# UNIT-2

## T.A.F.L

ONE SHOT VIDEO

JOIN TELEGRAM FOR NOTE

SUBSCRIBE MULTIATOMS+

# REGULAR EXPRESSION

$\{+, \cdot, *\}$

⇒ An expression written using set of operator $(+, \cdot, *)$ and describing a regular language is known as regular expression.

⇒ +, Union operator :- The union of two language L and M denoted $L \cup M$ is the string, is a regular language (L or M, both)

For Ex : $L = \{001, 10\}$  $M = \{\epsilon, 001\}$
$L \cup M = \{001, 10, \epsilon, 001\}$

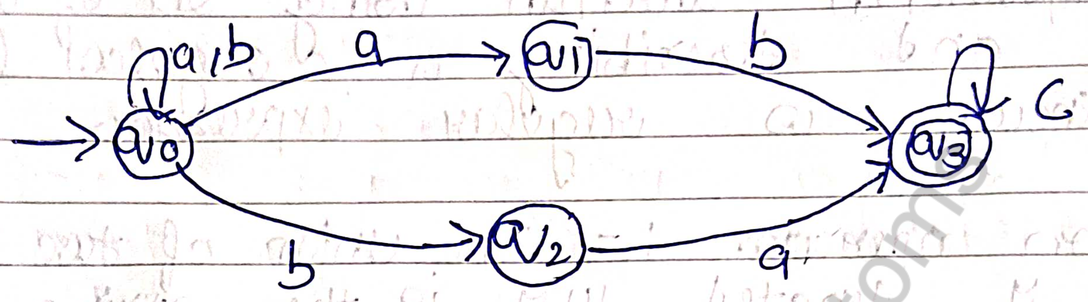
⇒ The concatenation of language L and M concatenation both this process is known concatenation

For ex : $L = \{001, 10\}$, $M = \{\epsilon, 001\}$
$L \cdot M = \{001, 10, 001001, 10001\}$

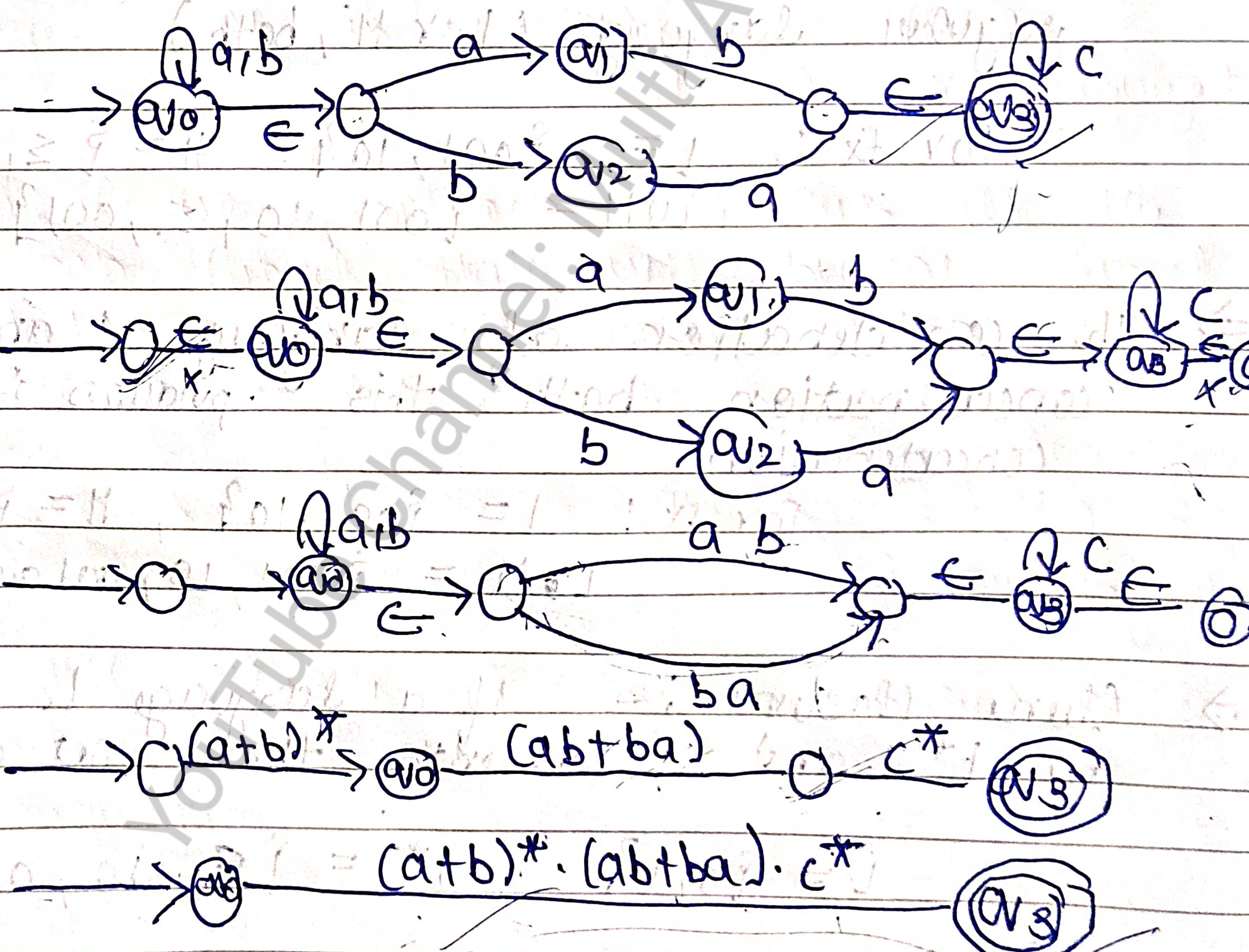
⇒ Cleene Closure :- If a language L is denoted by $L^*$ and represents the set of string

$L = \{0, 1\}$  $L^* = \{\epsilon, 00, 01, 011, 0011, 00011\}$

Q. Find the regular expression corresponding to the finite automata given below




Sol




$(a+b)^* (ab+ba) \cdot c^*$

Ans

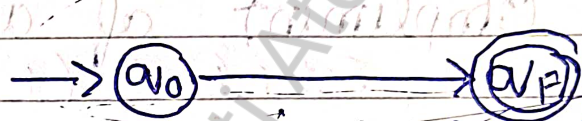
# CLOSURE PROPERTIES OF REGULAR LANG

* The union of two regular lang is regular
* The intersection of two regular lang^e is regular
* The compliment of a regular lang^e is regular
* The difference of two regular language is regular.
* The reversal of two regular lang^e is regular.
* The closure of a regular lang^e is regular
* The concatenation of regular lang^e is regular
* A homomorphism of regular language is regular.
* The inverse homomorphism of a regular language is regular

**Q.** Prove that compliment, Homomorphism, Inverse Homomorphism and closure of a Regular language is also regular?

**Sol** <u>Compliment</u> :-

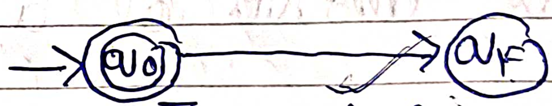
(i) Since L is a regular language so there exist a finite automata (FA) for it:

M =



M = FA for language L

Now if we compliment M, then all the non-final state will become final states and all final states will become non-final state.

F → N.F
N.F → F

$\overline{M}$ =



$\overline{M}$ = FA for lang$^c$ L.

So still it is a FA (finite automata) so the complement of L(i) is also regular language.

Homorphism :-

$$h : \Sigma \longrightarrow \Gamma^*$$

Properties:

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

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

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

$h(a) = 010$

$h(b) = 102.$

Now we have regular language L that have regular expression

$$L = aba^*$$

Now find homorphism of regular expression

$$h(aba^*) = h(a) \cdot h(b) \cdot h(a)^*$$
$$= 010 \cdot 012 \cdot (010)^*$$

Now, we saw that $aba^*$ is regular expression and it is regular language. the homorphism of regular expression is also produce a regular expression that why it is called regular language

⇒ **Inverse Homorphism**

let

$\Sigma = \{0,1,2\}$ $\quad\quad$ $\Gamma = \{a,b\}$

$h(0) = a$ $\quad\quad$ $h^{-1}(a) = 0$

$h(1) = ab$ $\quad\quad$ $h^{-1}(ab) = 1$

$h(2) = ba$ $\quad\quad$ $h^{-1}(ba) = 2$

Suppose we have given language $L = \{ababa\}$

Now find $h^{-1}(L) = \{022, 110, 102\}$

$ababa = 022$

$ababa = 110$

$ababa = 102$

L have regular expression, Now that you find $h^{-1}(L)$ is also have regular expression due to which $h^{-1}(L)$ is also a regular language.

⇒ **Closure (Star)**

⇒ let we have language L and for that language we have regular expression

$L \rightarrow$ Regular Exp. $\quad L^* \rightarrow$ r.e

Now if you find closure of regular expression it is also formed that.

∴ It is also a regular language.

# ARDEN THEOREM

⇒ Arden theorem state that if P and Q are two regular expression over Σ and if P does not contain ε then following in R given by $[R = Q + RP]$ has unique solution $[R = QP^*]$

⇒ Proof

Let us consider the equation $R = Q + RP$ —— ①
puth the value of R in eq ①

$$R = Q + (Q + RP)P$$
$$R = Q + QP + RP^2$$

We put the value of R again and again and we get the following eqn

$$R = Q + QP + QP^2 + QP^3 + \cdots$$
$$R = Q(\varepsilon + P + P^2 + P^3 + \cdots)$$

Since $(\varepsilon + a + a^2 + \cdots = a^*)$

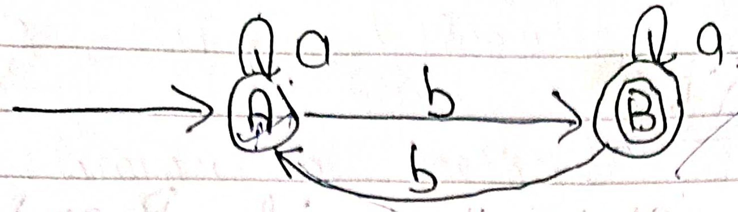
$$[R = QP^*]$$ Hence proved

Application :-

⊙ It helps to determine the regular expression of finite automata.

⊙ Arden theorem helps in checking the equivalence of two regular language

Q



Sol

$$A = \varepsilon + A \cdot a + B \cdot b \quad \text{--- (1)}$$

$$B = A \cdot b + B \cdot a \quad \text{--- (2)}$$

$$R = Q + RP$$

$$B = A \cdot b + Ba$$

$$R = QP^*$$

$$B = Aba^* \quad \text{--- (3)}$$

$$A = \varepsilon + A \cdot a + (Aba)^* b$$

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

$$R = Q + RP$$

$$[R = QP]^*$$

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

$$A = (a + ba^* b)^* \quad \text{--- (4)}$$

put in (iii)

$$\boxed{B = (a + ba^* b)^* ba^*}$$

Ans

# KLEENE THEOREM

=> Any Regular language can be accepted by finite automata.

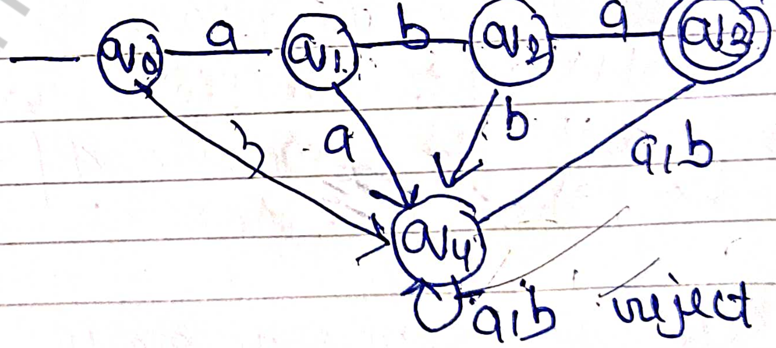
=> Kleene theorem has three parts :-

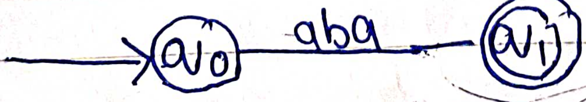
1) If Language can be accepted by FA then it can also be accepted by TG (Transition Graph.

2) If language can accepted by TG then it can also be expressed by RE (Regular Expression)

3) If any regular expressed by R.E then it can be accepted by finite automata.

Part 1 Proof   let s = aba   $\Sigma$ {a,b}

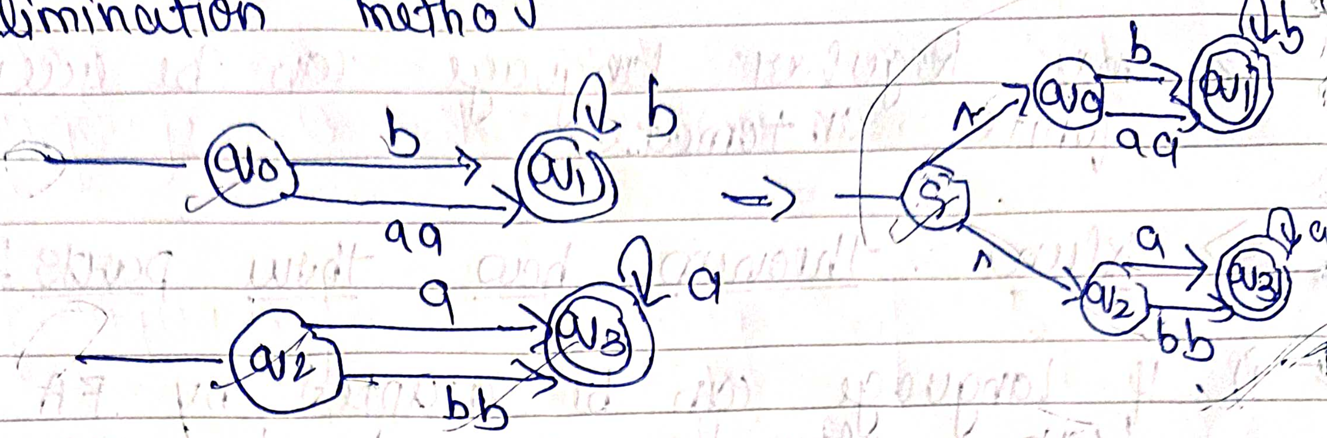



Transition Graph Proof

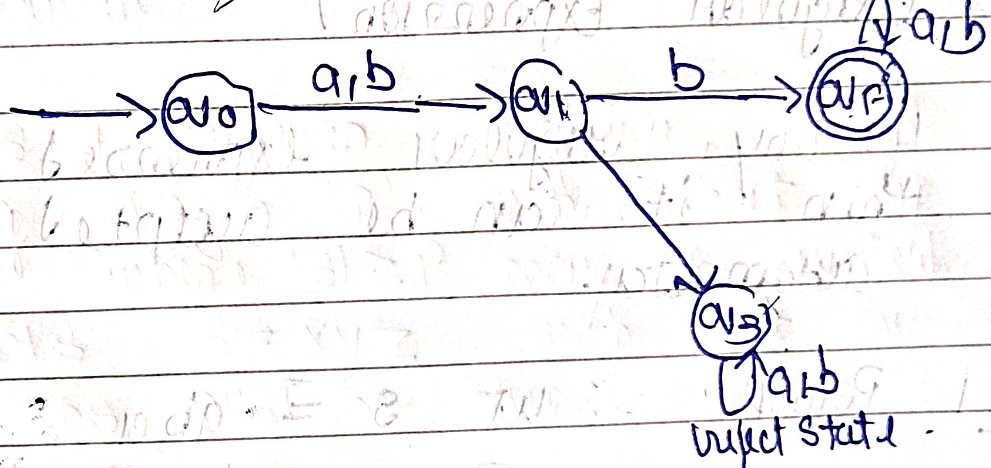
## Part 2 Proof (Transition Graph to Regular Exp)

- Elimination method




## Part3 Proof ( R.E to F.A )

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



# DECISION PROPERTIES

Approximately all the properties are decidable in case a finite automation. Here we will use machine model to proof decision properties.

(i) Emptiness :- To check whether a given FA accepts empty language or not.
→ If the resulting machine is free from final state then finite automata accept empty language otherwise non-empty

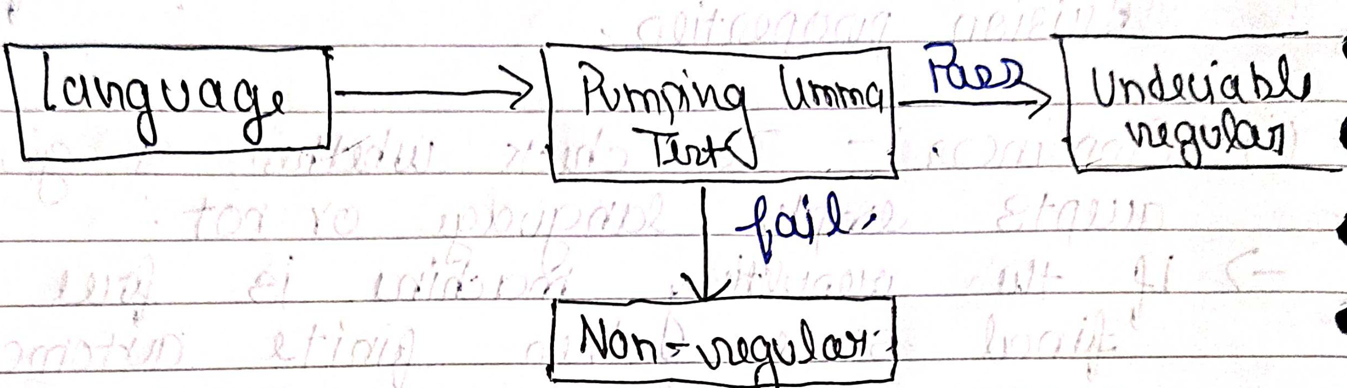
(ii) Finiteness :- It state that the language accepted by given FA is finite or not.
→ If the resulting machine do not contain loop or cycles then the finite automata accept finite language otherwise infinite lanaguage.

(iii) Membership :- Membership is a property to verify an arbitary string is accepted by a finite automata or not i.e it is a member of language or not.

(iv) Equality :- Two finite automata over $\Sigma$ are equivalent if they accept the same se of string over $\Sigma$.

# PUMPING LEMMA

- Pumping lemma is used to prove that a language is NOT REGULAR.




⇒ It cannot be used to prove that a language is regular.

⇒ 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 3 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) $|xy| \leq p$

Q Using Pumping lemma prove that the language $A = \{a^n b^n \mid n \geq 0\}$ is not Regular

Proof

Assume that A is regular

Pumping length $= P$

$S = a^P b^P$

$x \; y \; z$

let $P = 7$

$S = aaaaaaabbbbbbb$

case 1 : The y is in the 'a' part , $i=2$ $|xy| \leq P$

$\underbrace{aa}_{x}\underbrace{aaaaa}_{y}\underbrace{bbbbbbb}_{z}$ | $xy^iz = xy^2z$ ✗

aa aaaa aaaa abbbbbbb

$11 \neq 7$

case 2 : The y is in the 'b' part

$\underbrace{aaaaaaa}_{x}\underbrace{bbbbbb}_{y}\underbrace{b}_{z}$ | $xy^iz = xy^2z$ ✗

aaaaaaabbbbbbbbbbb

$7 \neq 11$

case 3 : The y is the 'a' and 'B' part

$\underbrace{aaaaa}_{x}\underbrace{aabb}_{y}\underbrace{bbbbb}_{z}$ | $xy^iz = xy^2z$ ✗

aaaaaaabbaabbbbbbb

$a^nb^n$ ✗

- $|xy| \leq P$ , false
- $|y| > 0$ , true.
- Hence It is not a regular language

UNIT- 2 COMPLETE THANK YOU