1. (a) Consider the language S^*, where $S = \{aa, b\}$. How many words does this language have of length 4? of length 5? of length 6? What can be said in general?

Sol. Length 4 :- { aabb, aaaa, bbbb, bbaa, baab } (5)

Length 5 :- { aaaab, bbbbb, aabbb, bbbaa, aabaa, baaaa, baabb, bbaab } (8)

Length 6 :- { aaaaaa, aabbbb, aabbaa, bbaaaa, bbbbbb, aaaabb, bbaabb, bbbbaa, baaaab, baabaa, aabaab, bbbaab, baabbb } (13)

Words of this language S^* are making a fibonacci series in the no. of words formed, when we increase the length of words by 1.

(b) Let $S = \{ab, bb\}$ and let $T = \{ab, bb, bbbb\}$. Show that $S^* = T^*$.

Sol. $S^* = \{ab, bb, abbb, bbab, abbbab, bbabab, abbbbb, bbabbb, bbbb, abab, abbbbb, abbbbbbb, \ldots\}$

$T^* = \{ab, bb, bbbb, abbb, bbab, abab, abbbbb, abbbbbbb, \ldots\}$

S is contained in T.

T is contained in S.

$S^* = T^*$.

(c) Give a regular expression for the language of all the words that do not have 'aa' as substring.

Sol. $b^*(abb^*)^*(\Lambda + a)$

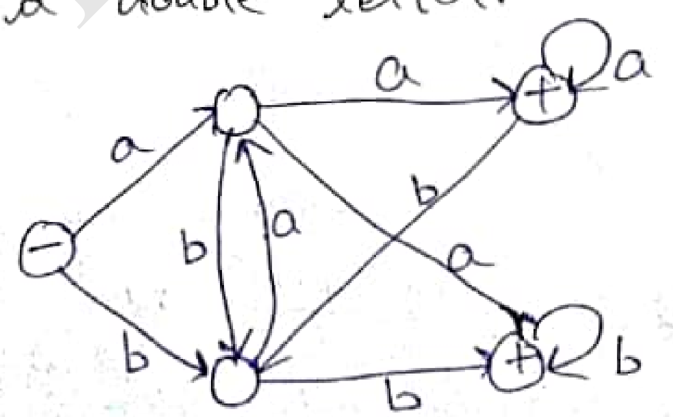
(d) Generate a CFG for $b^* a^*$.

Sol. $S \rightarrow XY$

$X \rightarrow bX \mid \Lambda$

$Y \rightarrow aY \mid \Lambda$

(e) Design a deterministic Finite automata for the language of all words that end in a double letter.



(7) Using pumping lemma, prove that language $a^n b^{2n}$, $n>=0$ is not regular.

Sol. Language is

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

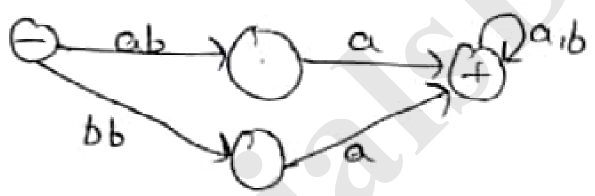
Pumping it in xy^nz form

If we consider a^n part to be y and rest of it be z, and we pump it sometimes, it they will remain half of the no. of b's. (x is null)

If we consider b^{2n} part to be y and a^n to be x and we pump y sometimes, then it will not remain double of no. of a's (z is n

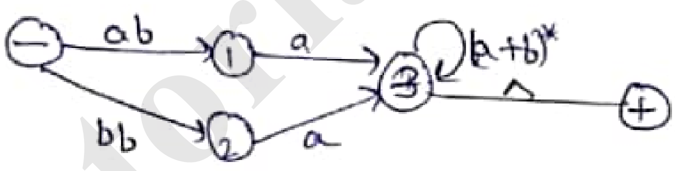
Hence the language is not regular.

(8) Convert the following transition graph into its equivalent regular expression.

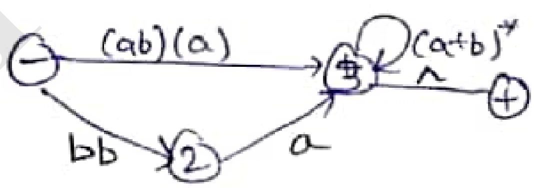


(1)

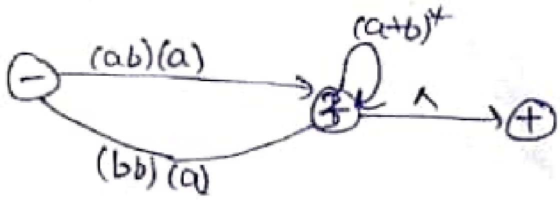
Sol



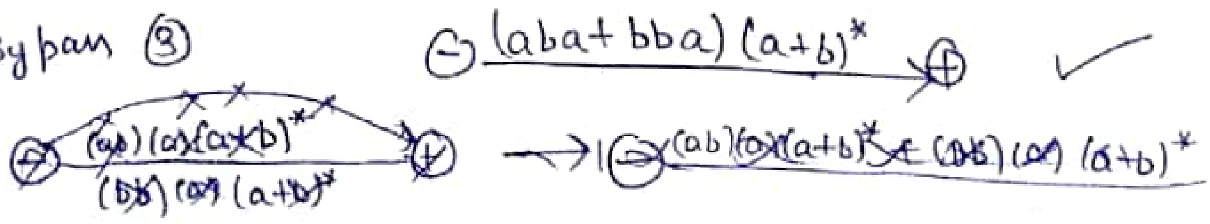
By Pass (1)



By Pass (2)



By pass (3)



R.E. $(ab)(a)(a+b)^* + (bb)(a)(a+b)^*$.

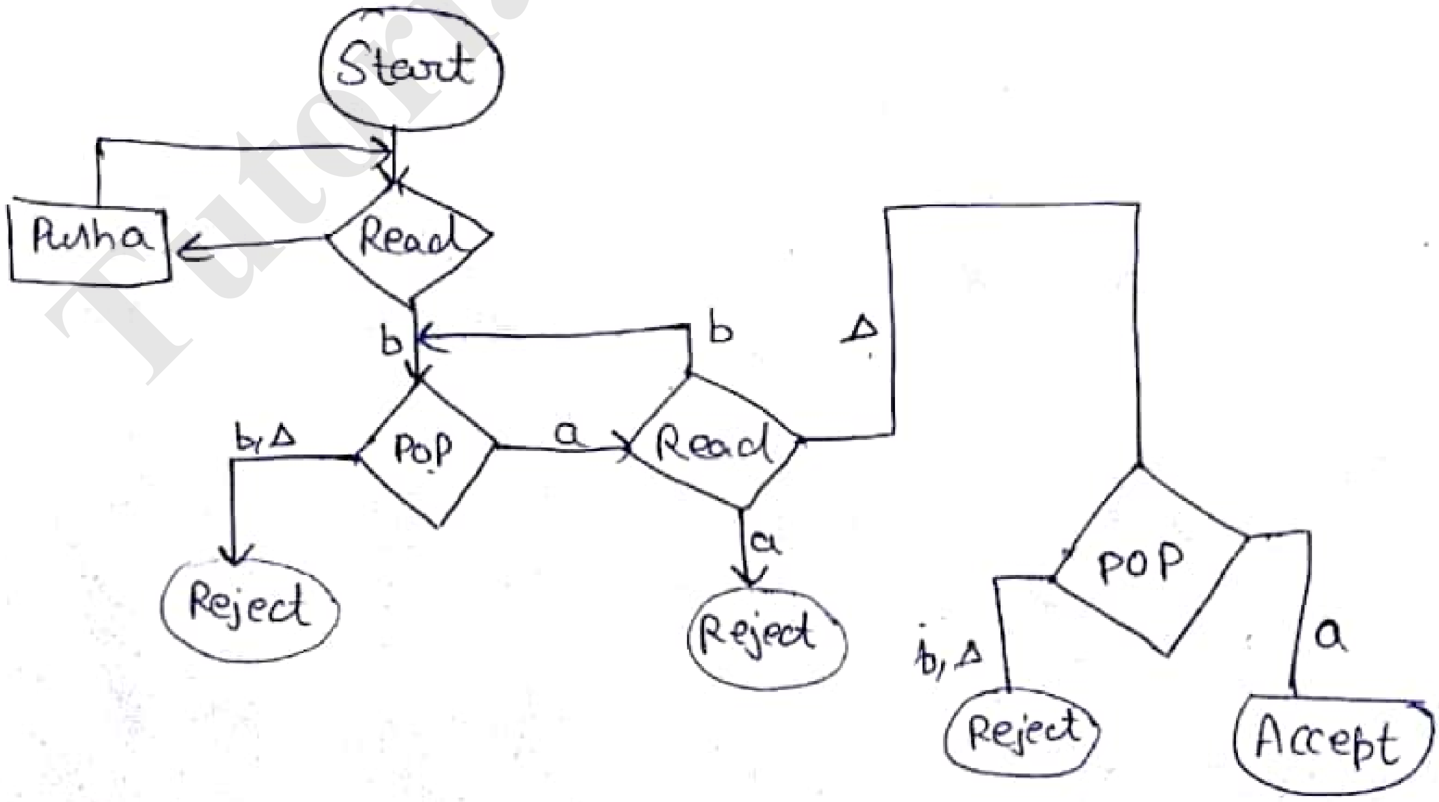
Q Show that the complement of a recursive language is also recursive.

Sol. If L is decided by TM $M = \{K, \Sigma, \delta, s, \{y, n\}\}$, then L is decided by TM $M' = \{K, \Sigma, \delta', s, \{y, n\}\}$ which is identical to M except that it reverses the roles of the two special halting states y and n. That is, δ' is defined as follows

$$\delta'(q, a) = \begin{cases} n & \text{if } \delta(q, a) = y, \\ y & \text{if } \delta(q, a) = n, \\ \delta(q, a) & \text{otherwise.} \end{cases}$$

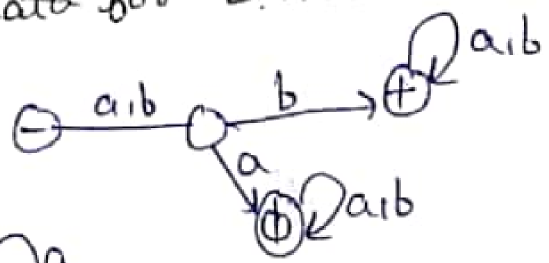
It is clear that $M'(w) = y$ if and only if $M(w) = n$ & therefore M' decides $\bar{L}$.

(i) Construct a push down automata for $a^n b^{n-1}$ where $n >= 1$.

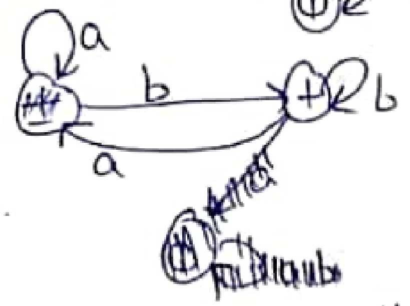


(j) If $L1 = (a+b)b(a+b)^*$ and $L2 = (a+b)^*b$. Find a regular expression and deterministic finite automata for $L1 \cap L2$.

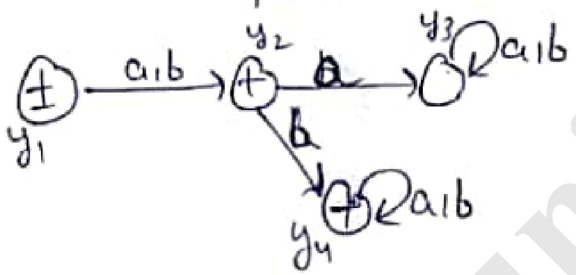
L1:



L2:



L_1':



L_2':

$\pm\ Z_1 = x_1$ or y_1

$+(Z_2)$ a at $Z_1 = x_1$ or y_2

$+(Z_3)$ b at $Z_1 = x_2$ or y_2

$+(Z_4)$ a at $Z_2 = x_1$ or y_3

(Z_5) b at $Z_2 = x_2$ or y_4

Z_4 a at $Z_3 = x_1$ or y_3

Z_5 b at $Z_3 = x_2$ or y_4

Z_4 a at $Z_4 = x_1$ or y_3

$+(Z_6)$ b at $Z_4 = x_2$ or y_3

$+(Z_7)$ a at $Z_5 = x_1$ or y_4

(Z_5) b at $Z_5 = x_2$ or y_4

(Z_4) a at $Z_6 = x_1$ or y_3

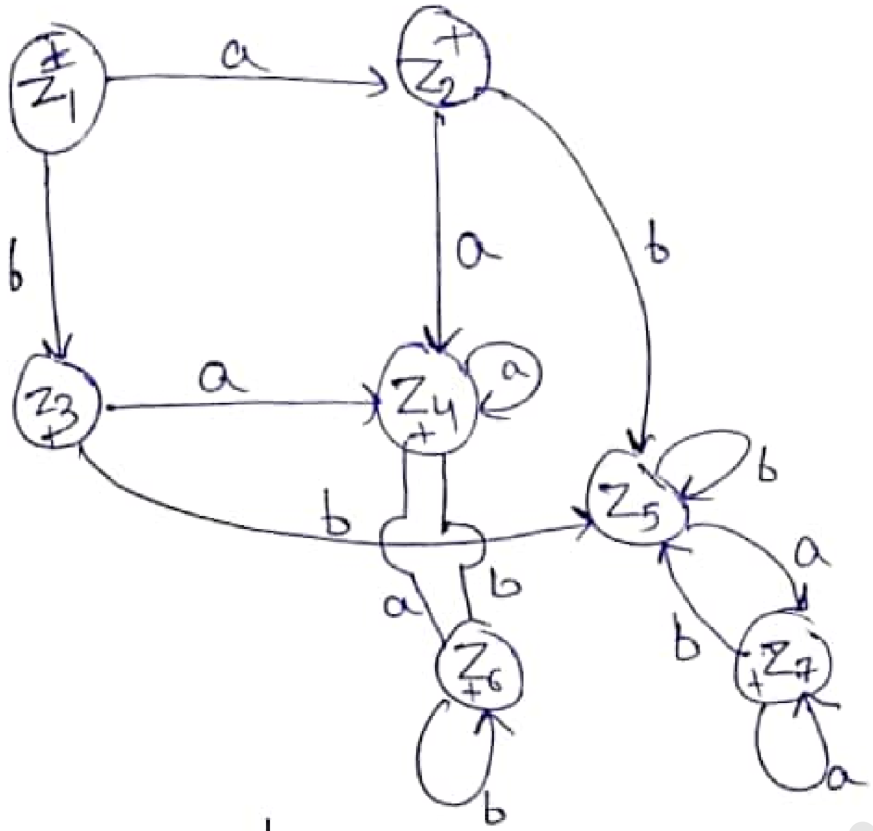
(Z_6) b at $Z_6 = x_2$ or y_3

(Z_7) a at $Z_7 = x_1$ or y_4

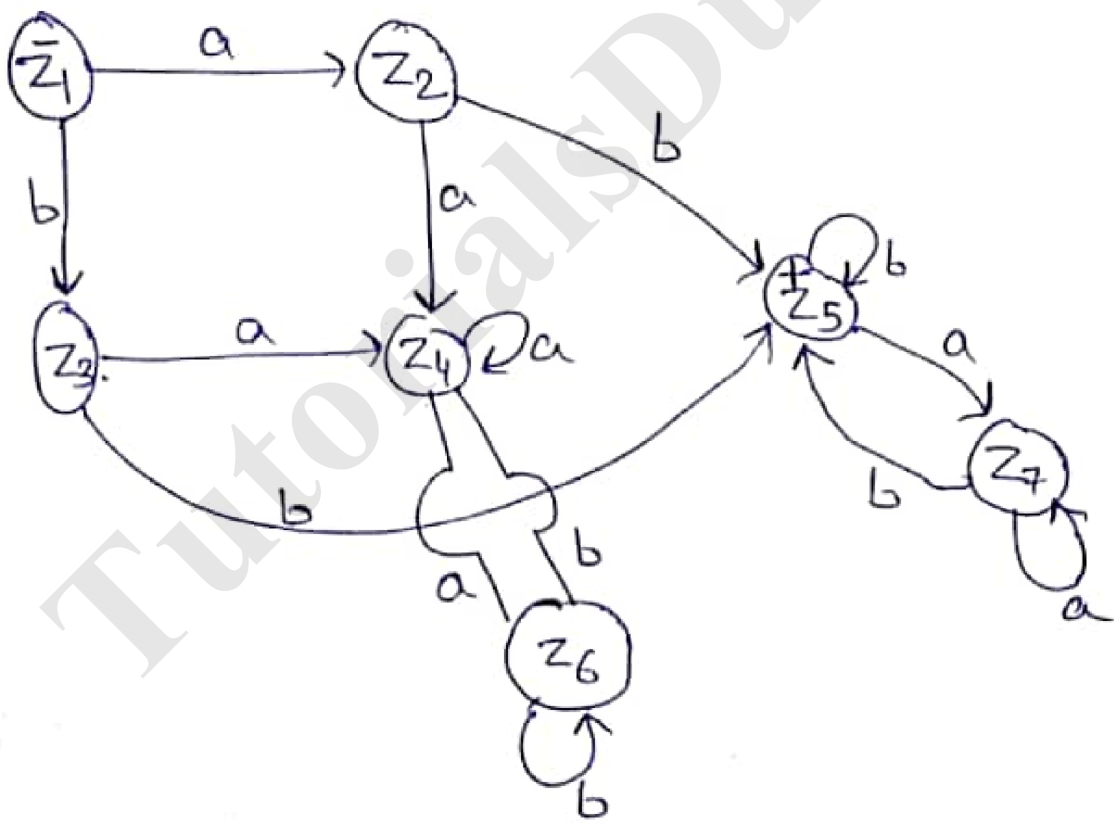
(Z_5) b at $Z_7 = x_2$ or y_4

Transition table

States	a	b
$\pm Z_1$	Z_2	Z_3
$+ Z_2$	Z_4	Z_5
$+ Z_3$	Z_4	Z_5
$+ Z_4$	Z_4	Z_6
Z_5	Z_7	Z_5
$+ Z_6$	Z_4	Z_6
$+ Z_7$	Z_7	Z_5



complement



2. Begin with the grammar:

$S \to ABC \mid BaB$

$A \to aA \mid BaC \mid aaa$

$B \to bBb \mid a \mid D$

$C \to CA \mid AC$

$D \to \epsilon$

(i) Elemant ϵ productions.

$S \to ABC \mid BaB \mid AC \mid a \mid Ba \mid aB$

$A \to aA \mid BaC \mid aaa \mid aC$

$B \to bBb \mid a \mid bb$

$C \to CA \mid AC$

(ii) Eliminate any unit production in the resulting grammar.

No unit productions cannot be derived.

(iii) Eliminate any useless symbols in the resulting grammar.

$S \to BaB \mid a \mid Ba \mid aB$

$A \to aA \mid aaa \mid aC$

$B \to bBb \mid a \mid bb$

(b) Using pumping lemma, prove that language $a^n b^n a^n$, $n >= 1$ is non-context free.

Sol. $w = uvxyz$

uv^3xy^2z

Cannot be in $\{a^n b^n a^n\}$ because,

All words in $\{a^n b^n a^n\}$ have exactly one occurence of the substring ab no matter what n is.

Now if either the v-part / y-part has the substring in it, uv^2xy^2z will have more than one substring of ab, so it cannot be in $\{a^n b^n a^n\}$.

Neither v nor y contains ab.

All words in $\{a^n b^n a^n\}$ have exactly one occurrence of the substring ba no matter what n is. Now if either the v-part or y-part has substring ba in it, then uv^2xy^2z has more than one such substring, which no word in $\{a^n b^n a^n\}$ does. Neither v nor y contain ba. But if v & y are blocks of one letter, then uv^2xy^2 has increased one or two clumps of solid letters (more a's if v is a's). But there are three clumps and they are not increased equally.

Ex. $a^{200}b^{200}a^{200} = a^{200}b^{70}b^{40}b^{90}a^{82}a^{3}a^{115}$

then

$$uv^2xy^2z \neq a^{200}b^{200}a^{200}$$
$$\neq a^n b^n a^n \text{ for any } n.$$

Therefore, $\{a^n b^n a^n\}$ is not a context-free L.

Q.3 (a) Prove that the regular languages are closed under complement.

Sol. If L is a regular language, we know from Kleene's theorem that there is some FA that accepts the language L. Some of the states of this FA are final states and, most likely, some are not. Let us reverse the final status of each state. If an input string formerly ended in a nonfinal state, it now ends in a final state.

and vice versa. This new machine we have built accepts all input strings that were not accepted by the original FA and rejects all the input strings that the FA used to accept. Therefore, this machine accepts exactly the language L'. So, by Kleene's theorem, L' is regular.

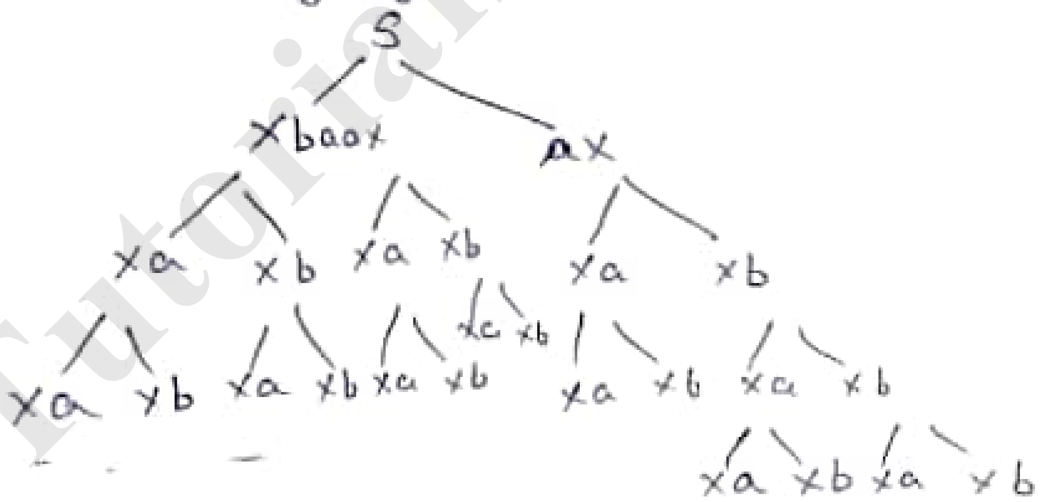
(b). Give a CFG for the language of all the words having 'bbb' as substring.

Sol $S \rightarrow X\,bbb\,X$

$X \rightarrow aX \mid bX \mid \Lambda$.

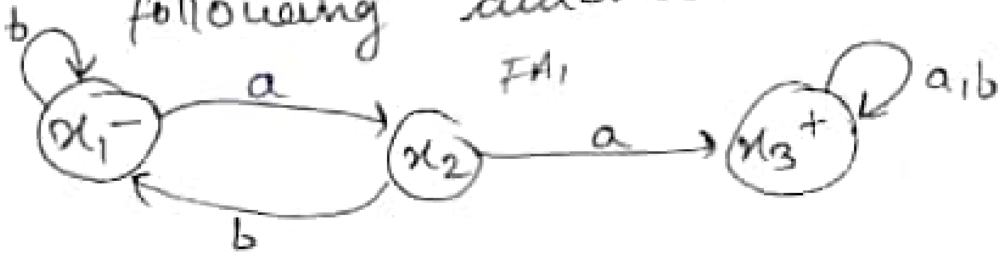
(c) Show that the following grammar is ambiguous
$S \rightarrow X\,baa\,X \mid aX \qquad X \rightarrow Xa \mid Xb \mid \Lambda$.

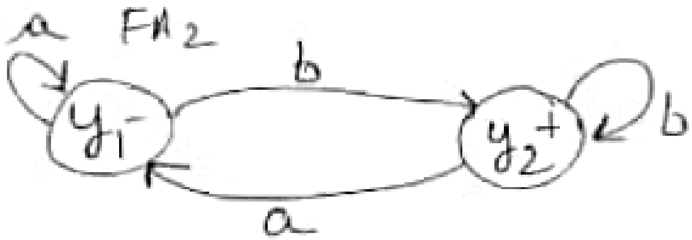
Sol. Total language tree



words like aabaa, abaa, etc. can be derived from both the syntax trees. Hence the grammar is ambiguous.

4 (a) Find FA1 · FA2 (concatenation) for the following automata





Sol. (−) $z_1 = x_1$

(z_2) a at $z_1 = x_2$

(z_1) b at $z_1 = x_1$

(z_3) a at $z_2 = x_3$ or y_1

(z_1) b at $z_2 = x_1$

(z_3) a at $z_3 = x_3$ or y_1

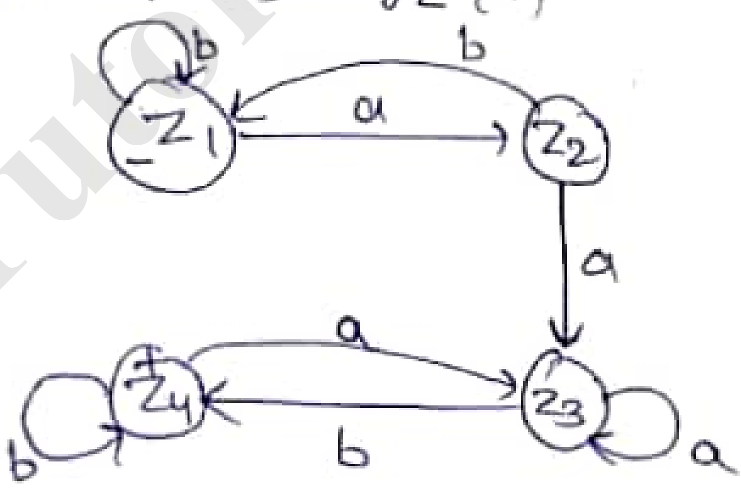
(z_4) b at $z_3 = x_3$ or y_2 (+)

(z_3) a at $z_4 = x_3$ or y_1

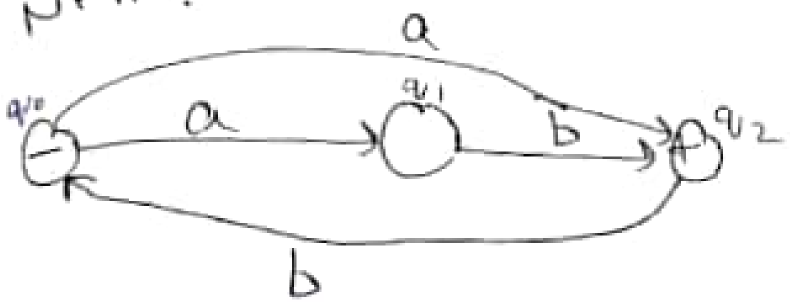
(z_4) b at $z_4 = x_3$ or y_2 (+)

Transition table

	a	b
(−) z_1	z_2	z_1
z_2	z_3	z_1
z_3	z_3	z_4
(+) z_4	z_3	z_4



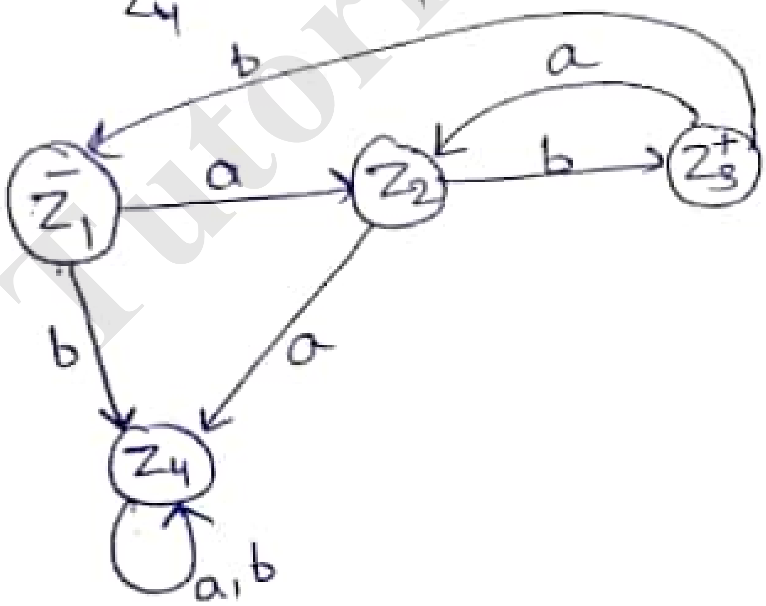
b) Find the equivalent DFA for the following NFA :-



	a	b
{q0}	{q1, q2}	φ
{q1, q2}	φ	{q0, q2}
{q0, q2}	{q1, q2}	{q0}
φ	φ	φ

Transition table

	a	b
→ z_1	z_2	z_4
z_2	z_4	z_3
(+) z_3	z_2	z_1
z_4	z_4	z_4



5(a) Describe the language for the following regular expression:-

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

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

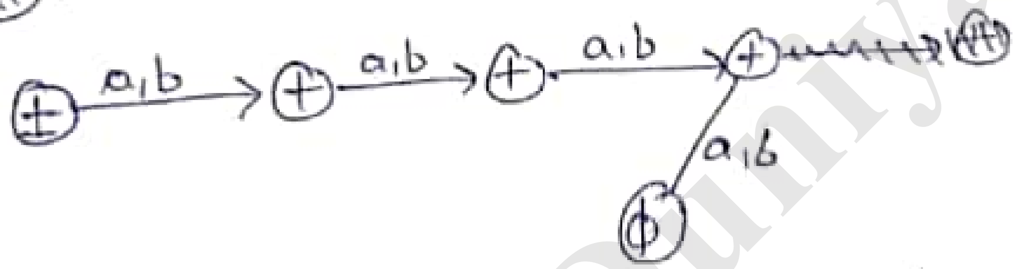
Sol. (i) All words with substring ab occuring atleast once.

(ii) All combination of a & b with even position letter always occupied by 'b'.

(b) Build a DFA with fewer than four letters.

~~(+)+++++~~

Sol.



(c) Give a regular expression for the language of all the words that do not have a double letter.

Sol. $(ab)^* + (ba)^* + \Lambda + a + b$

(a) Explain halting problem?

Sol. Suppose program P is written in C++, & input X of that prog P is also given. Then, we have a program,

$$halts(P, X) = \begin{cases} \text{return 'yes' if P halt on input } X. \\ \text{return 'no' if P does not halt (run forever) on input } X. \end{cases}$$

Now consider another program:-

diagonal (X)

a: if halts (X, X) then goto a else halt

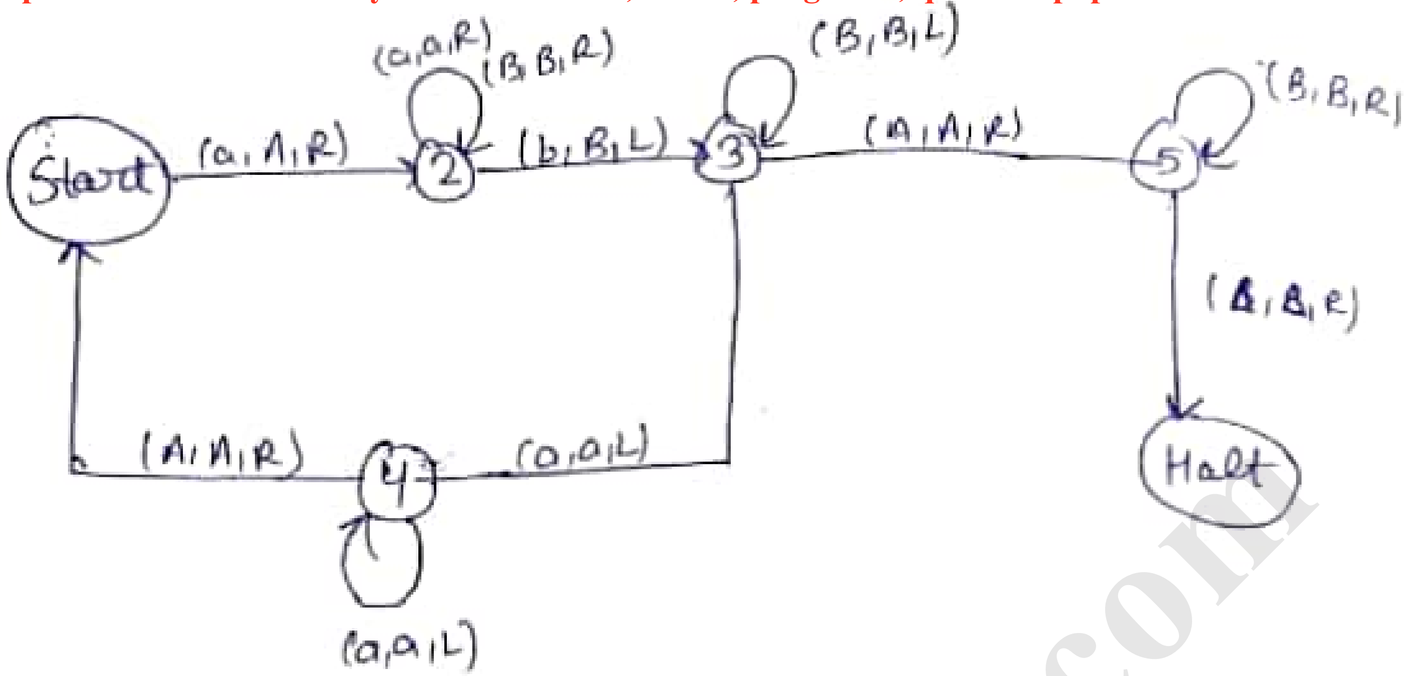
Here 'Halts' program decides that program X would halt if presented with itself as input, then diagonal (X) loops forever, otherwise it halts.

(b) Show that if language L is recursive, then L is recursively enumerable also.

Sol. Let $M = (K, \Sigma, \delta, s, H)$ be a T.M. let $\Sigma_0 \subseteq \Sigma - \{\sqcup, \triangleright\}$ be an alphabet, & let $L \subseteq \Sigma_0^*$ be a language. We say that M semidecides L if for any string $w \in \Sigma_0^*$, the following is true: $w \in L$ if and only if M halts on input 'w'.

A language L is recursively enumerable if and only if there is a TM 'M' that semidecides L.

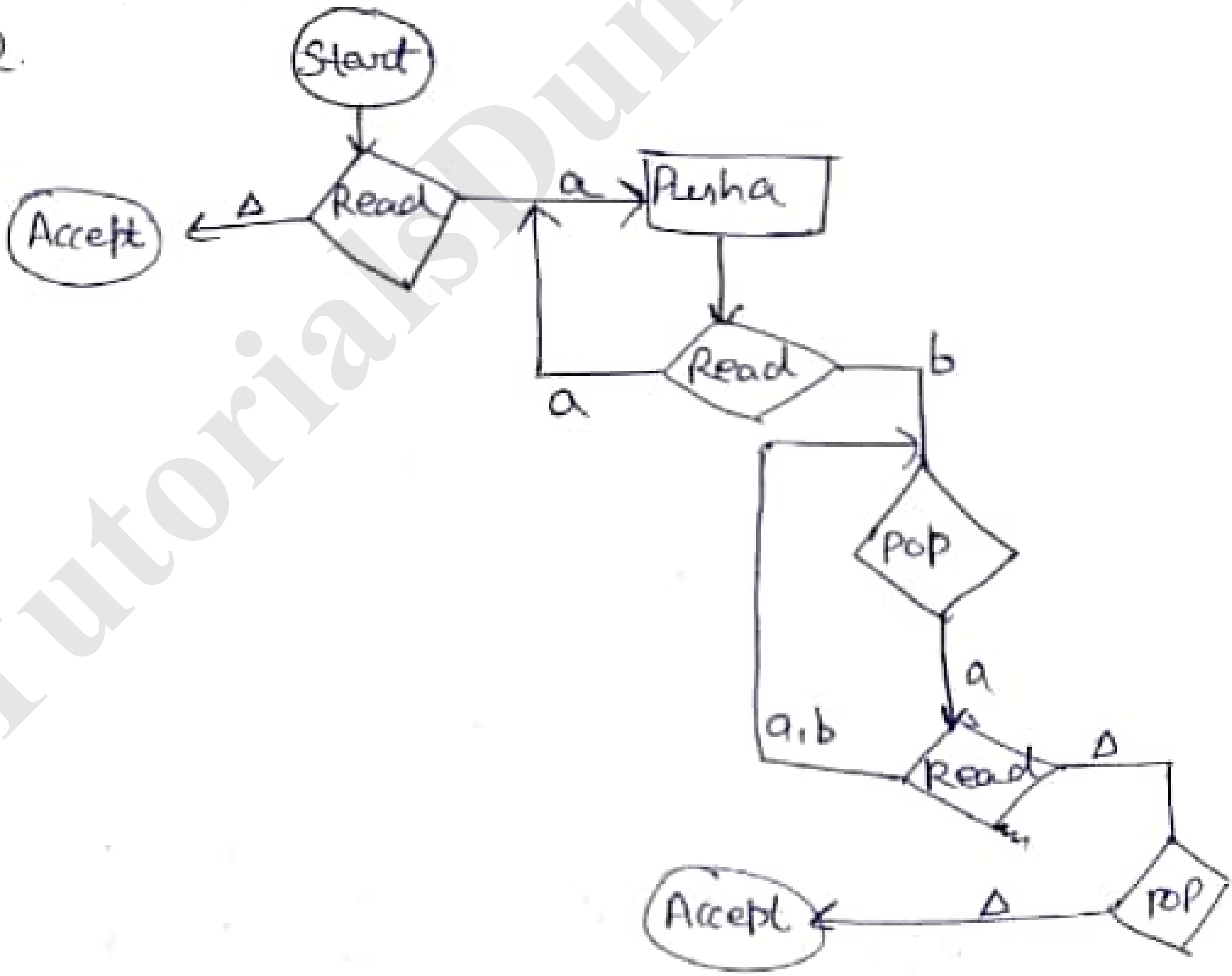
(c) Design a Turing Machine for $a^n b^n$ for $n >= 1$



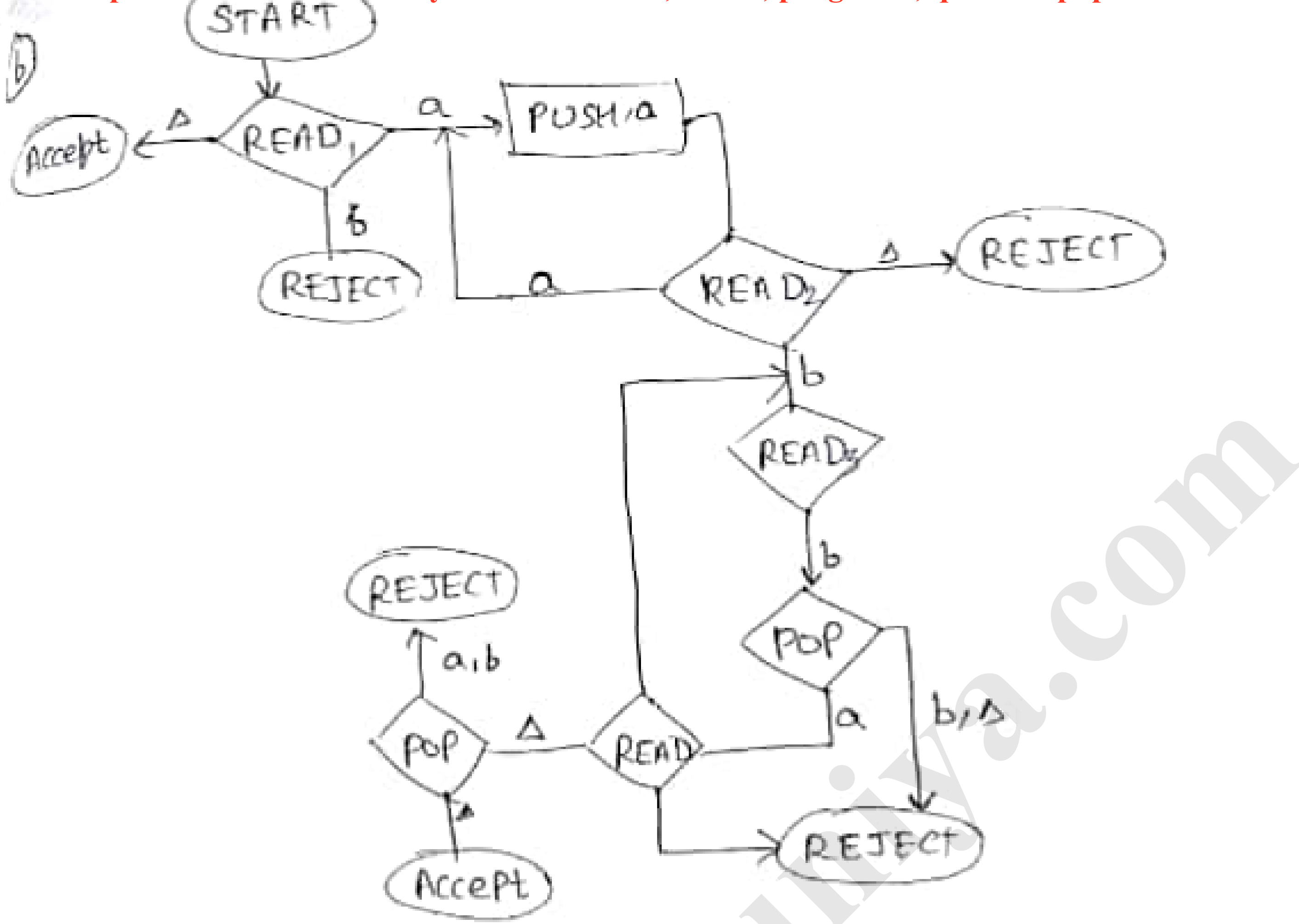
(7) Design a PDA for the following language:

$L = \{a^n S$, where S starts with length$(S) = n\}$

Sol.



(b)



(i) Define the language defined by above PDA.

~~where no b's are present after a's~~

Sol. $a^n b^{2n}$.

(ii) Trace the word 'abb' on the above PDA.

Sol. Firstly a is read and pushed on the stack, then b is read, and other next b is read, as then two b's are read on the tape, the pushed a is popped off the stack, and nothing is left on tape, hence a Δ is read and popped off and string is accepted.

$Read_1 \xrightarrow{a} Push\ a \longrightarrow$