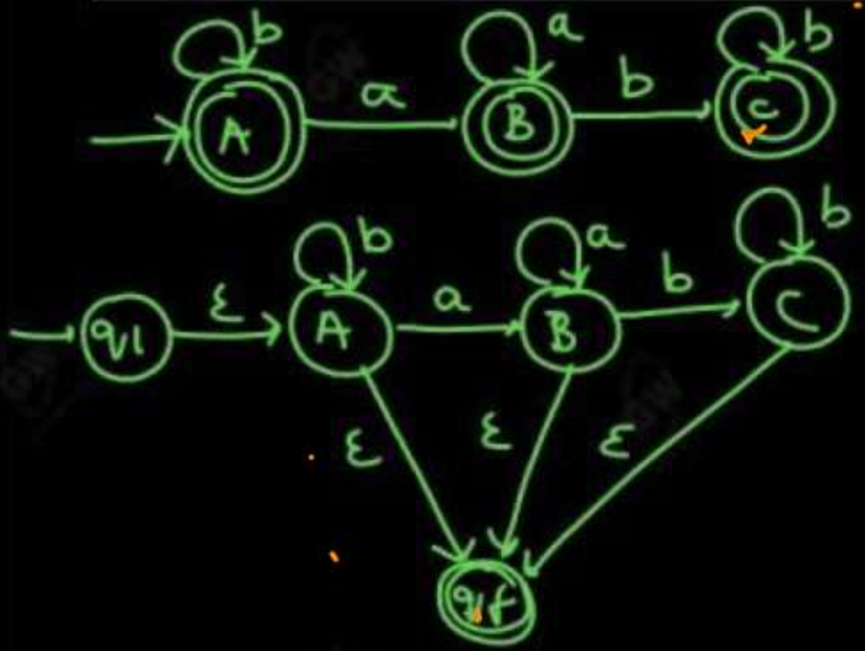
- Incoming in initial state
- Single final state
- Outing edge from final state



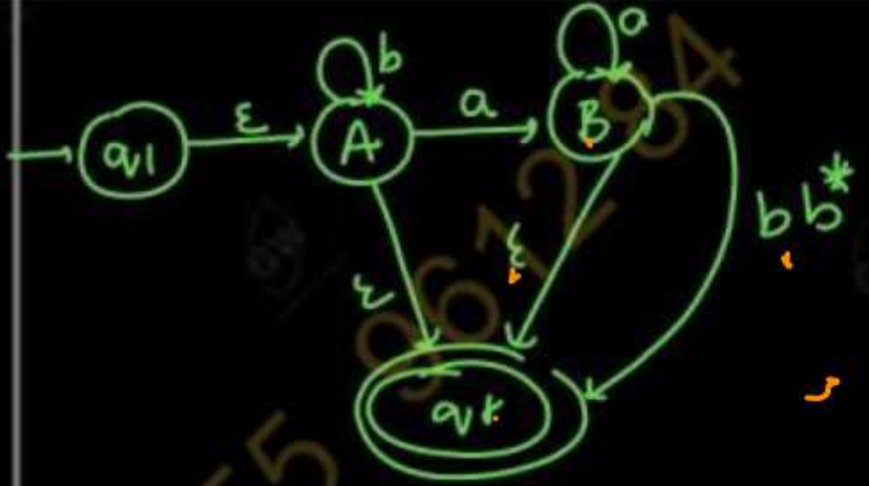
$c^*a(bc^*a+d)^*$

EXAMPLE 4

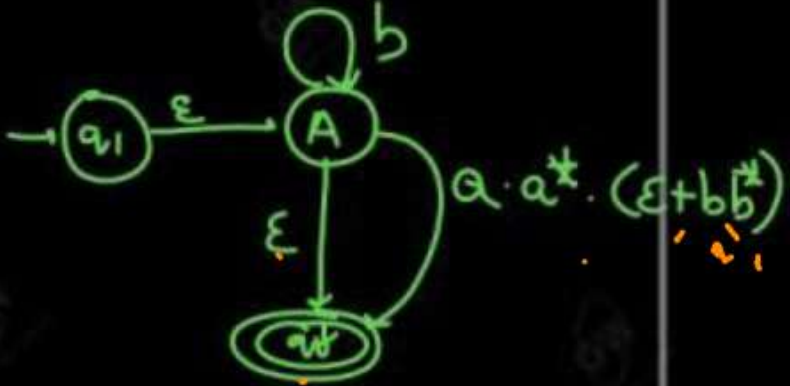


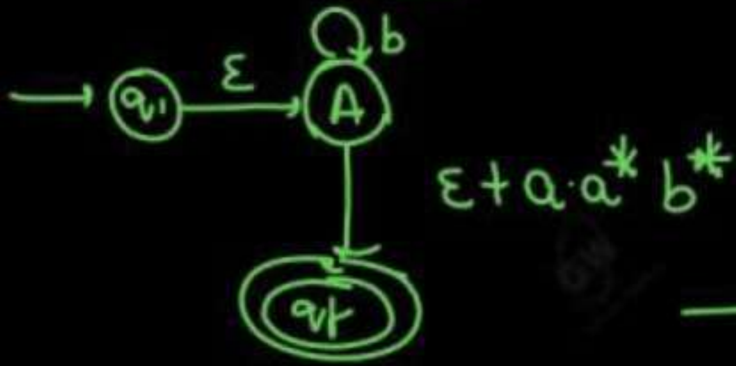


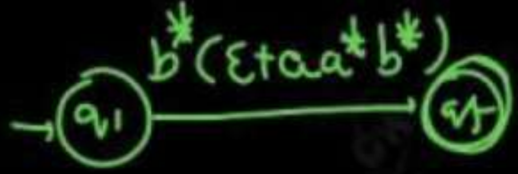
$bb^* \cdot \varepsilon = bb^*$



$\varepsilon + RR^* = R^*$

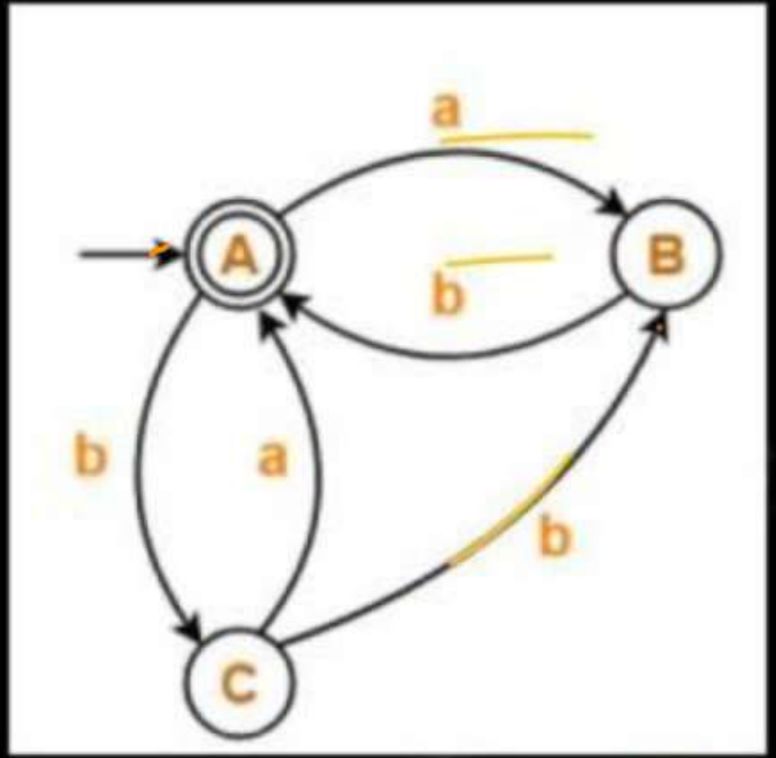


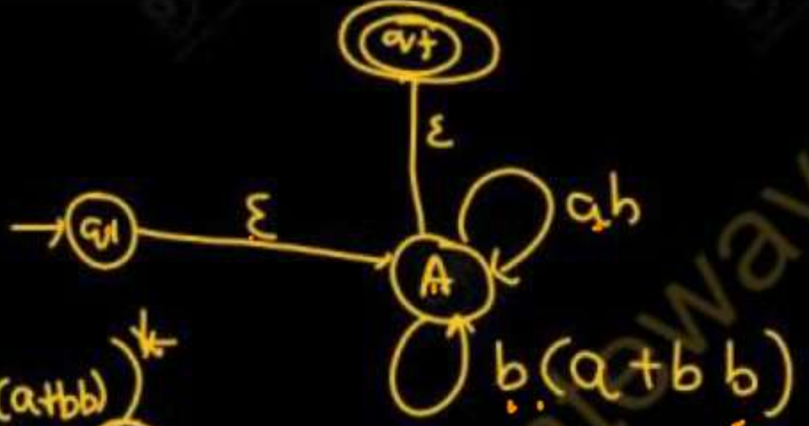


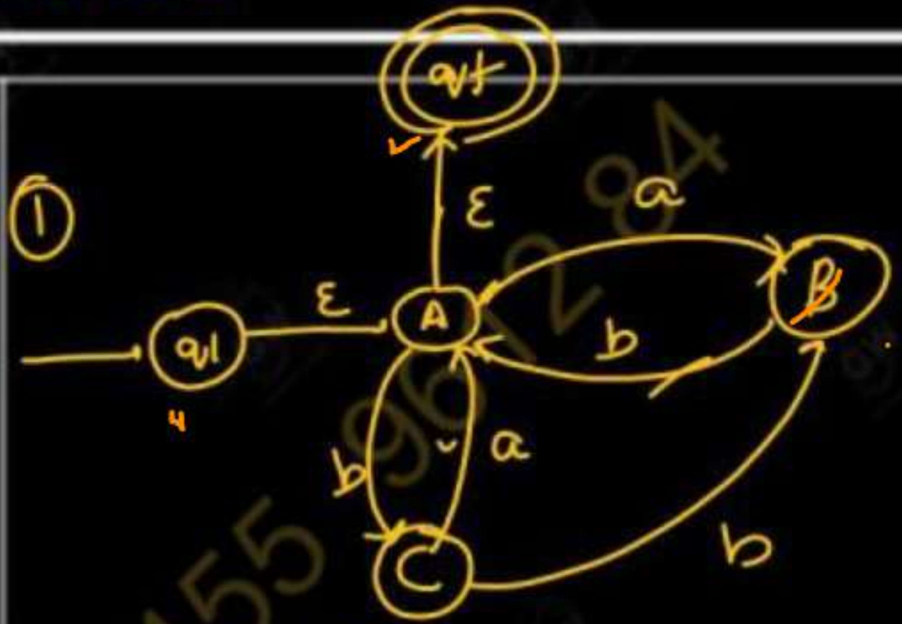


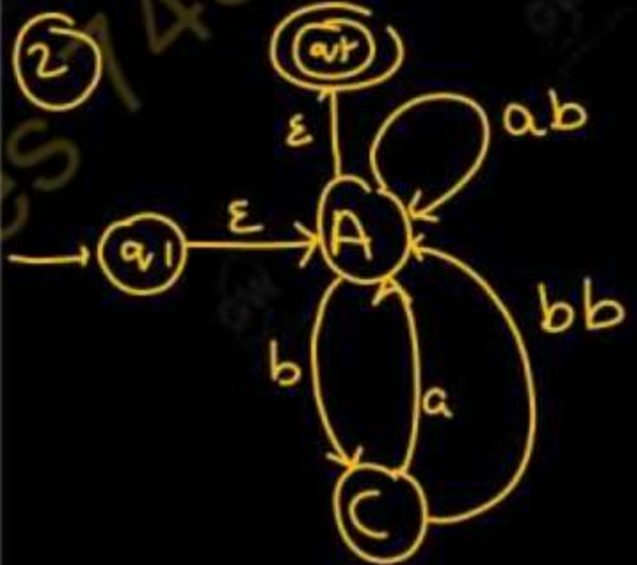
$RE = b^*(\varepsilon + aa^*b^*)$

EXAMPLE 5



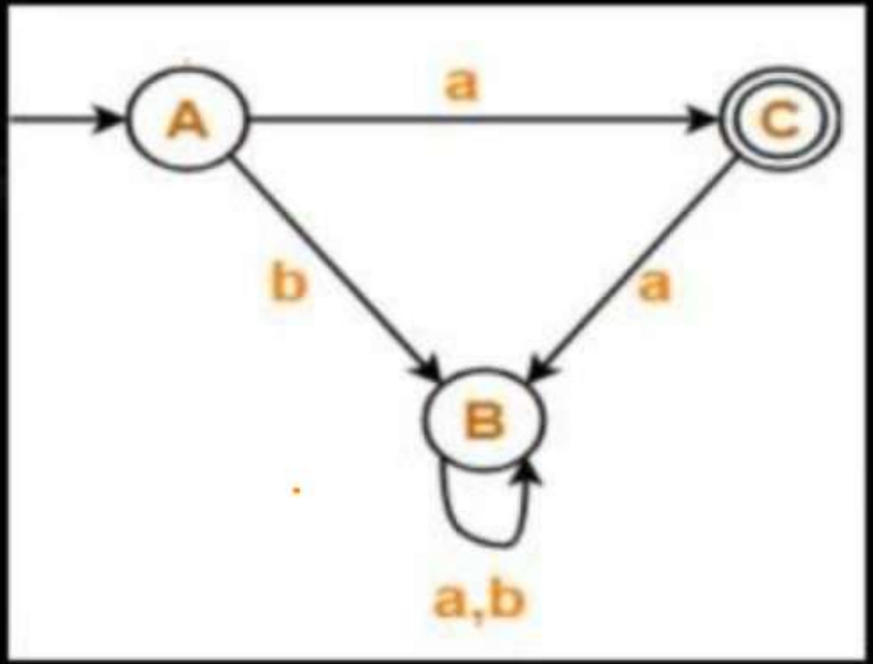






$$RE = (ab + b(a+bb))^*$$

EXAMPLE 6



R.E = a

Introduction

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

$L=\{a^nb^n \mid n>=1\}$ is not a regular langauge

Let L is Regular language

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

$P=4$

$|S| \geq P$

$S = aabb$

$|S| = 4$

$|S| \geq P$ $\quad |4| \geq 4$ – True

case 1

$S = xyz$

$\underset{x}{a}\,\underset{y}{a}\,\underset{z}{bb}$

$y = a$ $\quad |y|=1$ $\quad |y|>0$ $\quad 1>0$ T.

$xy = aa$ $\quad |xy|=2$ $\quad |xy| \leq P$ $\quad 2 \leq 4 = T$

$xy^iz \in L$ $\quad i \geq 0$

$a(a)^i(bb)$

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

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

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

$S: ab$

$|S| = 2$

$|S| \geq P$

$2 \geq 4$ – False

$S = aabb$

this not Regular language

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

Case II

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

① $|y| > 0$ $y = b$ $|y| = 1$

$1 > 0$ true

② $|xy| \leq p$ $xy = aab$ $|xy| = 3$

$3 \leq 4$ True

$S = \underbrace{aa}_{x}\underset{y}{(b)^i}\underbrace{b}_{z}$

$i=0$ $aa.\epsilon.b = aab \notin L$

$i=1$ $aabb \in L$

$i=2$ $aabbb \notin L$

Pumping Lemma For Regular Languages

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

let L is a Regular Language

L = { abb, aabbbb, aaabbbbbb . . . }

P = 6 (let)

S = abb

|S| = 3

|S| ≥ P

3 ≥ 6 – False

S = aabbbb

|S| = 6

|S| ≥ |

6 ≥ 6 T

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

① |y| > 0   y = bb

|y| = 2

2 > 0 T

|xy| ≤ P

② xy = aabbb

|xy| = |aabbb| = 5

5 ≤ P T

5 <= 6 T

S = aab (bb)^i b

i=0 aab ∈ b = aabb ∉ L

i=1 aabbbb ∈ L

i=2 aabbbbbb ∉ L

this not a Regular language

Case II

$S = \underset{x}{a}\underbrace{abb}_{y}\underbrace{bb}_{z}$

x = a |y| = 3

y = abb |xy| = 4

z = bb

① |y| > 0   3 > 0 T

② |xy| ≤ P   4 ≤ 6 T

③ S = xy^{i}z ∈ L i ≥ 0

S = a (abb)^i bb

i=0 a ∈ bb = abb ∉ L

i=1 aabbbb ∈ L

i=2 aabbabbbb ∉ L

from these two cases we proof this not a Regular language

Pumping Lemma For Regular Languages

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

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

let L is a regular language

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

$P = 3$

$S = aaaaa$

$|S| = 5$

$|S| \geq P$ $\quad 5 \geq 3$ true

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

$x = a$
$y = a$
$z = aaa$

① $|y| > 0$
$|a| > 0$
$1 > 0$ T

② $|xy| \leq P$
$|aa| \leq 3$
$2 \leq 3$ T

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

$S = a(a)^i aaa$

$i=0 \quad a\,\epsilon\,aaa = aaaa \notin L$

this not Regular

Case 2

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

$x = aa$
$y = a$
$z = aa$

① $|y| > 0$
$|a| > 0$
$1 > 0$ T

② $|xy| \leq P$
$|aaa| \leq 3$
$3 \leq 3$ T

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

$S: aa(a)^i aa$

$i=0 \quad aa\,\epsilon\,aa \quad aaaa \notin L$

$i=1 \quad aaaaa \in L$

$i=2 \quad aaaaaa \notin L$

this not a Regular One

Pumping Lemma For Regular Languages

L={a^n^2 | n>=0 } is not a regular language

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

$L = \{\epsilon, a^1, a^4 - a^9 - a^{16} - - \}$

let L is a Regular Language

$P = 3$

$S = aaaa$

$|S| \geq P$

$|aaaa| \geq 3$

$4 \geq 3$

true

$S = \underset{x}{a}\,\underset{y}{a}\,\underbrace{aa}_{z}$

$|y| > 0$ $|a| > 0$ $1 > 0$ T

$|xy| \leq P$ $|aa| \leq P$ $2 \leq 3$ T

$S = xy^iz \in L$ $i \geq 0$

$S = a(a)^i aa$

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

this not a regular Language

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

① $|y| > 0$ $|a| > 0$ $1 > 0$ true

② $|xy| \leq P$ $|aaa| \leq 3$ $3 \leq 3$ true

$S = xy^iz \in L$ $i \geq 0$

$aa(a)^i a$

$i=0$ $aa\epsilon a = aaa \notin L$

$i=1$ $aaaa \in L$

$i=2$ $aaaaa \notin L$

from these two case we confirmed that L is not a regular language

GW
GATEWAY CLASSES

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 (ε) = {ε})
- ϕ is a Regular Expression denoting an empty language. (L (ϕ) = { })
- 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) ∪ L(Y) where L(X+Y) = L(X) ∪ L(Y).
    - X . Y is a Regular Expression corresponding to the language L(X) . L(Y) where L(X.Y) = L(X) . L(Y)
    - R* is a Regular Expression corresponding to the language L(R*)where L(R*) = (L(R))*

$R = ab$

$R^* = (ab)^*$

If we apply any of the rules several times from 1 to 5, they are Regular Expressions.

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

Starting with a $\Sigma = a, b$

Pumping lemma

APPLICATION OF PUMPING LEMNA

- to prove that a particular language is non-regular

(7 Marks)

Prove the language l={a^n|n is a perfect cube} is not a regular language}

Let L is Regular Language

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

Let $P = 7$

$S = a$	$S = aaaa\ aaaa$
$\lvert S\rvert = 1$	$\lvert S\rvert = 8$
$\lvert S\rvert \geq P$	$\lvert S\rvert \geq P$
$1 \geq 7$	$8 \geq 7$
(False)	T

Case 1

$S = \underbrace{aa}_{x}\underbrace{aaa}_{Y}\underbrace{aaa}_{Z}$

1. $|Y| > 0$, $Y = aaa$, $|Y| = 3$, $3 > 0$ T
2. $|XY| \leq P$, $|aaaaa| \leq 7$, $5 \leq 7$ T
3. $S = xy^iz \in L \quad i \geq 0$

$aa(aaa)^i(aaa)$

$i=0 \quad aa\,\varepsilon\,aaa = a^5 \notin L$

this is not Regular lang.

Case 2

$S = \underbrace{aaaa}_{x}\ \underset{Y}{a}\ \underbrace{aaa}_{Z}$

1. $|Y| > 0$, $|a| > 0$, $1 > 0$ T
2. $|XY| \leq P$, $|aaaaa| \leq 7$, $5 \leq 7$ T
3. $S = xy^iz \in L \quad i \geq 0$

$aaaa(a)^i aaa$

$i=0 \quad aaaa\,\varepsilon\,aaa = a^7 \notin L$

$i=1 \quad aaaa\ a\ aaa = a^8 \in L$

$i=2 \quad aaaa\ aa\ aaa = a^9 \notin L$

this is not Regular Lang.

Case 3

$S = \underbrace{aaa}_{x}\ \underbrace{aaa}_{Y}\ \underbrace{aa}_{z}$

1. $|y| > 0$, $|aaa| > 0$, $3 > 0$ T
2. $|xy| \leq P$, $|aaaaaa| \leq 7$, $6 \leq 7$ T
3. $S = xy^iz \in L \quad i \geq 0$

$aaa(aaa)^i aa$

$i=0 \quad aaa\,\varepsilon\,aa = a^5 \notin L$

$i=1 \quad aaa\ aaa\ aa = a^8 \in L$

$i=2 \quad aaa\ aaaaaa\ aa = a^{11} \notin L$

Kleene 's theorem

- If a langauge 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 theorm is proved by proving the three following part-
- **KLEENE THEORM PART 1**

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

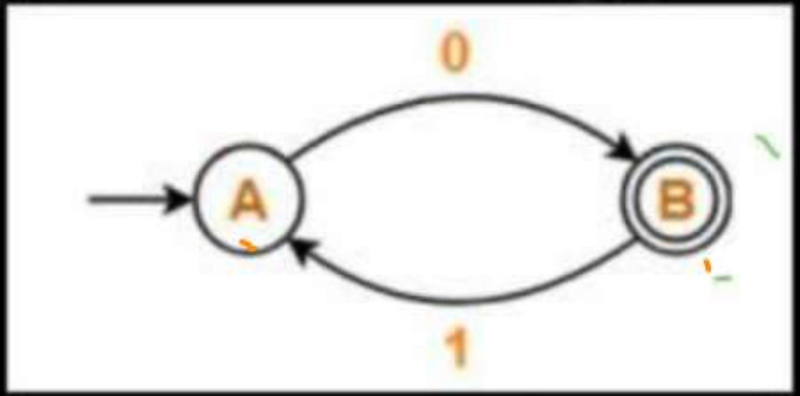
PROOF: every finite automata is considered to be the transition graph then nothing to proof

- **KLEENE THEORM PART 2**

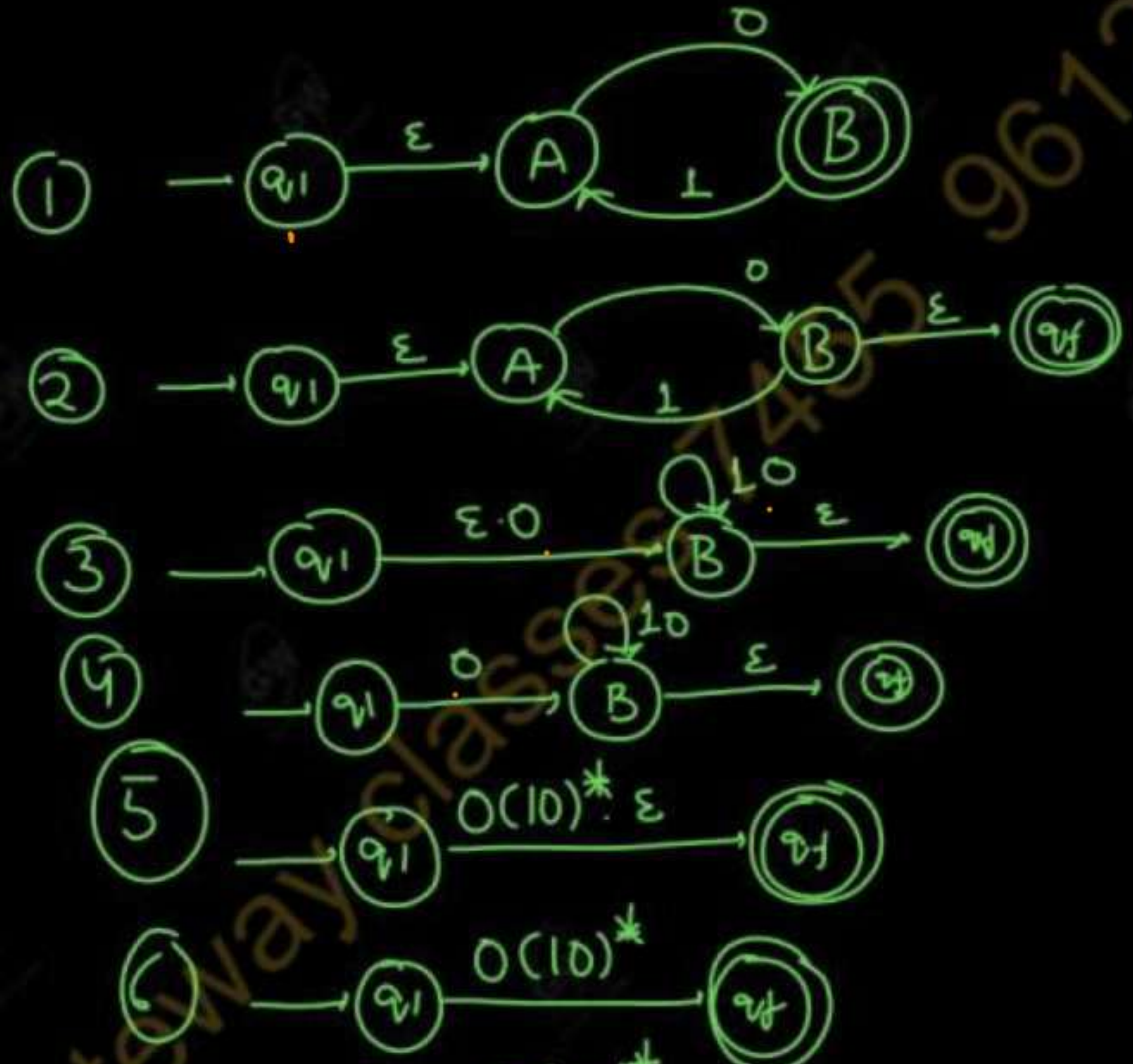
If a lanaguge is accepted by transition graph then it must be expressed by regular expression.

PROOF

Arden's theorem



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

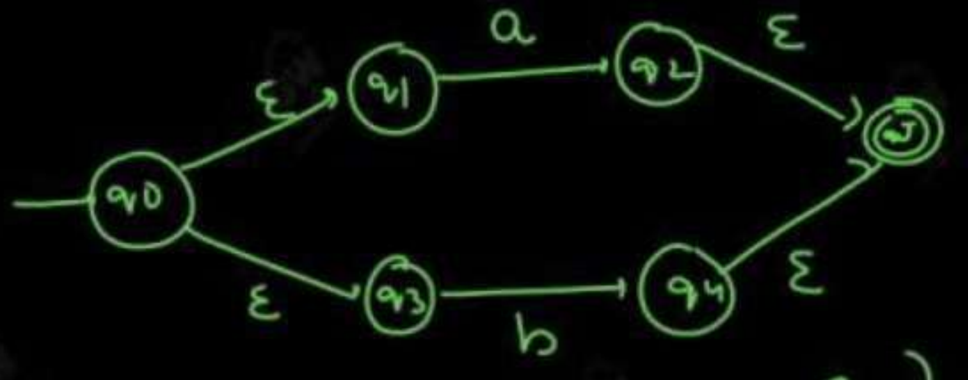


RE - $0(10)^*$

KLEENE THEORM PART 3

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

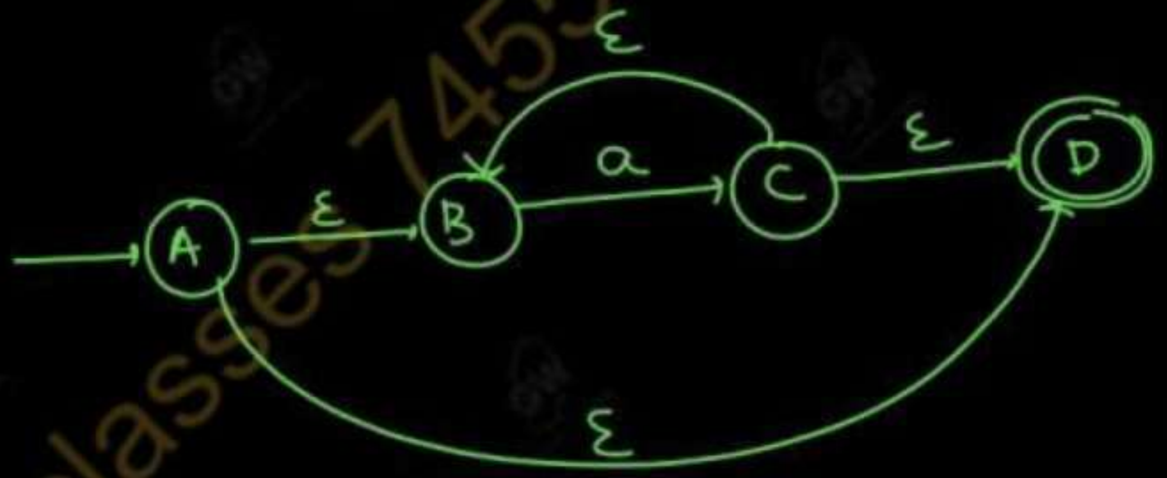
R.E = $a+b$



(Finite automata)

a^*

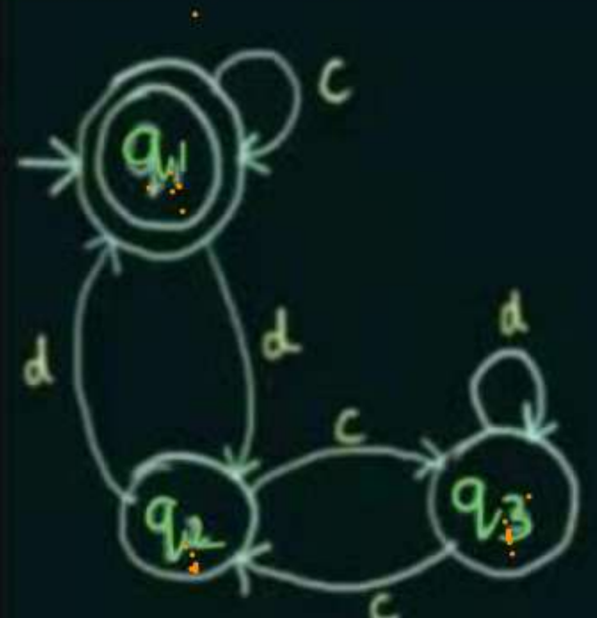
$L = \{\epsilon, a, aa, aaa ---- \}$



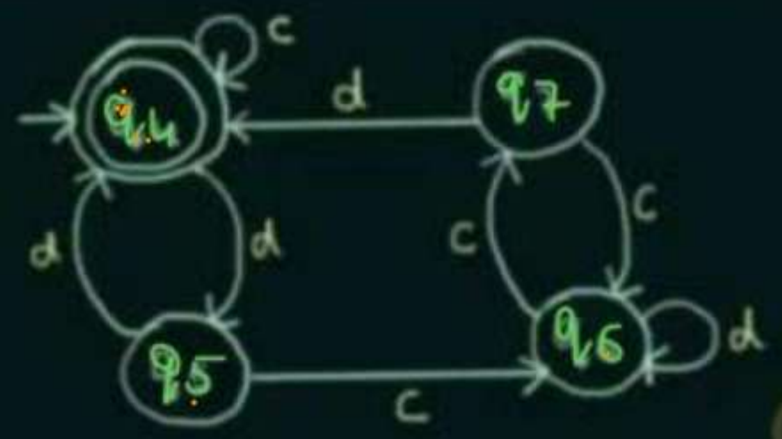
Equivalence of two finite automata

1. **For any pair of states { q_i, q_j} the transition for input a belong** Σ 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 inital state must be the final state for them to be equivalent

Equivalence of two finite automata



A



B

Intial state of automata A is q_1 & final state is q_1 $\Rightarrow$ cond^n ✓
Intial state of automata B is q_4 & final state is q_4

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

above two condition are satisfied So we Can say that two automata are equivalence to each other.

Equivalence of two finite automata

	c	d
{q1, q4}	{q1, q4} FS FS	{q2, q5} IS IS
{q2, q5}	q3, q7 IS IS	q1, q6 FS IS

→ this prove that A & B are equivalence.

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

A → q1 initial & final

B → q4 → initial & final

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 pPigeon is aasigned to each Pigeonhole then n-m pigeon are left without assigned to any Pigeon hole

Thank You