CS & IT ENGINEERING

Theory of Computation

Finite Automata

Lecture No. 22

By- DEVA Sir

01 Closure properties for Regular Languages

02

03

04

05

Closure properties for Regular languages:

1. Union ✓
2. Intersection ✓
3. Complement ✓
4. Difference ✓
5. Concatenation ✓
6. Reversal ✓
7. Kleene star ✓
8. Kleene plus ✓
9. Symmetric Difference ✓
10. Subset ✗
11. Prefix ✓
12. Suffix ✓
13. Substring ✓
14. Substitution(L) = f(L) ✓
15. Homomorphism(L) = h(L) ✓
16. ε-free Homomorphism ✓
17. Inverse Homomorphism ✓
18. Quotient ✓
19. $\frac{1}{2}(L)$ = Half(L) ✓
20. Second half(L) ✓
21. one-third(L) = $\frac{1}{3}(L)$ ✓
22. Middle $\frac{1}{3}(L)$ ✓
23. Last $\frac{1}{3}(L)$ ✓
24. Finite U ✓
25. Finite ∩ ✓
26. Finite − ✓
27. Finite • ✓
28. Finite ⊆ ✓
29. Finite f ✓
30. Inf U ✗
31. Inf ∩ ✗
32. Inf − ✗
33. Inf • ✗
34. Inf ⊆ ✗
35. Inf f ✗

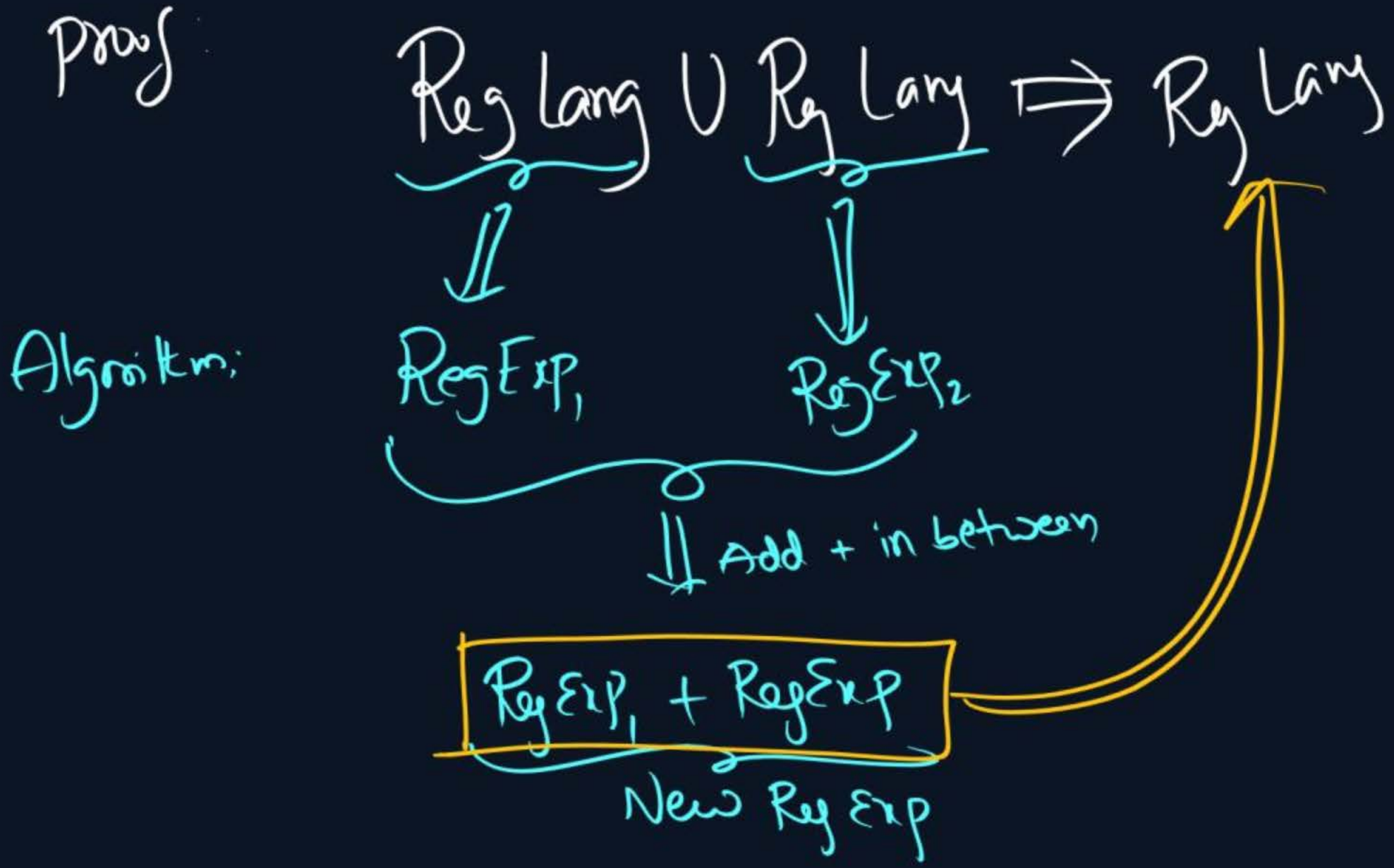

R^X
R_SI_All

① Union for Regular Languages

→ closed

$Reg_1 \cup Reg_2 \Longrightarrow$ Always Regular

Proof:


Reg Lang U Reg Lang ⇒ Reg Lang
Algorithm:
RegExp1
RegExp2
Add + in between
RegExp1 + RegExp2
New Reg Exp

I) $L_1 = \phi$, $L_2 = \Sigma^*$ $\Rightarrow$ $L_1 \cup L_2 = \Sigma^* = L_2$

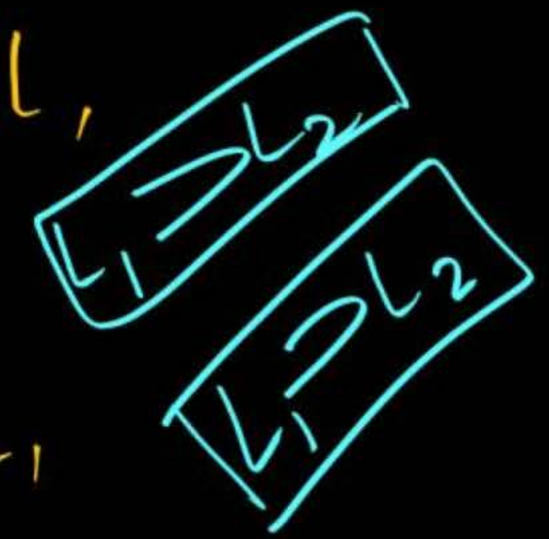
II) $L_1 = a^*$, $L_2 = (aa)^*$ $\Rightarrow$ $L_1 \cup L_2 = a^* = L_1$

III) $L_1 = Any$, $L_2 = \phi$ $\Rightarrow$ $L_1 \cup L_2 = L_1$

IV) $L_1 = \Sigma^*$, $L_2 = Any$ $\Rightarrow$ $L_1 \cup L_2 = \Sigma^* = L_1$

V) $L_1 = a^* b^*$, $L_2 = a^*$ $\Rightarrow$ $L_1 \cup L_2 = L_1$

VI) $L_1 = a(a+b)^*$, $L_2 = a^+ b^+$ $\Rightarrow$ $L_1 \cup L_2 = L_1$



H.W

ϕ

Σ^*

$a^n b^n$

$\overline{a^n b^n}$

I) Reg Lang$_1$ $\cup$ Reg Lang$_2$ $\Rightarrow$ Regular

II) Reg Lang $\cup$ Non Reg lang $\Rightarrow$ either Reg or Not reg

III) Non Reg Lang$_1$ $\cup$ Non reg lang$_2$ $\Rightarrow$ either Reg or Not reg

Reg ∪ Non Reg ⟹ Either Reg or Non-Reg

Case 1: Σ^* ∪ Any Non reg ⟹ Regular

Case 2: ϕ ∪ Any Non reg ⟹ Not reg

Non Reg ∪ Non Reg ⟹ Either Reg or Not Reg

Case 1: $a^n b^n \cup \overline{a^n b^n} \Rightarrow \underbrace{(a+b)^*}_{\text{Reg}}$

Case 2: $a^n b^n \cup a^n b^n \Rightarrow \underbrace{a^n b^n}_{\text{Not Reg}}$

$$\overline{a^n b^n} = (a+b)^* - \{a^n b^n\}$$

$$= \boxed{\{a^m b^n \mid m \neq n\} \cup (a+b)^* ba (a+b)^*}$$

$$a^* b^* \Rightarrow a^n b^n$$

$$a^m b^n \mid m \neq n$$

$$\bar{L} = \Sigma^* - L$$

Maths: $\overline{m=n} \implies m \neq n$

Language: $\overline{\{a^m b^n \mid m=n\}} \implies \{a^m b^n \mid m \neq n\} \cup \{X\,ba\,X\}$

$$\bar{L} = \overline{\{\varepsilon, ab, a^2b^2, \ldots\}}$$

$$\bar{L} = \Sigma^* - \{\varepsilon, ab, a^2b^2, \ldots\}$$

$$= \{a, b, ba, aab, baa, \ldots\}$$

$$= \{a^m b^n \mid m \neq n\} \cup \{ba\}$$

A) Regular

B) Not Regular

C) Either Reg or Not Reg

D) None

IV) If $L_1 \cup L_2$ is Regular then L_1 is Either Reg or Not Reg

Reg $\cup$ L_2 $\Rightarrow$ Regular possible

NonReg $\cup$ L_2 $\Rightarrow$ Regular possible

V) If $L_1 \cup L_2$ is Not Reg then L_1 is Either Reg or Not Reg

$\rightarrow$ Reg$^{\phi}$ $\cup$ L_2^{notreg} $\Rightarrow$ Not reg possible

Not Reg $\cup$ L_2 $\Rightarrow$ Not Reg possible

(Not Reg: q_1, q_2; L_2: q_1, q_2)

② Intersection for regular languages

↳ Closed

$Reg_1 \cap Reg_2 \Rightarrow$ Always Regular

i) $\phi \cap L \Rightarrow \{\} \cap L \Rightarrow \{\} \Rightarrow \phi$

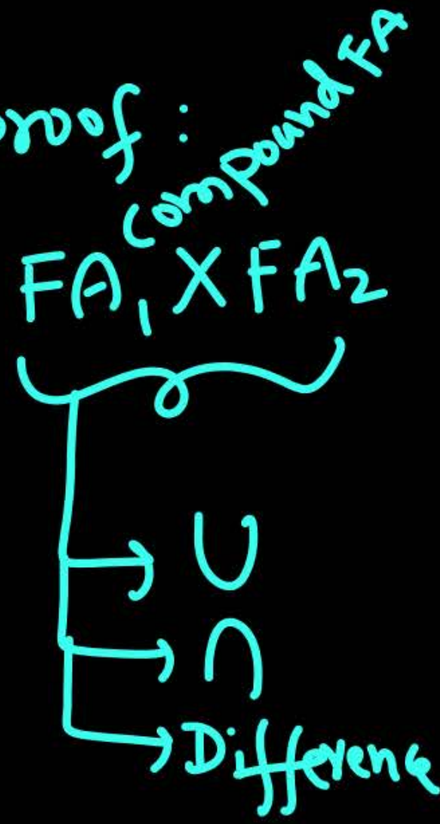
ii) $\Sigma^* \cap L \Rightarrow$ Set of all strings $\cap L \Rightarrow L$

iii) $a^* \cap b^* \Rightarrow \{\varepsilon, a, a^2, \cdots\} \cap \{\varepsilon, b, b^2, \cdots\} \Rightarrow \{\varepsilon\}$

iv) $a^+ \cap b^+ \Rightarrow \phi$

v) $a^*b^* \cap a^* \Rightarrow a^*$

Proof:



I) If L_1 is Reg and L_2 is Reg then $L_1 \cap L_2$ is Regular

II) If L_1 is Reg and L_2 is Not Reg then $L_1 \cap L_2$ is Either Reg or Not Reg

case 1: $\phi \cap L_2 \Rightarrow$ Reg

case 2: $\Sigma^* \cap L_2 \Rightarrow$ Not reg

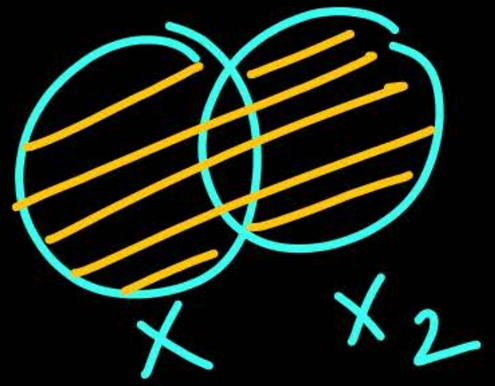
III) If L_1 is Not reg and L_2 is Not Reg then $L_1 \cap L_2$ is Either Reg or Not Reg

case 1: $\{a^n b^n\} \cap \{b^n a^n\} \Rightarrow \{\varepsilon\}$ reg

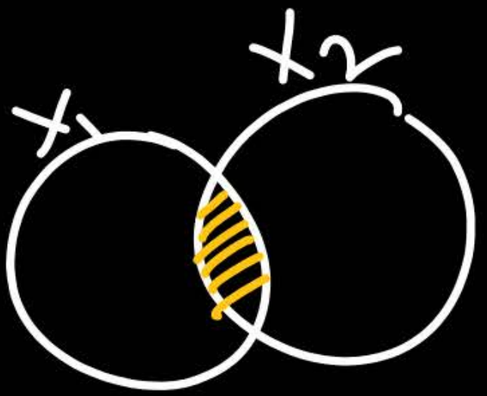
case 2: $\{a^n b^n\} \cap \{a^n b^n\} \Rightarrow$ Not reg

IV) If $L_1 \cap L_2$ is Reg then L_1 is either reg or not reg

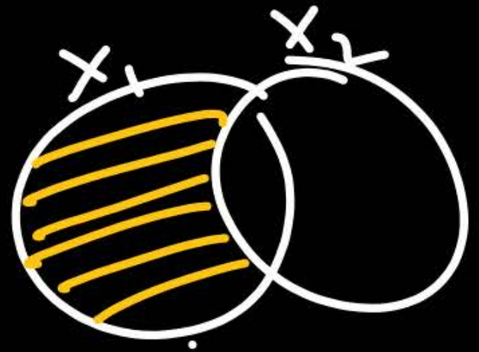
V) If $L_1 \cap L_2$ is Not reg then L_1 is either reg or not reg



$$X_1 \cup X_2 = \{x \mid x \in X_1 \text{ or } x \in X_2\}$$



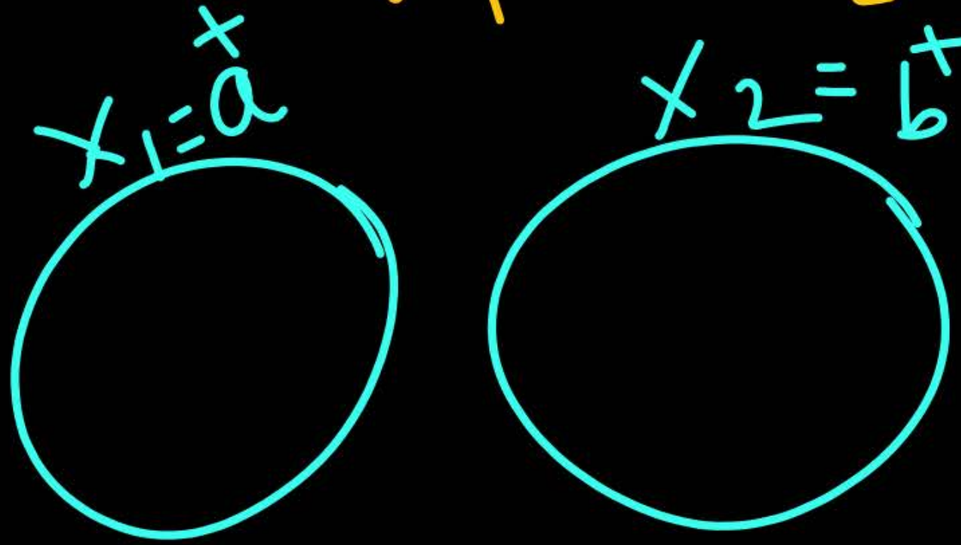
$$X_1 \cap X_2 = \{x \mid x \in X_1 \text{ and } x \in X_2\}$$



$$X_1 - X_2 = \{x \mid \underbrace{x \in X_1 \text{ and } x \notin X_2}_{x \text{ is only in } X_1}\}$$

If $X_1 \cap X_2 = \phi$ then

X_1 and X_2 are Disjoint Sets



Complement for Regular Languages :

↳ closed

$\overline{Reg}$ ⇒ Always Regular

I) L is Reg iff $\bar{L}$ is Reg

II) L is not reg iff $\bar{L}$ is not reg

III) $\overline{\text{Not Reg}}$ ⇒ Not Reg.

Proof :

L is Regular

⇓ ⇑

Design DFA

⇓ ⇑ finals ↔ non finals

modified DFA

⇓ ⇑

$\bar{L}$ is Regular

i) $L = \phi \Rightarrow \bar{L} = \Sigma^*$

ii) $L = \Sigma^* \Rightarrow \bar{L} = \phi$

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

iii) $L = a\Sigma^* \Rightarrow \bar{L} = \varepsilon + b\Sigma^*$

iv) $L = b\Sigma^* \Rightarrow \bar{L} = \varepsilon + a\Sigma^*$

v) $L = \Sigma^* a \Rightarrow \bar{L} = \varepsilon + \Sigma^* b$

vi) $L = \Sigma^* a \Sigma^* \Rightarrow \bar{L} = b^*$

vii) $L = \Sigma^* b \Sigma^* \Rightarrow \bar{L} = a^* = \varepsilon + a^+$

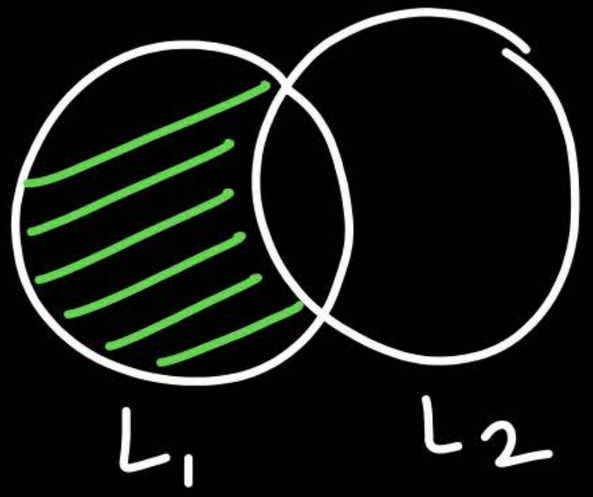
viii) $L = \Sigma^+ \Rightarrow \bar{L} = \{\varepsilon\}$

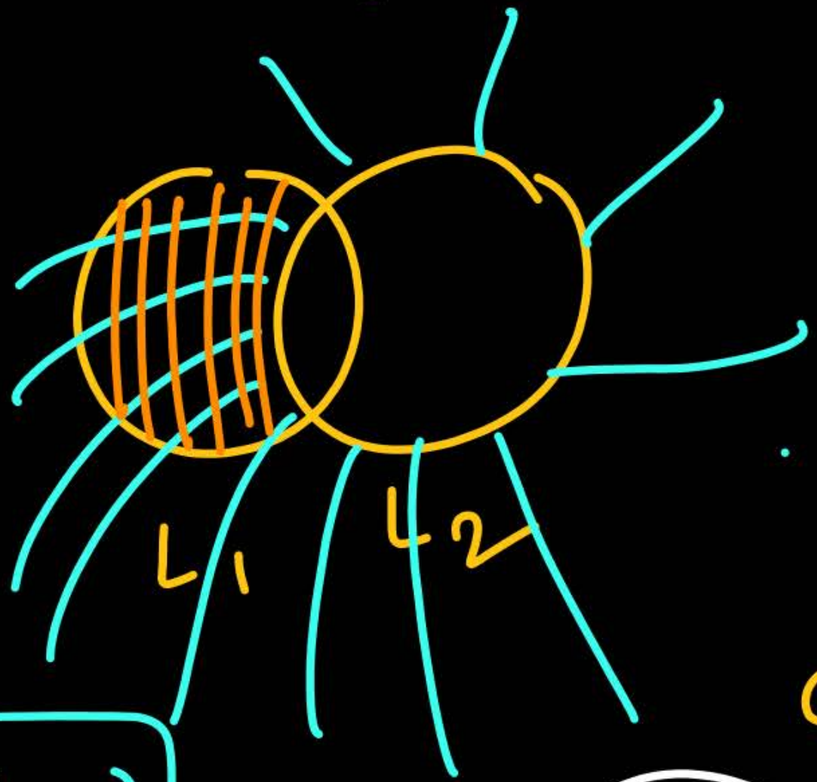
④ Difference for regular languages

↳ closed

$Reg_1 - Reg_2 \Longrightarrow$ Always Regular

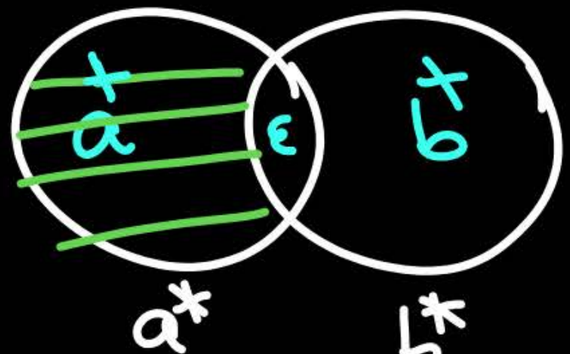
Def I) $L_1 - L_2 = L_1 \cap \overline{L_2}$





Def II) $L_1 - L_2 = L_1 - (L_1 \cap L_2)$

$a^* - b^* \Rightarrow a^+$



$= a^+$

i) $L_1 = \phi$, L_2 = Any $\Rightarrow$ $L_1 - L_2 = \phi$

$L_2 - L_1 = L_2$

ii) $L_1 = \Sigma^*$, L_2 = Any $\Rightarrow$ $L_1 - L_2 = \Sigma^* - L_2 = \overline{L_2}$

$L_2 - L_1 = L_2 - \Sigma^* = \phi$

iii) $L_1 = a^*$, $L_2 = b^*$ $\Rightarrow$ $L_1 - L_2 = a^* - b^* \Rightarrow a^+$

$L_2 - L_1 = b^* - a^* \Rightarrow b^+$

$\Sigma = \{a\}$ iv) $L_1 = a^*$, $L_2 = (aa)^*$ $\Rightarrow$ $L_1 - L_2 = a(aa)^*$

All a's, even a's

$L_2 - L_1 = \phi$

I) $L - \phi = L$

II) $L - \Sigma^* = \phi$

(5) Concatenation for regular languages

↳ closed

$L_1 \cdot L_2 \Rightarrow$ Always Regular

L_1 reg $\cdot$ L_2 reg

$L_1 \Rightarrow RE_1$

$L_2 \Rightarrow RE_2$

$L_1 \cdot L_2 \Rightarrow RE_1 . RE_2$

New Regular Exp.

i) $L_1 = \phi$, $L_2 = Any$ $\Rightarrow$ $L_1 . L_2 = \phi$

$L_2 . L_1 = \phi$

ii) $L_1 = \Sigma^*$, $L_2 = \{a\}$ $\Rightarrow$ $L_1 . L_2 = \Sigma^* a$

$L_2 . L_1 = a\Sigma^*$

iii) $L_1 = a^*$, $L_2 = b^*$ $\Rightarrow$ $L_1 . L_2 = a^* b^*$

$L_2 . L_1 = b^* . a^*$

iv) $L_1 = a^*$, $L_2 = (aa)^*$ $\Rightarrow$ $L_1 . L_2 = L_1$

$L_2 . L_1 = L_1$

v) $L_1 = \Sigma^*$, $L_2 = a\Sigma^*$ $\Rightarrow$ $L_1 . L_2 = \Sigma^* a \Sigma^*$

$L_2 . L_1 = L_2$

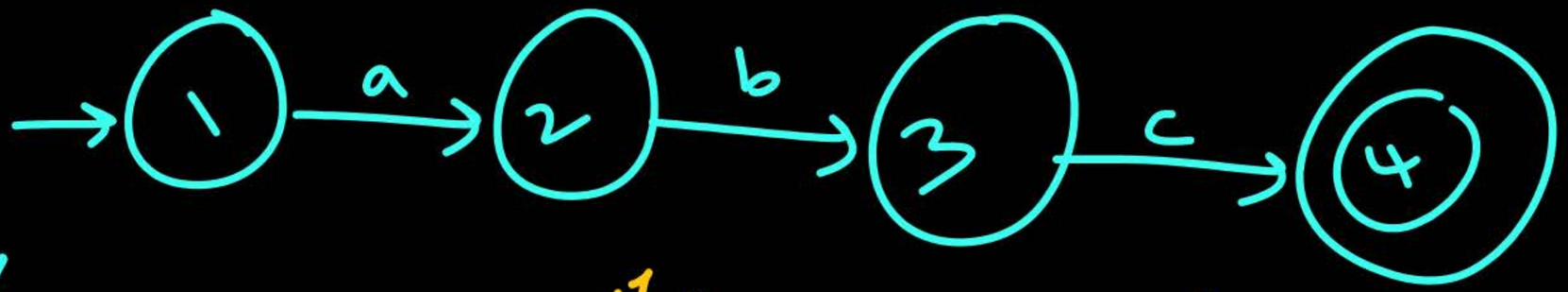
⑥ Reversal for regular languages

↳ closed

If L is Reg, L^{Rev} ⇒ Regular

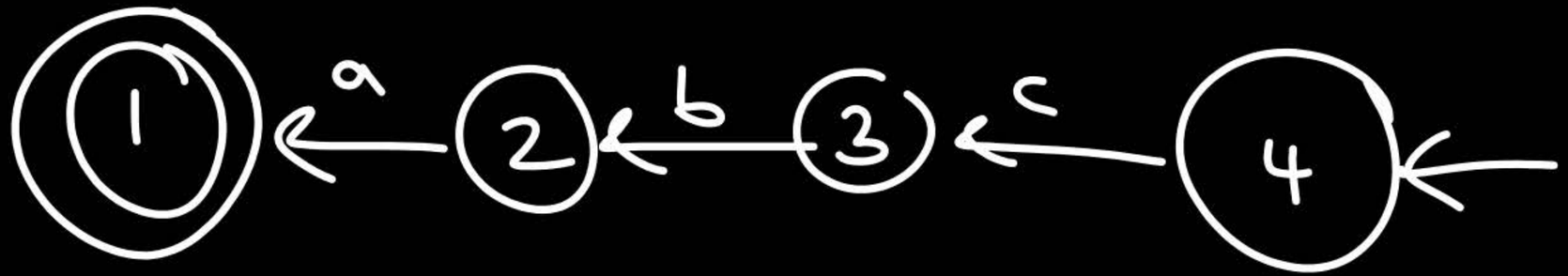
(Reg)^{Reversal} ⇒ Regular

$L = \{abc\}$



$L^R = L^{Rev} = \{cba\}$

Step1 : Make one final
Step2 : Initial ↔ final
Step3 : Reverse all transitions



$$L = \{a, ab, abc\}$$

$$\downarrow Rev$$

$$L^{Rev} = \{a, ba, cba\}$$

i) $L = \phi \Rightarrow L^{Rev} = \phi = L$

ii) $L = \Sigma^* \Rightarrow L^{Rev} = \Sigma^* = L$

iii) $L = a\Sigma^* \Rightarrow L^{Rev} = \Sigma^* a$

iv) $L = \Sigma^* b \Rightarrow L^{Rev} = b\Sigma^*$

v) $L = ab^* \Rightarrow L^{Rev} = b^* a$

vi) $L = \Sigma^* a \Sigma^* \Rightarrow L^{Rev} = \Sigma^* a \Sigma^*$

vii) $L = a^* b^* \Rightarrow L^{Rev} = b^* a^*$

viii) $\left(L^{Rev}\right)^{Rev} = L$

ix) $\overline{\left(\overline{L}\right)} = L$

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

$$(\Sigma^*)^{Rev} = \{\varepsilon, a, b, aa, ab, ba, bb, \ldots\}$$

$$= \Sigma^*$$

I) Reversal of Regular Language is Regular

II) Reversal of Not regular language is Not Regular

Note :

I) L is Reg iff $\bar{L}$ is Reg

II) L is Reg iff L^{Rev} is Reg

(7) Kleene star (closure) (kleene closure)

If L is Reg then L^* is Regular

Note:

i) If L^* is Regular then L is either Reg or Not Reg

ii) If L^+ is Regular then L is either Reg or Not Reg

(8) Kleene plus (Positive closure)

If L is Reg then L^+ is Regular

$\{a^{prime}\}^*$ is Regular

But $\{a^{prime}\}$ is not reg

i) $L = \phi \Rightarrow L^* = \{\varepsilon\}$

$L^+ = \phi = L$

ii) $L = \Sigma^* \Rightarrow L^* = \Sigma^* = L$

$L^+ = \Sigma^* = L$

iii) $L = \{a, b\} \Rightarrow L^* = (a+b)^*$

$L^+ = (a+b)^+$

iv) $L = a^* + b^* \Rightarrow L^* = (a+b)^*$

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

v) $L = a^* b^* \Rightarrow L^* = (a+b)^*$

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

vi) $L = b^* a^* \Rightarrow L^* = (a+b)^*$

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

vii) $L = a^+ + b^+ \Rightarrow L^* = (a+b)^*$

$L^+ = (a+b)^+$

viii) $L = a^+ b^+ \Rightarrow L^* = (a^+ b^+)^*$

$L^+ = (a^+ b^+)^+$

$\neq (a+b)^*$

ix) $L = a + b^* \Rightarrow L^* = (a+b)^*$

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

$$(a^+b^+)^+ \neq (ab)^+$$

$$\{ab, aab, abb, \ldots \ldots \text{abab}, \text{abaab}, \ldots\} \supset \{ab, abab, (ab)^3, (ab)^4 \ldots\}$$

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

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

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

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

$$= (a+b)^+ + \varepsilon$$

$$L = a^* + b^*$$

$$\Downarrow$$

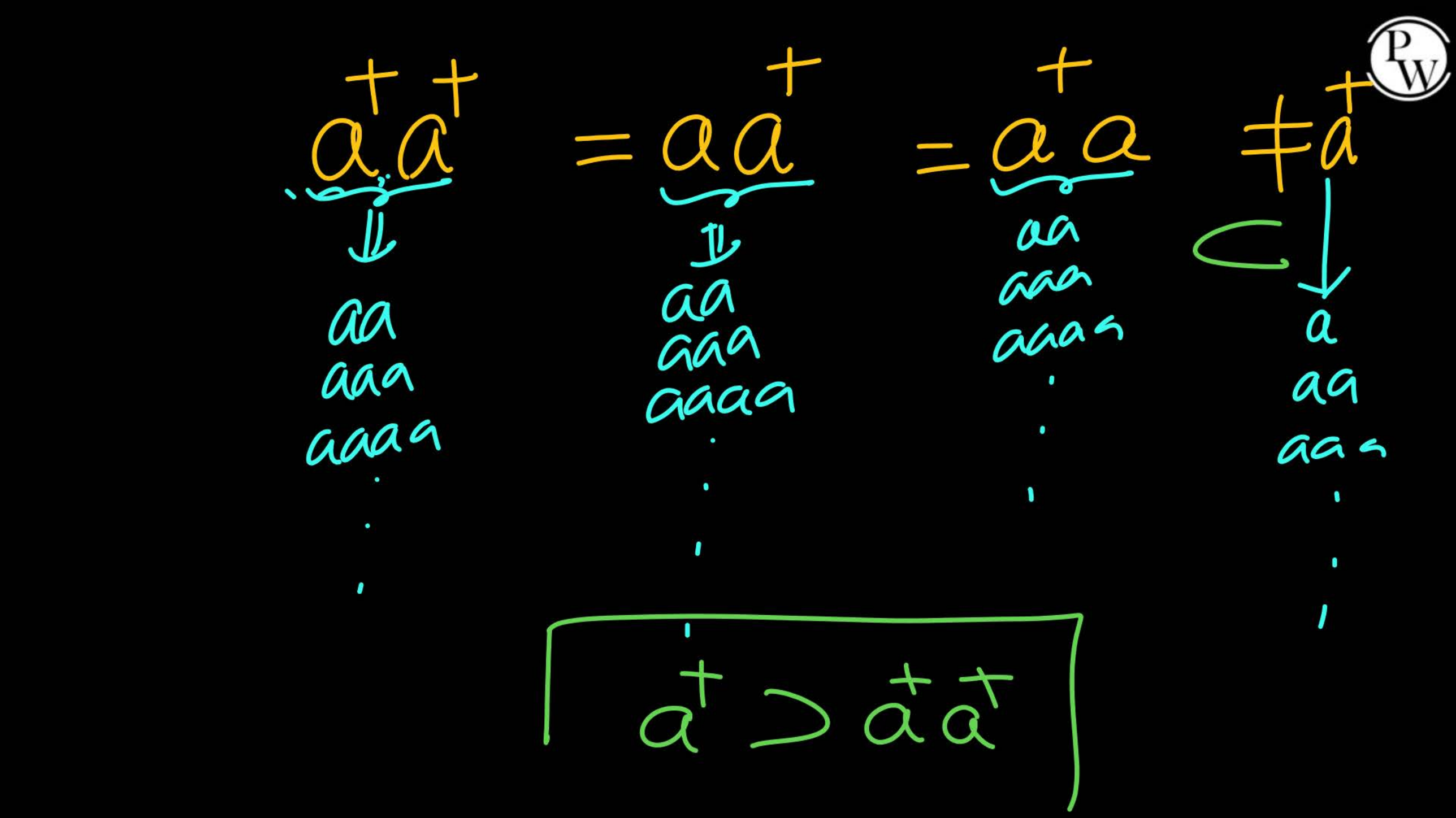
$$L^* = (a^* + b^*)^*$$

$$= (a+b)^* = \Sigma^*$$

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

$$= \{\varepsilon, a, b, aa, ab, ba, bb, \ldots\}$$

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


$a^+ a^+$
$= a^+ a$
$= a a^+$
$\neq a^+$
aa
aaa
aaaa
aa
aaa
aaaa
aa
aaa
aaaa
a
aa
aaa
$a^+ \supset a^+ a^+$

$aba(a+b)^*$ — some strings

$ababb(a+b)^*$ — some

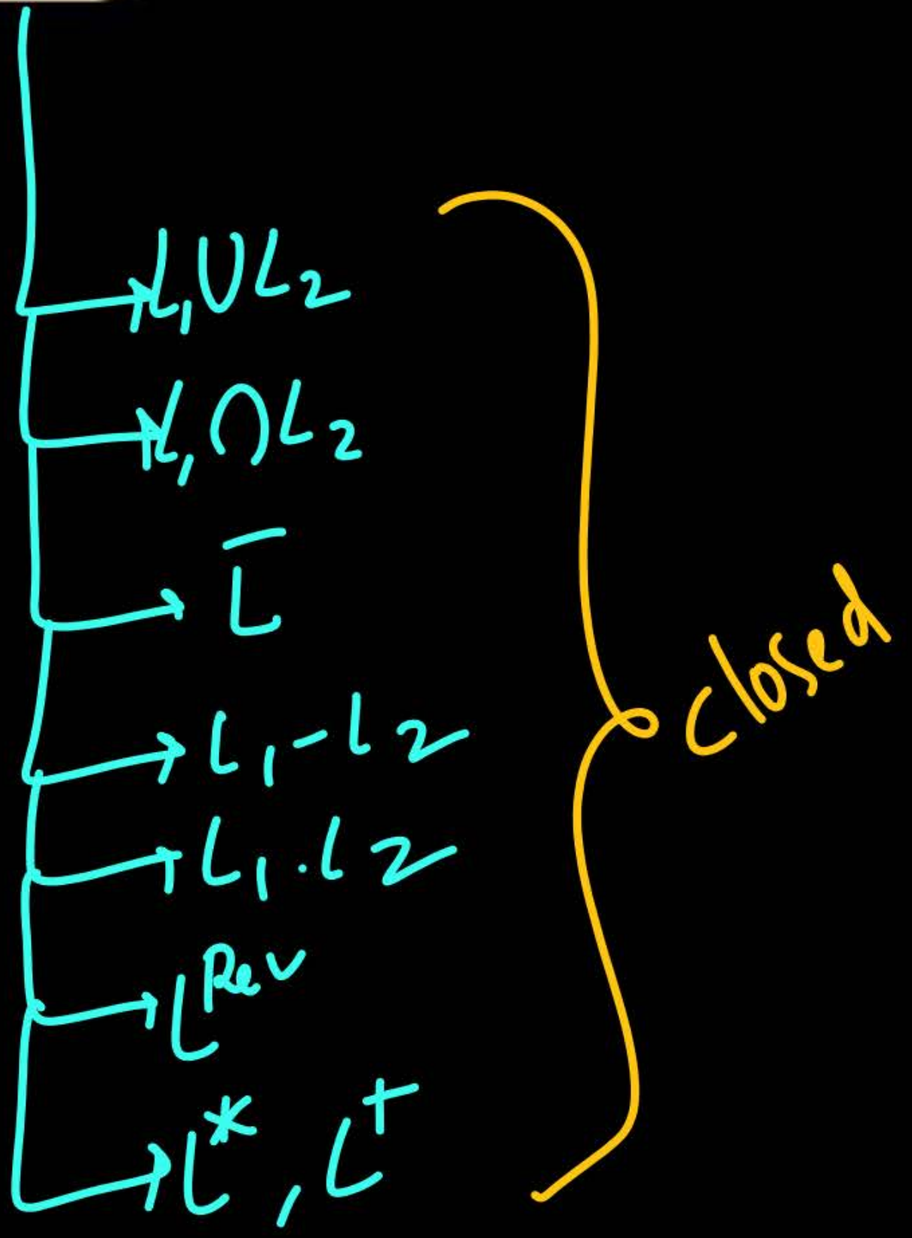
$= \{ab, aba, abb, \ldots\}$

L = Set of all strings starting with ab

$\neq aba(a+b)^* \subset ab\Sigma^*$

$= ab\Sigma^*$

Summary




THANK YOU
GW
SOLDIERS !