Regular Expressions

Regular language → F.A. (finite automata)
→ R.E. (Regular expressions)

a+b (OR) a.b (AND) a* (Iteration)

Q.) Using regular expression describe the language consisting of all string $\Sigma = (0,1)$ with at least two consecutive 0s.

→

$$r = (0+1)^*.0.0.(0+1)^*$$

Q.) Using regular expression represent language $\Sigma = (0,1,2)$ such that every string from the language contains any number of 0s followed by any number of 1s followed by any no. of 2s.

→

$$r = 0^*.1^*.2^*$$

Q.) Using ~~reo~~ R.E. represent the language $\Sigma = (a,b)$ with all string starting & ending with 'a' & with any no. of 'b' in between

→

$$a.b^*.a$$

0101011

25/7/23.

Q.) If $L(r)$ = { ~~set of a~~ $\Sigma = (0,1)$ } ending with '011'. find 'r'

→

$r = \cancel{(0^* + 1^*).0.1.1}$

$r = (0+1)^*.0.1.1$

Q.) $L(r)$ = { a, c, ab, cb, abb, cbb, abbb, ... }. r = ?

→

$r = (a+c).b^*$

Q.) $L(r)$ = { $\Sigma = (a,b)$ } containing ~~atleast~~ exactly 2 a's.

→ $r = b^*.a.b^*.a.b^*$

Q.) represent language over $\Sigma = (a,b)$ containing atleast 1 a & 1 b.

→ $\cancel{a.a^*.b.b^*}$ $\cancel{(a+b)^* (a+b)^*}$

$\cancel{(a+b)^* (a+b).(a+b)^*}$, $\cancel{(a+b)^* ((a.b)+(b.a)) (a+b)^*}$

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

or

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

Q.) Using R.E. represent { all strings of a's & b's } containing atleast one combination of double letters

→

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

Q.) $L(r)$ = { ϵ, x, xx, xxx, xxxx, xxxxx, ... }

→

$r = \cancel{(\phi + x^*)}\ (\phi + x)^5$

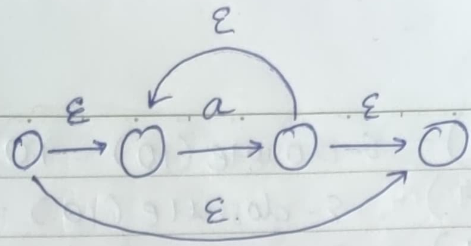
28/7/23

Q. ~~Reg~~ R.E → DFA (Subset method)

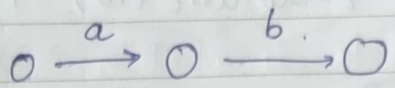
Step 1: Construction of transition diagram for given RE by using NFA with ε moves

28/7/23

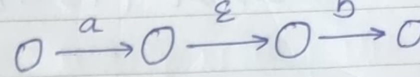
a* :



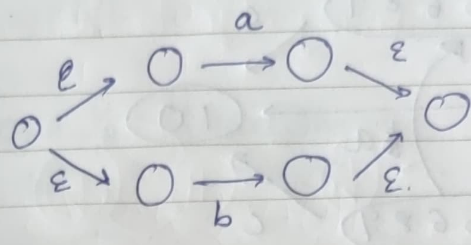
a.b :



or

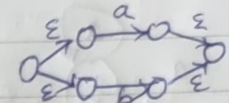


a+b :

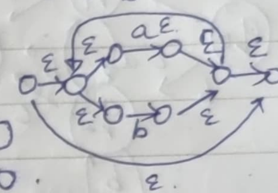


Q.) DFA for R.E. ~~(a+b)*c~~. (a+b)* c.

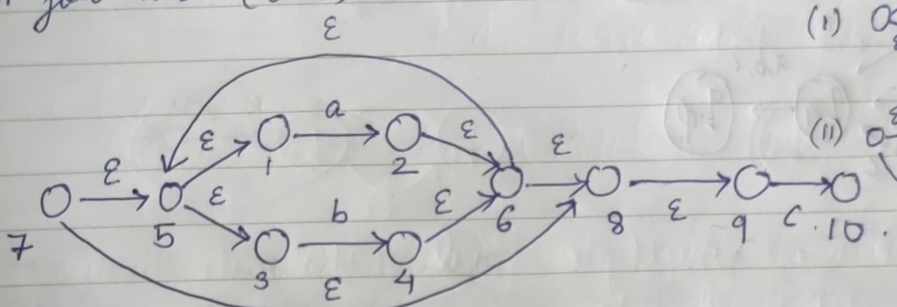
(i)



(ii)



(iii)



states ε-closure
7 7 5 1 3 8 9

a successor of (1 3 5 7 8 9)
ε-closure of (δ(1,a) ∪ δ(3,a) ∪ δ(5,a) ∪ δ(7,a) ∪ δ(8,a) ∪ δ(9,a))
ε-closure (2 ∪ φ ∪ φ ∪ φ ∪ φ ∪ φ) = ε-closure (2)
= 2 6 8 9 5 1 3 ⇒ 1 2 3 5 6 8 9

b successor (1 3 5 7 8 9) = ε-closure (4) = 4 6 8 9 5 1 3
= 1 3 4 5 6 8 9

a successor (1 2 3 5 6 8 9) = ε-closure (2) = 1 2 3 5 6 8 9

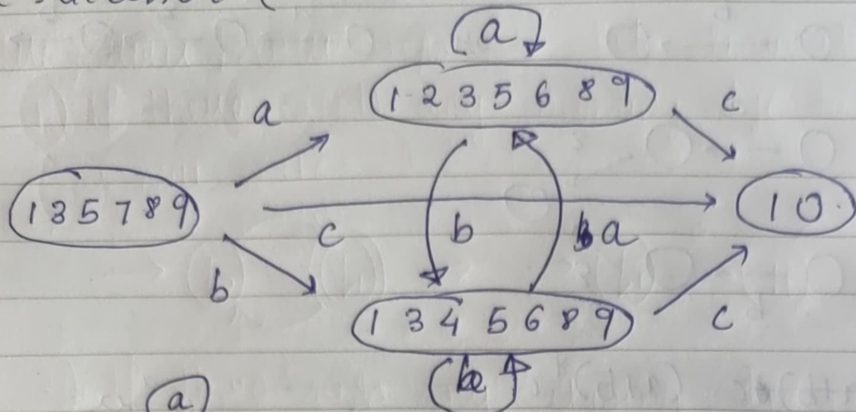
b successor (1 2 3 5 6 8 9) = ε-closure (4) = 1 3 4 5 6 8 9

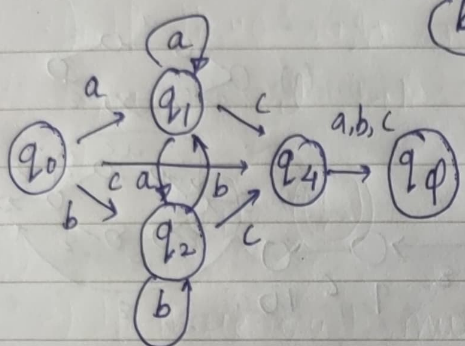
b successor (1 3 4 5 6 8 9) = ε-closure (4) = 1 3 4 5 6 8 9

a successor (1 3 4 5 6 8 9) = ε-closure (1) = 1 2 3 5 6 8 9

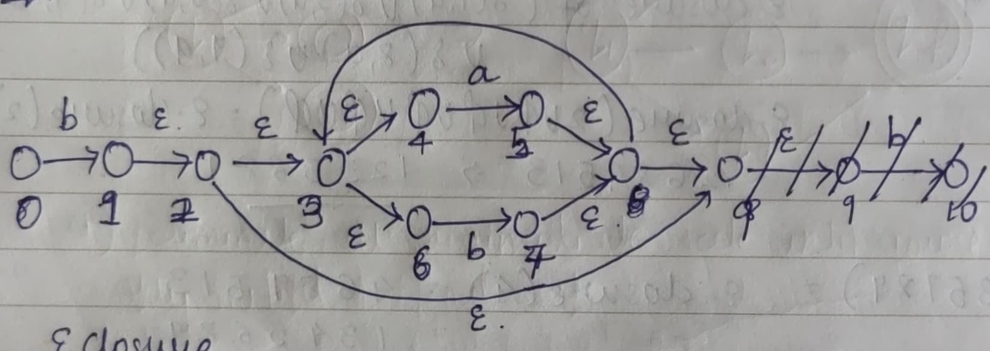
28/7/23

c ~~closure~~ successor (7,51,35789) = ε-closure (10) = 10
c successor (1 2 3 5 6 8 9) = ε-closure (10) = 10
c successor (1 3 4 5 6 8 9) = ε-closure (10) = 10



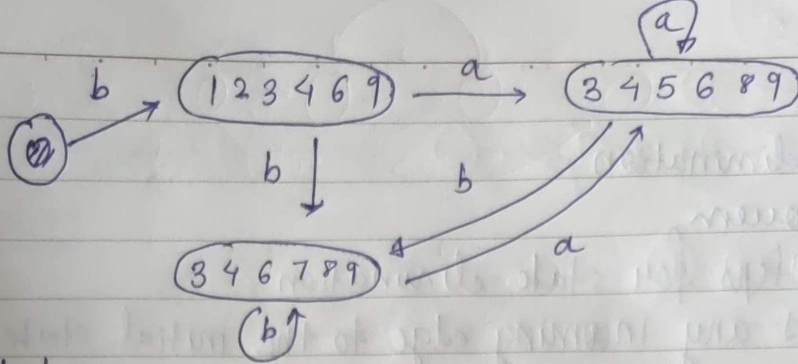


Q.) DFA for R.E. b.(a+b)*



ε closure
ε-closure (1) = 1 2 3 4 5 9
a successor (1 2 3 4 5 9) = ε closure (5) = 5 8 9 3 4 6
= 3 4 5 6 8 9
b successor (1 2 3 4 6 9) = ε closure (7) = 7 8 9 3 4 6
= 3 4 6 7 8 9
a successor (345689) = ε closure (5) = 345689
b successor (345689) = ε closure (7) = 346789
a successor (346789) = ε closure (5) = 345689
b successor (346789) = ε closure (7) = 346789

28/7/23

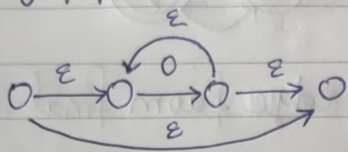


1/8/23

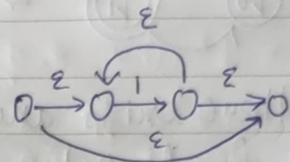
Q.) Construct DFA for following R.E

$r = 0^* . 1^* . 2^*$

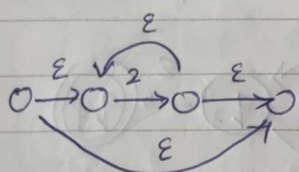
0^* :

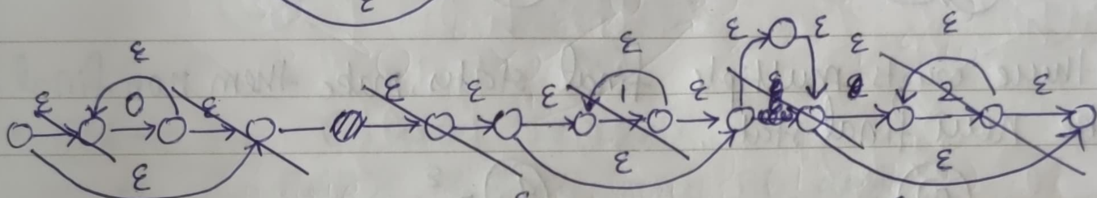


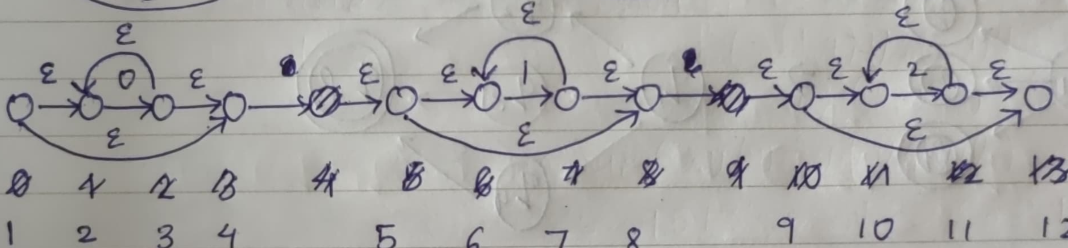
1^*

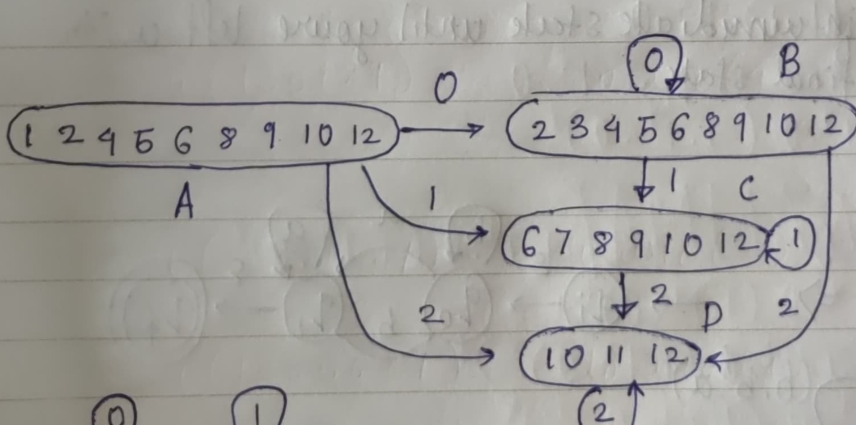


2^* :

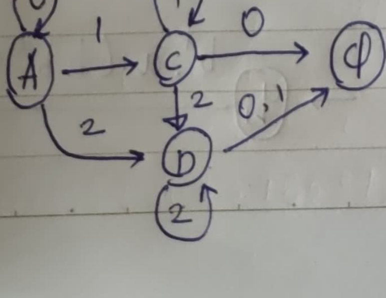








	0	1	2
A	B	C	D
~~B~~	~~B~~	~~C~~	~~D~~
C	Φ	C	D
D	Φ	Φ	D.



1|8|23.

→ DFA to R.E

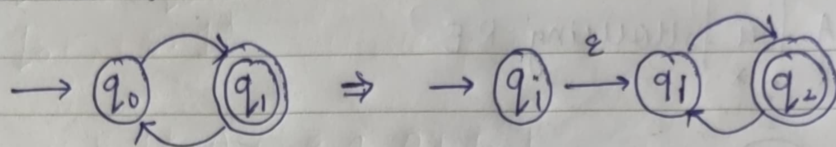
→ state/ loop elimination

→ Arden's theorem

We follow below steps for state elimination:

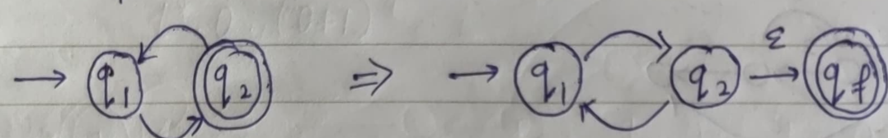
1.) If there exists any incoming edge to the initial state, add a new state.

Ex:



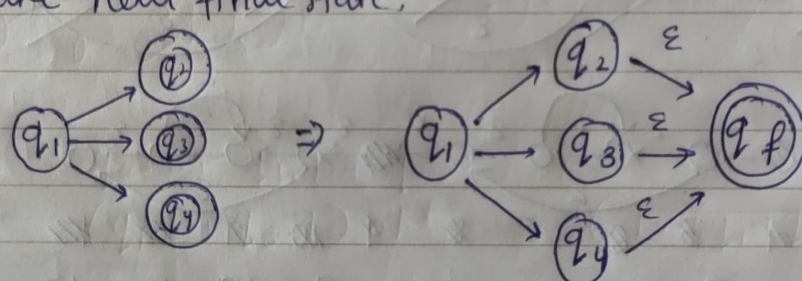
2.) If there exists any outgoing edge from final state, create new final state

Ex:



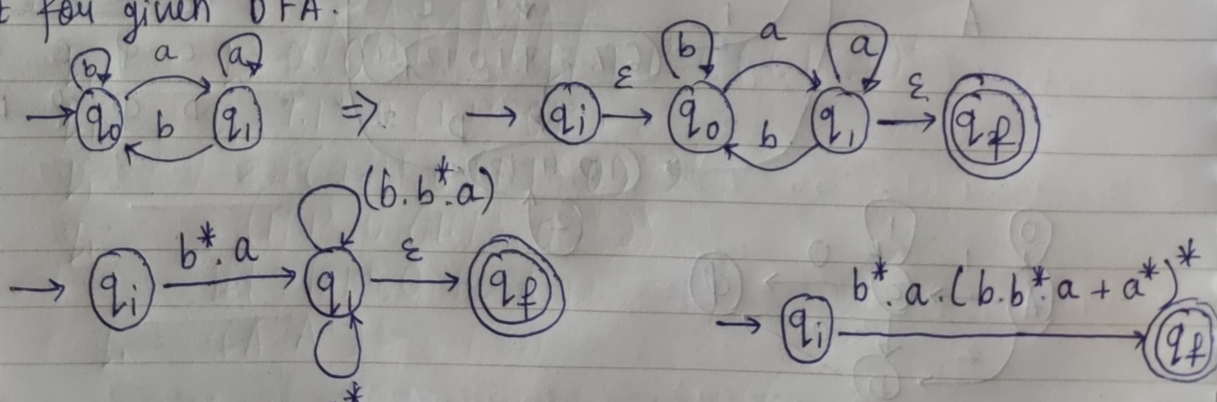
3.) If there exists multiple final states, make them non final & make new final state.

Ex:

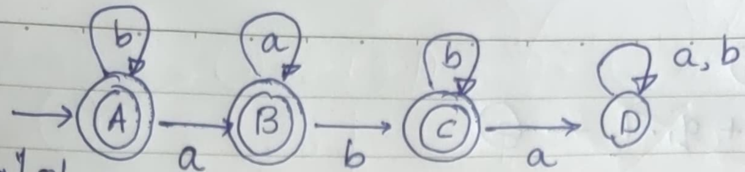


4.) Eliminate all intermediate state until you're left with only initial & final state.

Q.) RE for given DFA.

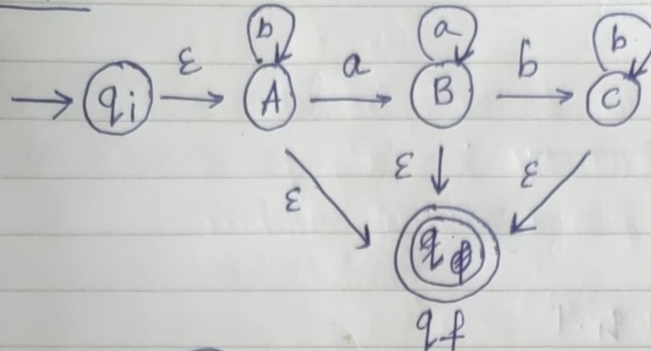


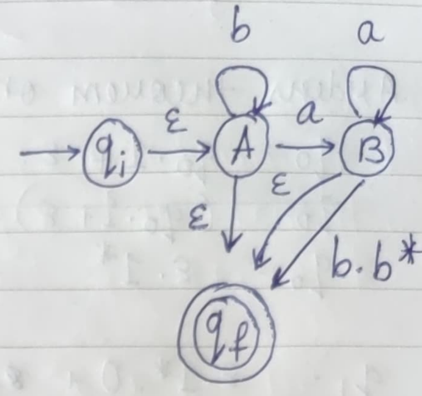
Q.)

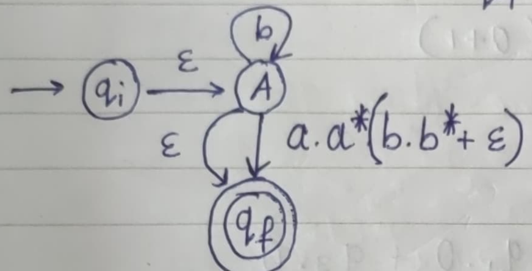


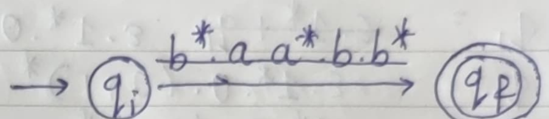
4/8/23

→









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

→ <u>DFA to R.E</u> (Arden's Theorem)

→ If ~~R=P+RQ~~ P, Q, R are RE and

1.) $R = P + RQ$ or $R = RQ + P$
then, $R = PQ^*$

II.) $R = P + QR$ or $R = QR + P$
then, $R = Q^*.P$

<u>Proof</u>:-

* $R = P + RQ$

$R = P + (P + RQ).Q = P + P.Q + RQ^2$

$R = P + P.Q + (P + RQ)Q^2 = P + P.Q + P.Q^2 + RQ^3$

$\vdots \qquad\qquad \vdots$

$R = P(\varepsilon + Q + Q^2 + Q^3 + \dots) = P.Q^*$

* $R = P + Q.R$

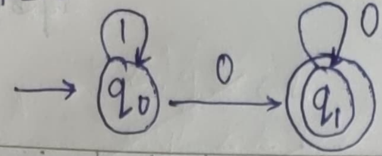
$R = P + Q.(P + Q.R) = P + Q.P + Q^2.R$

$R = P + Q.P + Q^2.(P + Q.R) = P + QP + Q^2.P + Q^3.R$

$R = P(\varepsilon + Q + Q^2 + Q^3 + \dots).P =$ ~~P.Q~~ $Q^*.P$

4|8|23.

Q.)



$q_0 = q_0 . 1$ — ①

$q_1 = q_0 . 0 + q_1 . 0 + q_1 . 1$ — ②

Ardens theorem on ①

$q_0 = q_0 . 1 + \varepsilon$

$\cancel{q_0 = (q_0 . 1 + \varepsilon) . 1 + \varepsilon}$

$q_0 = \varepsilon . 1^*$

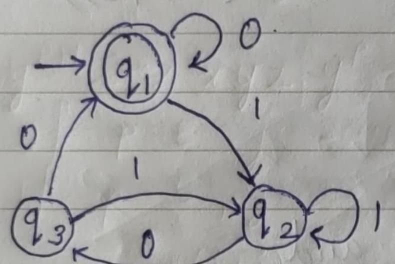
$q_1 = \varepsilon . 1^* . 0 + \varepsilon . q_1 . 0 + q_1 . 1$

$q_1 = \cancel{\varepsilon . 1^* + q_1}\ \varepsilon . 1^* . 0 + q_1 (0+1)$

$q_1 = \varepsilon . 1^* . 0 . (0+1)^*$

$q_1 = 1^* . 0 . (0+1)^*$

Q.)



$q_1 = q_1 . 0 + q_3 . 0$

$q_2 = q_1 . 1 + q_2 . 1 + q_3 . 1$

$q_3 = q_2 . 0$

$q_2 = q_2 . 1 + q_1 . 1 + q_3 . 1$

$q_2 = (q_1 . 1 + q_3 . 1) . 1^* = \cancel{(1 . (q_1 + q_3)) . 1^*}$

$q_3 = (q_1 . 1 + q_3 . 1) . 1^* . 0$

$q_3 = q_3 . 1 . 1^* . 0 + q_1 . 1 . 1^* . 0$

$q_3 = q_1 . 1 . 1^* . 0 \, (1 . 1^* . 0)^*$

$q_1 = q_3 . 0 . 0^* = q_1 . 1 . 1^* . 0 \, (1 . 1^* . 0)^*$

$q_1 = q_1 . 7 . 1^* . 0 \, (1 . 1^* . 0)^* + \varepsilon .$

$q_1 = \varepsilon . (1 . 1^* . 0 \, (1 . 1^* . 0)^*)^*$

$q_1 = (1 . 1^* . 0 . (1 . 1^* . 0)^*)^*$

let $1 . 1^* . 0$ be A

$q_1 = (A . A^*)^*$

7/8/23

→ Pumping lemma for regular language :-

$r = a^n b^n \quad n \geq 1$.

let 'L' be a regular language, then there exists a constant 'n' (which depends on 'L') such that, for every string in 'w' in 'L' $|w| \geq n$. We can break 'w' into 3 strings,

$w = xyz$, such that

1.) $y \neq \varepsilon$.

2.) $|xy| \leq n$.

3.) $\forall k \geq 0, \; xy^k z \in L$.

Q.) 'w' in L $|w| \geq n$.

$w = xyz$ such that

$w = a^3b^3 = \underbrace{aa}_{x}\underbrace{ab}_{y}\underbrace{bb}_{z}$.

If $k=1$, $xyz = aaabbb \in L$.

if $k=2$, $xy^2z = aaababbb \notin L$.

If $k=3$, $xy^3z = aaabababbb \notin L$.

$\therefore a^3b^3$ is not a regular expression

Q.) $w = \epsilon . a . a . a . b . b . b$.

$x = \varepsilon$, $y = aaa$, $z = bbb$.

If $k=1$, $xyz = aaabbb \in L$

If $k=2$, $xy^2z = \cancel{a}\,\varepsilon(aaa)^2bbb = aaaaaabbb \notin L$.

If $k=3$, $xy^3z = aaaaaaaaabbb. \notin L$.

Q.) $L = \{ a^p \mid p \text{ is prime} \}$.

$w = xyz$. $\quad w = a^i . a^j . a^k = a^p$

$\downarrow \downarrow \downarrow$

$a^i \; a^j \; a^k \quad i+j+k = p$.

7/8/23.

$x \cdot y z \in L$.

$\cancel{x \cdot y^{p} \cdot z \in L}$.

$\cancel{x \cdot y^{p+1} \cdot z \in L}$.

$\cancel{x \cdot y^{p+1} z = xyz \cdot (y)^{p}}$.

$=$

let a word 'w' belongs to 'L' be split into 3 substring such that, $w = xyz$, where $x = a^{i}$, $y = a^{j}$, $z = a^{k}$ & $i+j+k = p$ where p is a prime number.

$$w = a^{i} a^{j} a^{k} \Rightarrow a^{(j+j+k)} \Rightarrow a^{p}$$

If the language were to be prime then the pump word $xy^{p+1} z \in L$.

$$a^{p+jp}$$

$$x \cdot y^{p+1} z \Rightarrow xyz \cdot y^{p} \Rightarrow a^{p} (a^{j})^{p} \Rightarrow a^{p(1+j)}$$

if $p = 5$, $j = 2$, then $(i+j)p = 15 \notin L$.

If $p = 5$, $j = 3$, then $(i+j)p = 20 \notin L$.

Hence $a^{p(i+j)}$ is not always prime so we can conclude given language is not regular.

Q.) $L = \{ a^{n^2} \mid$ n² is perfect square ~~n is prime~~ $\}$.

$$w = a^{i} \cancel{g} a^{j} a^{k} \Rightarrow a^{i+j+k} \Rightarrow$$