Phase- 2

Question 16: Let $\Sigma = \{a, b\}$ for each $k \geq 1$, let C_k be language of all strings in it with exactly k places from right-hand end. Thus $C_k = \Sigma^* a \Sigma^{k-1}$. Describe an NFA with k+1 states which recognises C_k in terms of both state diagram & explaination.

Answer:- As the given language $C_k = \Sigma^* a \Sigma^{k-1}$

$\forall k \geq 1$ and $\Sigma = \{a, b\}$

Let N be NFA with k+1 states that recognises C_k

Then we can draw simply

for transition function $\delta(q_i, a)$ can be

As we no. of states, $\underset{k+1}{}$ $Q = \{q_0, q_1, \dots, q_k\}$

q_0 is start state over $\Sigma = \{a, b\}$ alphabets

final state set is $\{q_k\}$.

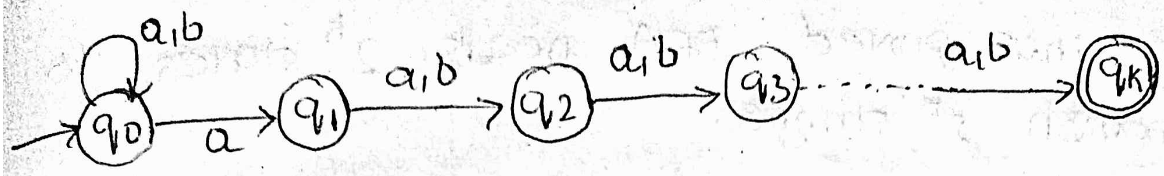
Then, $$\delta(q_i, a) = \begin{cases} \{q_0, q_1\} & \text{if } i = 0 \\ \{q_{i+1}\} & \text{if } 0 < i < k \\ \phi & \text{if } i = k \end{cases}$$

$$\delta(q_i, b) = \begin{cases} \{q_0\} & \text{if } = 0 \\ \{q_{i+1}\} & \text{if } 0 < i < k \\ \phi & \text{if } i = k \end{cases}$$

$$\delta(q_i, \epsilon) = \phi \quad \forall i$$

Thus NFA is defined $N = (Q, \Sigma, \delta, q_0, F)$
N

where all parameters are defined earlier. Then we have an NFA N



that recognises C_k.

Question 17: Considering $C_K = \Sigma^* a \Sigma^{K-1}$ that is all strings with K places with a's from right hand (same as previous q.). Prove that for each K, no DFA can recognise C_K with fewer than 2^K states.

Answer:- We know DFA is with no ambiguity in moving to next step. Thus when we say DFA can't exist that is with less than 2^K states to recognise C_K.

Then DFA enters two different states after reading two different strings.

Say two strings r, s where each are with same length K and same suffix z (rz, sz)

Thus $r = r_1 \cdots r_K$ and $s = s_1 \cdots s_K$

let they have position which differs i.e 'i' $r_i \neq s_i$

i.e $r_i = a$ $s_i = b$ Thus $r_i \neq s_i$

Now, let $z = b^{i-1}$

So, rz or sz string has K^{th} bit from end is 'a'

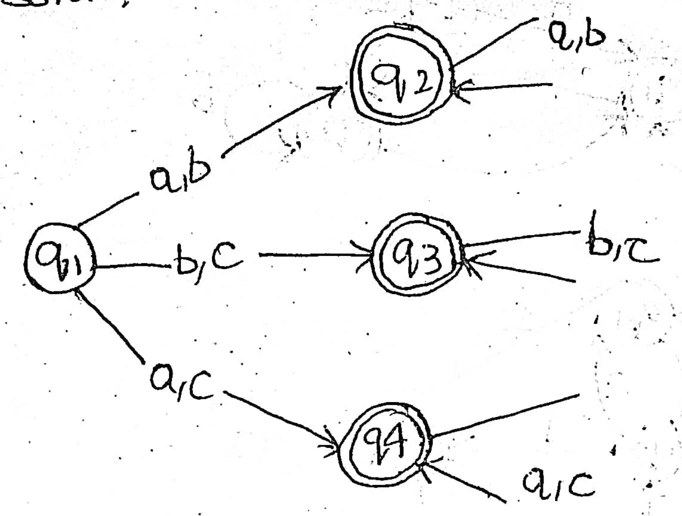
Problem 18:-

(a) Give a non-detrministic finite automata that caputres the regular expression $(a|b)^+|(a|c)^+|(b|c)^+$ in graphical (state diagram form.

(b) Prove the language $L = \{0^m 1^n \mid m \neq n\}$ is non-regular.

Answer:-

(a) One way to build NFA for this regular expression:-



(b) Given, language $L = \{0^m 1^n \mid m \neq n\}$

let's prove by contradiction say L is regular

then claim:- $\bar{L}$ (L's complement) is regular

proof:- we know if L is regular (assume)

$\bar{L}$ is regular (regular languages are closed under complement)

Thus, $\bar{L} = \begin{cases} 0^m 1^m \mid m = n \\ \text{not in form of } 0^m 1^n \text{ strings} \end{cases}$

Now, $\bar{L} \cap 0^* 1^*$ be a language.

$\bar{L} \cap 0^* 1^* = \{0^m 1^n \mid m = n\}$

$\Rightarrow \bar{L} \cap 0^* 1^* = \{0^n 1^n \mid n \geq 0\}$

Thus $\bar{L} \cap 0^* 1^*$ becomes non-regular ($\because$ Pumping lemma)

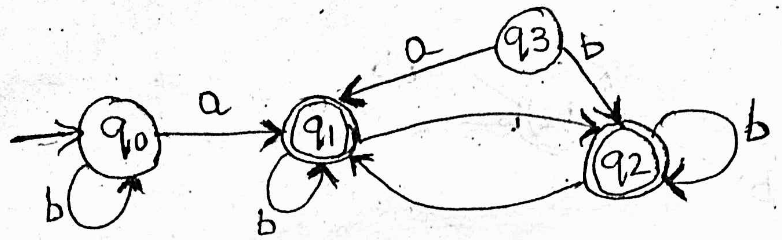
which contradicts $\bar{L}$ is regular

Thus, also contradicts L is regular, so, L_2 is non-regular

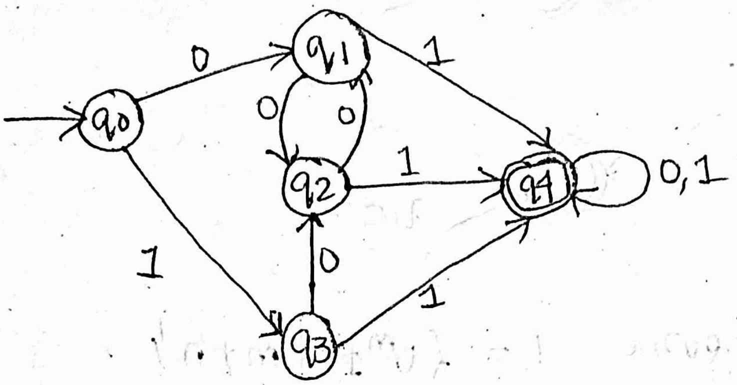
Problem 19:

Minimize the given DFA's.

(a)

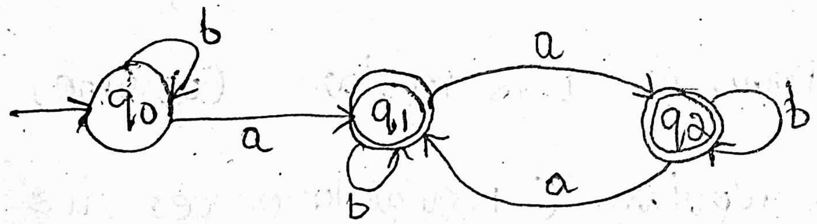


(b)



Answer:-

(a) As q_3 is inaccessible from initial state,



(eliminated inaccessible state and associate edges)

	a	b
→ q_0	$*q_1$	q_0
$*q_1$	$*q_2$	$*q_1$
$*q_2$	$*q_1$	$*q_2$

thus, transitions are

$\delta(q_0, a) = \{*q_1\}$

$\delta(q_0, b) = \{q_0\}$

$\delta(*q_1, a) = \{*q_2\}$

$\delta(*q_1, b) = \{*q_1\}$

$\delta(*q_2, a) = \{*q_1\}$

$\delta(*q_2, b) = \{*q_2\}$

Now, using Equivalence theorem

that is, variable $P_0 = \{q_0\}\ \{q_1, q_2\}$

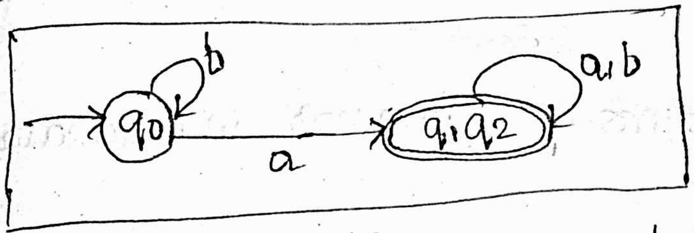
$P_1 = \{q_0\}\ \{q_1, q_2\}$

that is P_i is set with non-final, final states

Thus as P_1 is also same ($\because$ P_0 is itself distinguishable)

$P_0 = P_1$

Then, from P_1, q_1, q_2 are equivalent and merged together to form minimal DFA.



Thus minimal DFA is obtained

(b) As everything is accesible from initial states and no dead states too. Then, occured transistion states / functions are:-

$\delta(q_0, 0) = q_1$ $\delta(q_1, 0) = q_2$

$\delta(q_0, 1) = q_3$ $\delta(q_1, 1) = {}^*q_4$

$\delta(q_2, 0) = q_1$ $\delta(q_3, 0) = q_2$

$\delta(q_2, 1) = {}^*q_4$ $\delta(q_3, 1) = {}^*q_4$

$\delta(q_4, 0) = {}^*q_4$ $\delta(q_4, 1) = {}^*q_4$

From equivalent theorem

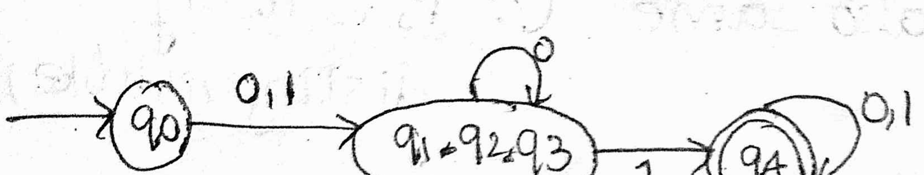
$P_0 = \{q_0, q_1, q_2, q_3\} \{q_4\}$ $P_1 = \{q_0\} \{q_1, q_2, q_3\} \{q_4\}$

$P_2 = \{q_0\} \{q_1, q_2, q_3\} \{q_4\}$

So, as $P_1 = P_2$

from P_2 q_1, q_2, q_3 are equivalent and can be merged

Thus minimal DFA is:-



Problem 20

Let $\Sigma = \{1, \#\}$ and lot. $Y = \{w | w = x_1 \# x_2 \# \ldots \# x_k$ for $k \geq 0$, each $x_i \in 1^*$ and $x_i \neq x_j$ for $i \neq j\}$. Prove Y is not regular.

Answer from given,

Y accepts the strings in $x_1 \# x_2 \# \ldots \# x_k$ form where $x_1 x_2 \cdots x_k$ are substrings formed with any no. of 1's.

x_i is any no. of 1's ($\because x_i \in 1^*$)

We know that,

pumping lemma has three conditions and any language satisfied those called to be regular.

Proving on contradiction. Say Y is regular then,

let s be string with less than p length ('p' is pumping length)

Then s is divided into three. Say $S = uvw$

Here, $s = x_1 \# x_2$ for $k = 2$ and $x_1 \neq x_2$

Consider $x_1 = 1^p 1$ and $x_2 = 11 1^p$ ($\because x_1, x_2$ can be any no. of 1's)

Now observe they are different substrings and depends on p value. ($x_1 = 111$, $x_2 = 1111$ for $p = 2$)

Thus, length of s is longer than p and $s \in Y$.

Let 111#1111 be string $\in$ V. The pumping length of string is 2.

111#1111 into three parts ($\because$ pumping lemma condition)

those xyz.

$x = 1 \quad v = 1 \quad w = 1\#.1111$

$S = 111\#1111$

$= \frac{1}{u} \cdot \frac{1}{v} = \frac{1\#\ 1111}{w}$

$S = (1).(1)^i\ (1\#\ 1111)$

$= \frac{1}{u} = \frac{11}{v} = \frac{1\#\ 1111}{w}$ (when $i=2$)

String after pumping is $1111\#1111 \in Y$.

(Since substring $x_1 = x_2$)

Thus, violated the def of language Y. Thus pumping lemma condition.

Y is not a regular language.

Problem 21

As we know about types of grammar in automata that is :- $\alpha \rightarrow \beta$

type 0 Unrestricted	type 1 Context-Sensitive
$\lvert\alpha\rvert \geq 1$ $(V+T)^* V (V+T)^* \rightarrow (V+T)^*$	should be type 0 $\lvert\alpha\rvert \leq \lvert\beta\rvert$

type 2 Context-free	type 3 Regular
should be type 1 $\lvert\alpha\rvert = 1$ (variable)	most restricted $V \rightarrow VT/T$ (or) $V \rightarrow TV/T$ form (extended too)

Thus, answer following

(a) $G = (N, T, P, S)$

$N = \{S, A, B\}$ $T = \{a, b, c\}$ $P: S \rightarrow aSa$

Write a grammar which is type 2 but not type 3.

(b) Given CFG. Derive a grammar for $S \Rightarrow aB$ which is neither leftmost nor rightmost derivation

CFG :- $S \rightarrow aB$ $S \rightarrow bA$

$B \rightarrow b$ $A \rightarrow a$

$B \rightarrow bS$ $A \rightarrow aS$

$B \rightarrow aBB$ $A \rightarrow bAA$

(c) Derive a grammar for given language that to be type 0 but not type 1.

Answer:-

(a) we know, a grammar to be type 2 it should have left-hand with 1 variable.

Thus, $S \rightarrow aAa$

$A \rightarrow bB$

$B \rightarrow bB$

$B \rightarrow c$

It is type 2 but not type 3 ($\because$ type 3 is $V \rightarrow VT \mid T$ form)

(b) $S \Rightarrow aB$ is deried to :-

$\Rightarrow aaBB$ ($\because B \rightarrow aBB$)

$\Rightarrow aaBb$ ($\because B \rightarrow b$)

$\Rightarrow aabSb$ ($\because B \rightarrow bS$)

$\Rightarrow aabbAb$ ($\because S \rightarrow bA$)

$\Rightarrow aabbab$ ($\because A \rightarrow a$)

Thus terminal state is arrived

(c)

$G = (N, T, P, S)$

$N = \{S, A, B, C, D, E\}$ $T = \{a, b, c\}$ $P: S \rightarrow ABCD$

(c) $BCD \rightarrow DE$

$D \rightarrow aD$

$D \rightarrow a$

$E \to bE$

$E \to c$

Thus, it is type 0 since $|\alpha| \geq 1$

but not type 1 since $BCD \to DE$ violates

$|\alpha| \leq |\beta|$ rule.

Question 22: Prove that for each $n > 0$, a language B_n exists where

(a) B_n is recognizable by an NFA that has n states, and

(b) If $B_n = A_1 \cup A_2 \cdots \cup A_k$, for regular language A_i then atleast one of A_i requires a DFA with exponentially many states.

Answer:-

~~let prove this by contradiction~~

(a)

Given $n > 0$ B_n exists. Now, to prove

Claim:- B_n is recognizable by an NFA with n states.

Base case:- $n=1$

$B_n = \{\epsilon, 0, 1\}$

NFA $N = (\{q_0\}, \Sigma, \delta, q_0, \{q_0\})$

with $\delta(q_0, \epsilon|0|1) = q_0$ (single state acceptance)

Induction:-

Let B_n into two regular expressions

L, M of k_1, k_2 length $k_1, k_2 < n$ $k_1 + k_2 = n$

from hypothesis,

NFA's accepting L, M has atleast k_1, k_2 states.

And as B_n is regular expression - we know it is closed under concatenation, union, star operation.

Thus, B_n is recognizable by NFA with n states.

(b) Given, $B_n = A_1 \cup A_2 \cup \cdots A_k$.

for A is regular, If DAFA is constructed which is equalent to DFA of NFA (given)

Then, we know DFA has atmost 2^n states and atleast n states. Thus equivalent DFA too.

So, every language is recognized by a DFA.

corresponding DFA $\forall A_i$.

As one state that there is atleast one DFA which require 2^i states. (pigeon hole)

To recognize a language among all the A_i.

thus exponentially many states (2^i). DFA has atleast one of A_i.

Problem 23:-

Let $\Sigma = \{0,1\}$ and let

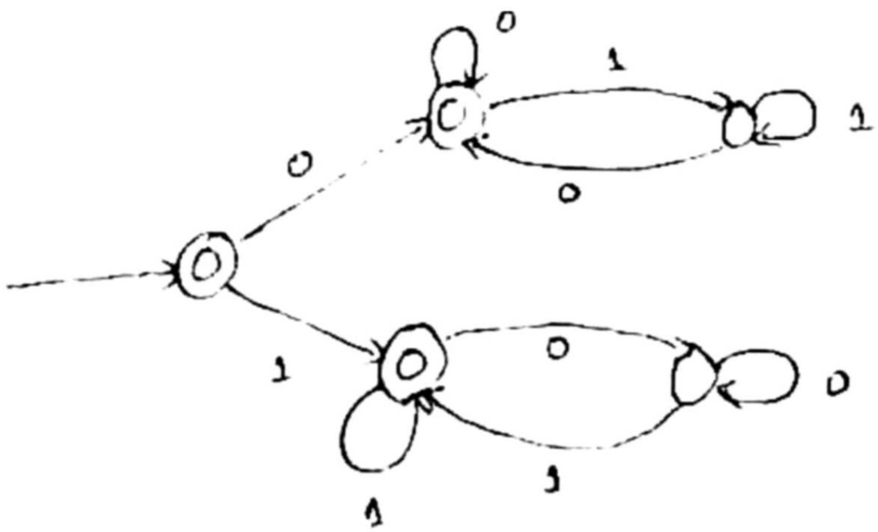
D = {w|w contains an equal number of occurance of the substring 01 and 10}

Thus 101 ∈ D because 1010 contains two 10's and one 01. Show that D is a regular language.

Solution:-

language D = {w|w contains equal no of occurances of substring 01 & 10} $\Sigma = \{0,1\}$

We know D is regular when D is identified by a DFA. Following DFA recognizes the language D.



The DFA recognize the language D

Thus, D is regular

Question ~~18~~ 24:

(a) Let $A = \{0^k u 0^k \mid k \geq 1$ and $u \in \Sigma^*\}$. Show that A is regular.

Answer:- Given, language $A = \{0^k u 0^k \mid k \geq 1 \,\&\, u \in \Sigma^*\}$

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

We know that, p be minimum pumping length

and another string $s = 0^k u 0^k$ then $p = (2k + |u|)$

as we know string which is not equal to zero length

can be splitted to xyz.

Where $|xy| \leq p$ for $x = 0^k, y = u, z = 0^k$.

pumping lemma also satisfies three conditions.

① $u = \epsilon$ then $0^k 0^k \in A$

$s = xy^i z$

$s = 0^k \epsilon 0^k \quad (i=0)$

$= 0^k 0^k \in A$

So, $\forall i = 0$ S is specified language.

(2) $S = xyz$ then $x = z = 0^k$.

$(|x| + |y| + |z|) \leq p$

thus, $k + |y| + k \leq p$ then $|y| \leq p - 2k$

So, $0^k 1 0^k \in A$ $(y = 1)$

S is specified. Thus $\forall y \in \Sigma^*$.

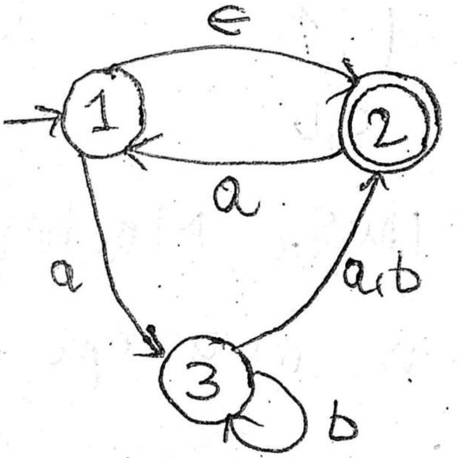
(3) $S = xy^{p-2k}z$

then, $S = 0^k y^{p-2k} 0^k$ $y^{p-2k} \in \Sigma^*$

So, all says $A \in$ regular as $S \in$ specified language. A satisfied pumping lemma. Then, A is regular.

Question 25:

From a lemma "If a language is regular, then it is described by regular expression". Convert the following NFA to equivalent DFA.



Answer $Q' = \{ \phi, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\} \}$

$(\because Q' \subseteq Q)$

Also as this is ε-NFA, we have the following set E where $E(R) = \{ q \mid q$ is reached from R on moving along ε arrows$\}$

Thus we have:- $R \subseteq Q$

$E(\phi) = \phi$	$E(\{1,2\}) = \{1,2\}$
$E(\{1\}) = \{1,2\}$	$E(\{2,3\}) = \{2,3\}$
$E(\{2\}) = \{2\}$	$E(\{1,3\}) = \{1,2,3\}$
$E(\{3\}) = \{3\}$	$E(\{1,2,3\}) = \{1,2,3\}$

$\delta'(R,a) = \{q \in Q \mid q \in \delta(r,a), \text{ for some } r \in R\}$

$\delta'(\phi,a) = \phi$

$\delta'(\phi,b) = \phi$

$\delta'(\{1\},b) = E(\delta(1,b))$
$= E(\phi) = \phi$

$\delta'(\{1\},a) = E(\delta(1,a))$
$= E(\{3\}) = \{3\}$

$\delta'(\{2\},a) = E(\delta(2,a))$
$= E(\{1\})$
$= \{1,2\}$

$\delta'(\{2\},b) = E(\delta(2,b))$
$= \phi$

$\delta'(\{3\},a) = E(\{2\})$
$= \{2\}$

$\delta'(\{3\},b) = E(\delta(3,b))$
$= E(\{2,3\})$
$= \{2,3\}$

8

$\delta'(\{1,2\},b) = E(\delta(1,b)) \cup E(\delta(2,b))$

$\Rightarrow \phi \cup \phi = \phi$

$\delta'(\{1,3\},a) = E(\{3\}) \cup E(\{2\})$
$= \{2,3\}$

$\delta'(\{1,3\},b) = E(\{\phi\}) \cup E(2,3)$
$= \{2,3\}$

$\delta'(\{2,3\},a) = E(\{1\}) \cup E(\{2\})$
$= \{1,2\}$

$\delta'(\{2,3\},b) = E(\phi) \cup E(\{2,3\})$
$= \{2,3\}$

$\delta'(\{1,2,3\},a) = E(\{3\}) \cup E(\{1\}) \cup E(\{2\})$

$\Rightarrow \{3\} \cup \{1,2\} \cup \{2\} = \{1,2,3\}$

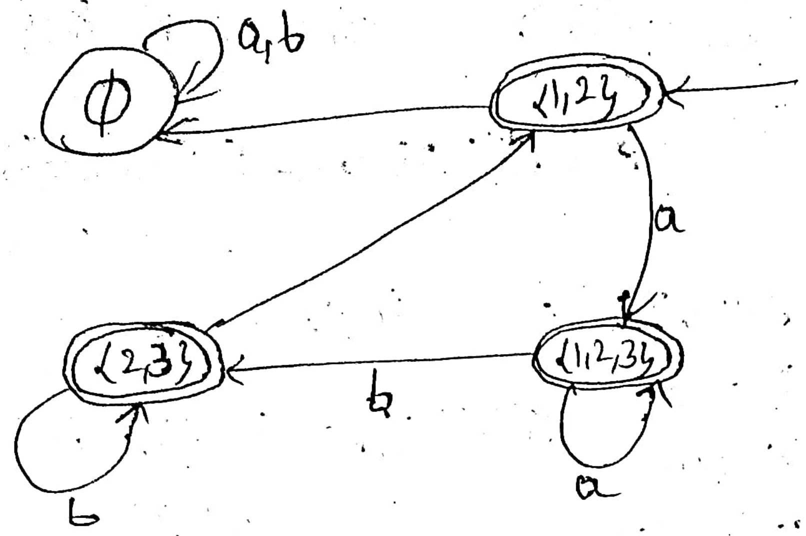
$\delta'(\{1,2\},a) = E(\delta(1,a)) \cup E(\delta(2,a))$
$= \{3\} \cup \{1,2\}$
$= \{1,2,3\}$

lly $\delta'(\{1,2,3\},b)$
$= \{2,3\}$

Thus start state can be $q'_0 = E(q_0) = E(\{1\}) = \{1,2\}$

Thus Equivalent DFA in simplified way

is :-



Question 26:- Answer following context free grammar G is:-

$R \to XRX \mid S$

$S \to aTb \mid bTa$

$T \to XTX \mid X \mid \epsilon$

$X \to a \mid b$

(a) Variables of G are:-
from grammer we can say non-terminal symbols are. R, S, T, X

(b) Terminals are a, b

(c) State variables of G is R (since top most left hand var)

(d) Three strings in L(G):-

String 1	String 2	String 3
$R \to S$	$R \to S$	$R \to S$
$R \to aTb$ ($\because S \to aTb$)	$R \to aTb$	$R \to bTa$ ($\because S \to bTa$)
$R \to aXb$ ($\because T \to X$)	$R \to a\epsilon b$	$R \to bXa$ ($\because T \to X$)
$R \to abb$ ($\because X \to b$)	$R \to ab$	$R \to baa$ ($\because X \to a$)

Thus, abb, ab, baa are three strings in L(G)

(e) Not in L(G) string are :- aba, a, ε
since no obtainable on observing the grammar.

(f) $T \xRightarrow{*} aba$ is true to obtain

(g) So, the description in English of L(G) is It contains the strings created by a,b terminal symbols and not any kind of palindromes.

Question 27:- Give a context-free grammar that generates the following. (1) In all parts, the alphabet Σ is $\{0,1\}$.

(a) $\{w \mid w$ starts and ends with the same symbol$\}$ is given by ?

(b) $\{w \mid$ the length of w is odd and its middle symbol is a $0\}$

(c) $\{w \mid w = w^R$, w is a palindrome$\}$

(2) ~~Give a CG~~

(g) $A = \{a^i b^j c^k \mid i = j$ or $j = k$ where $i, j, k \geq 0\}$

Is your grammar ambigous ? Why or why not?

(1) a. Answer:- $S \rightarrow 0P0 \mid 1P1 \mid 0 \mid 1$

$P \rightarrow 0P \mid 1P \mid \varepsilon$

(1) b $S \rightarrow 0 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1$

(1) c $S \rightarrow 0 \mid 1 \mid 0S0 \mid 1S1 \mid \varepsilon$

(2) Yes, here ambigouity rises:

Since given language can be splitted into two :- $A_1 = \{a^i b^j c^k \mid i = j,\ i,j,k \geq 0\}$

$A_2 = \{a^i b^j c^k \mid j = k,\ i,j,k \geq 0\}$

Thus, A_1 value of i,j are equal ; equal no. of a's, b's in the language A_1.

So, $S \to S_1 \mid S_2$ ($\because$ A is union of grammar of two languages)

A_1 context-free gramer	A_2 Context-free-grammar
$S_1 \to S_1 c \mid E \mid \varepsilon$	$S_2 \to a S_2 \mid F \mid \varepsilon$
$E \to aEb \mid \varepsilon$	$F \to bFc \mid \varepsilon$

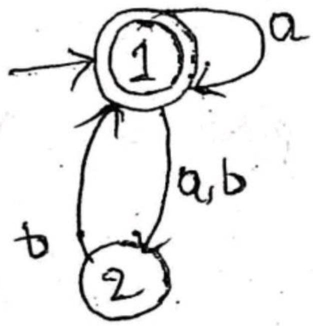
So, for generating any string $w = a^n b^n c^n$ using A either S_1 or S_2 can be used.

Therefore, context-free grammar of A language is ambiguous.

Problem 28:-

From a lemma "If a language is regular, then it is described by regular expression". Convert the following NFA to equivalent DFA.

(a)



Answer:- $Q' = P(Q)$

which is subset of all set of Q (NFA).

Now, from this $Q' = \{\phi, \{1\}, \{2\}, \{1,2\}\}$

Transistion function δ'

Alphabet set Σ ; $b, a \in \Sigma$

Now, an element P in Q'.

$\delta'(P, a) = \{q \in Q \mid q \in \delta(r, a) . r \in R\}$

$\delta'(\phi, a) = \delta(\phi, a) = \phi$

$\delta'(\phi, b) = \delta'(\phi, b) = \phi$

$\delta'(\{1\},a) = \delta(1,a) = \{1,2\}$

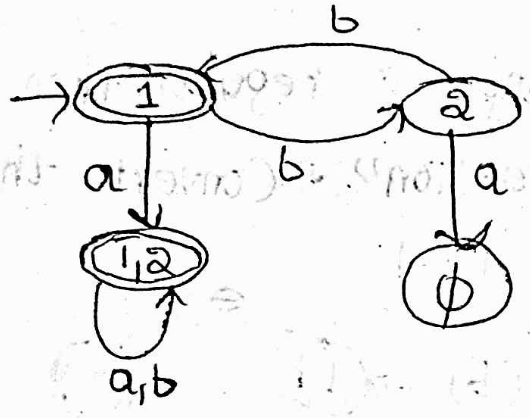
$\delta'(\{2\},a) = \delta(2,a) = \phi$ $\quad \delta'(\{2\},b) = \delta(2,b) = \{1\}$

$\delta'(\{1,2\},a) = \delta(\{1,2\},a) = \{1,2\} \cup \phi = \{1,2\}$

$\delta'(\{1,2\},b) = \delta(\{1,2\},b)$

$= \delta((1,b) \cup (2,b)) = \{2\} \cup \{1\} = \{1,2\}$

Here $q_0' = \{1\}$ $\quad (\because q_0' = \{q_0\})$



[Since $F' = \{R \in Q' \mid R$ has accepts state of NFA$\}$]

Thus, Machine M accepts the possible state where NFA is in accept state.

Problem 29:-

Consider the language $L = \{ w | w$ has even number of a's and one a two b's $\}$. The language L is the intersection of two simpler languages say L_1 and L_2.

Solution

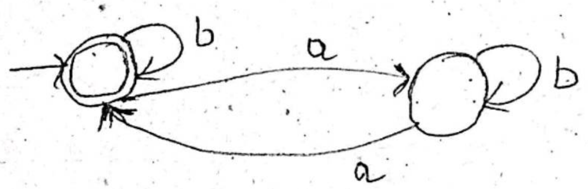
$L_1 = \{ w | w$ even no. of a's $\}$

$L_2 = \{ w | w$ one or two b's $\}$

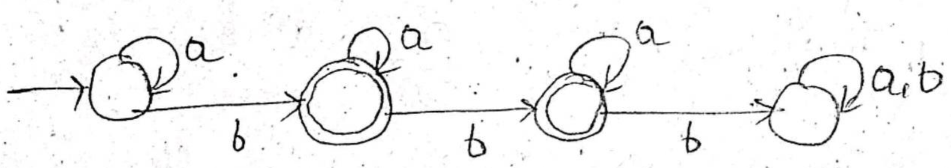
L be recognised DFA M

L_1, L_2 be recognised by DFAs M_1, M_2

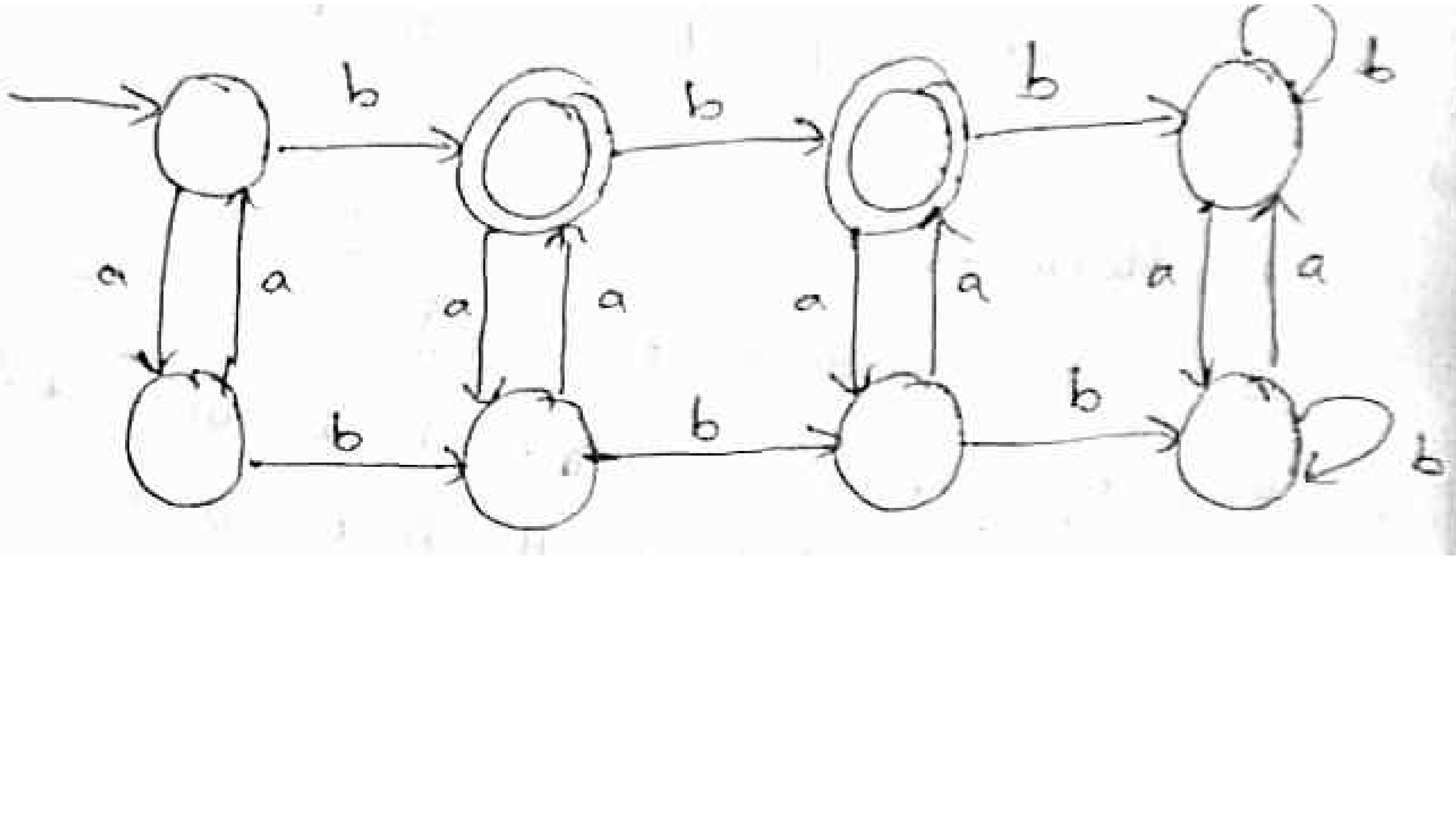
state diagram of M_1 that recognises L_1 is



state diagram of M_2 that recognises L_2 is



M accepts string if and only if both M_1, M_2 accept it (since L is intersection of L_1, L_2)


b
b
b
b
a
a
a
a
a
a
a
a
b
b
b
b

Problem 30:-

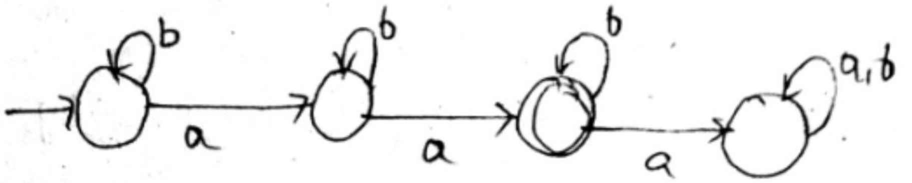
L = {w | w has at excatly two a's and atleast two b's}. The language L is intersection of two simpler lang. So, construct DFA for this.

Now L₁ = {w | w has excatly two a's}

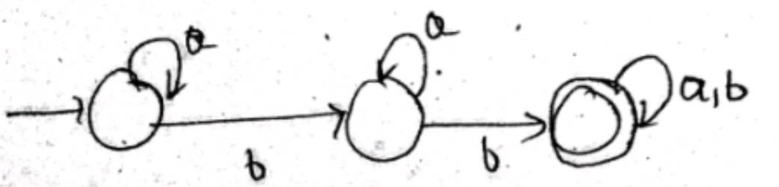
L₂ = {w | w has atleast two b's}

M be DFA that recognises L and M₁ M₂ be DFA's which recognises L₁, L₂.

state diagram of M₁

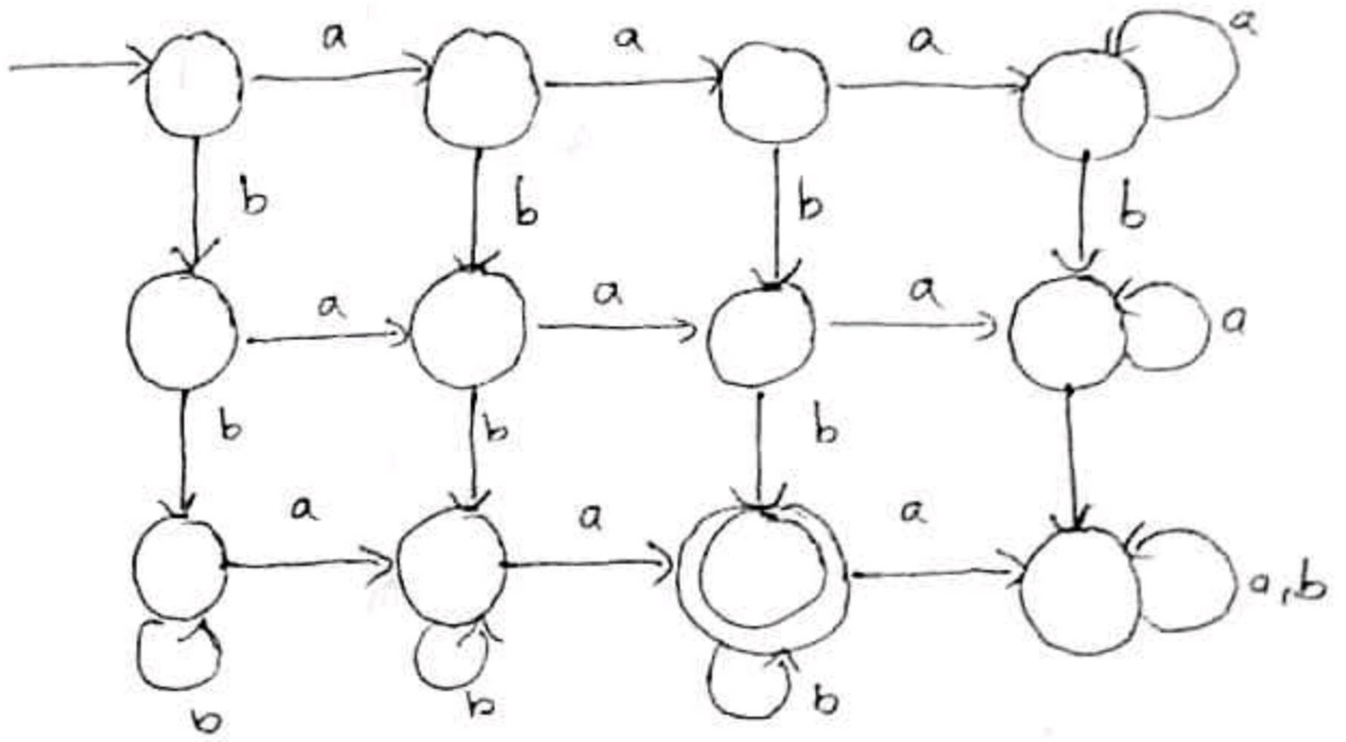


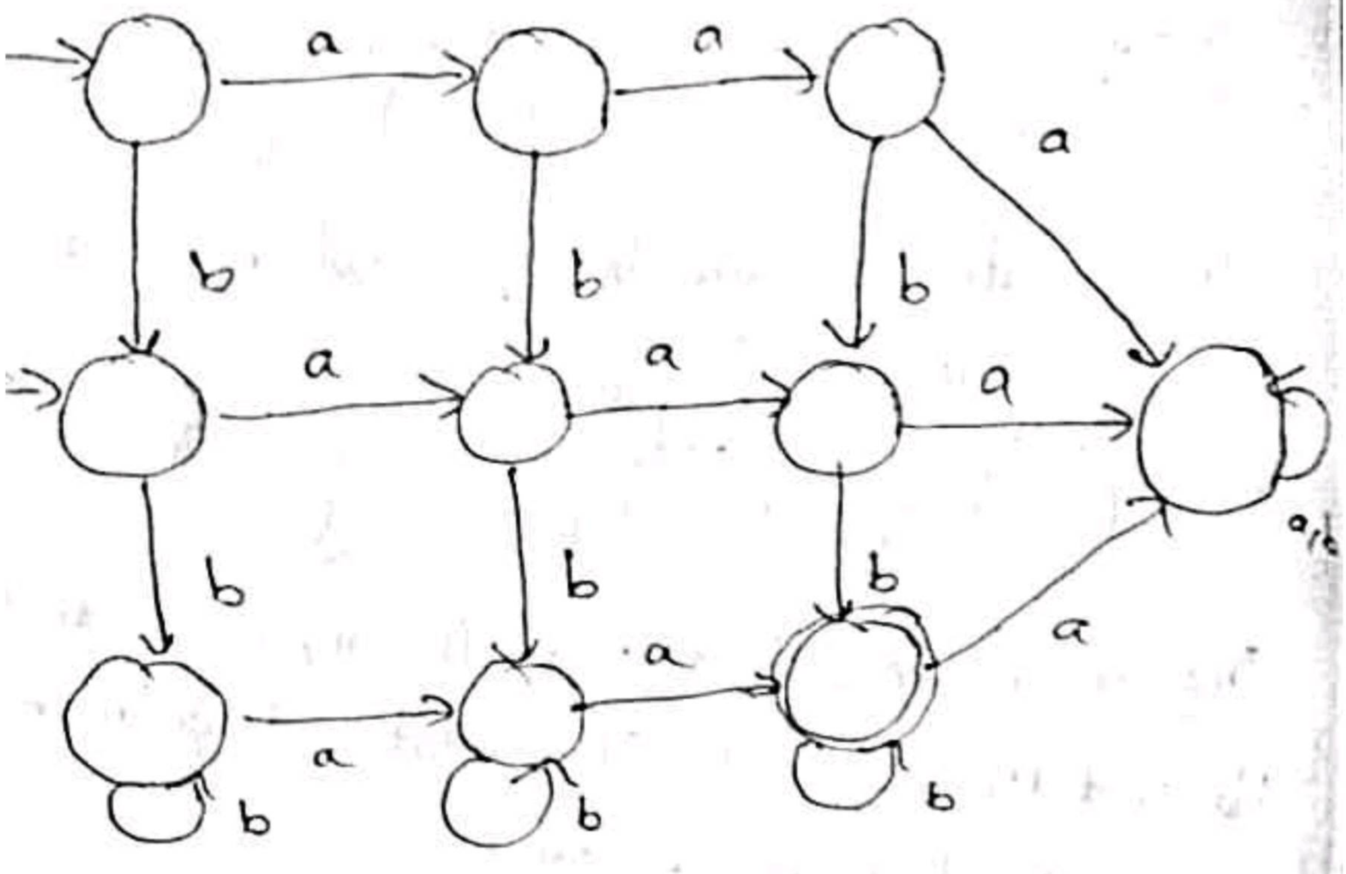
M₂ :-



machine M will accept if both M₁ M₂ accep it.

state diagram of M is :-


a
a
a
a
b
b
b
b
a
a
a
a
b
b
b
a
a
a
a,b
b
b
b


a
a
a
b
b
b
a
a
a
b
b
b
a
a
a
b
b
b

Problem 31:-

let $\Sigma = \{0, 1, +, -\}$ and

ADD = $\{x = y+z \mid x, y, z$ are binary integers, and x is the sum of y and $z\}$

show that ADD is not regular.

On contradiction, assume ADD is a regular lang.

Consider a string with form $1^{p+1} = 10^p + 1^p \in ADD$

(Here p is pumping length)

ADD here, say $S = 111 = 100 + 11$

$$= \underset{u}{1}\ \underset{v}{1}\ \underset{w}{1 = 100+11}$$

pump the middle part uv^iw $(i \geq 0)$

if $i = 2$ $\quad V = 11$

$$S = (1)\ (1)^i\ (1 = 100 + 11)$$

$$= \underset{u}{1}\ \underset{V}{11}\ \underset{w}{1 = 100+11}$$

string after pumping is $111 = 100 + 11 \in ADD$

(Since addition of two binary 100, 11 is not equal to the binary number 1111)

But contradicts, pumping lemma condition is violated.

ADD is not regular language.

—*—

Problem 32 :- Give CFG that generate

(a) { w | w contains at least three 1s } , Σ {0,1}

Ans :- CFG generated can be $N \rightarrow P1P1P1P$

$P \rightarrow 0P \mid 1P \mid \varepsilon$

(b) { w | the length of w is odd }

Ans :- Here CFG can be :-

$Q \rightarrow 0 \mid 1 \mid 00P \mid 01P \mid 10P \mid 11P$

$Q \rightarrow 0 \mid 1 \mid 0P0 \mid 0P1 \mid 1P0 \mid 1P1$

Problem 33 :- Let C be a context-free language and R be a regular language. Prove that $C \cap R$ language is context-free.

Ans :- Let say P is PDA that recognises C

D is DFA that recognises R.

Let P' be PDA which have set of states of P, D.

then P' has track of D and do as P.

Thus we have a PDA that recognises $C \cap R$ accepts word s if and only if it stops a state q that belongs to $F_P \times F_D$ (∵ states accepts by P, D)

then $C \cap R$ is recognised by P'

thus, $C \cap R$ is context-free.

Problem 34:-

Construct the equivalent Pushdown Automata (PDA) to the following context free grammar (refer ques 26)

$R \rightarrow XRX \mid S$ $\qquad$ $T \rightarrow XTX \mid X \mid \varepsilon$

$S \rightarrow aTb \mid bTa$ $\qquad$ $X \rightarrow a \mid b$

Answer:-

From lemma:- A language is context-free if and only if some pushdown automata recognizes it.

let the machine M (pushdown automata) recognize context-free language.

we know CFG has one variable (non-terminal state) at left hand side.

$\varepsilon, R \rightarrow XRX$ $\qquad$ $\varepsilon, T \rightarrow \varepsilon$

$\varepsilon, R \rightarrow S$ $\qquad$ $\varepsilon, X \rightarrow a$

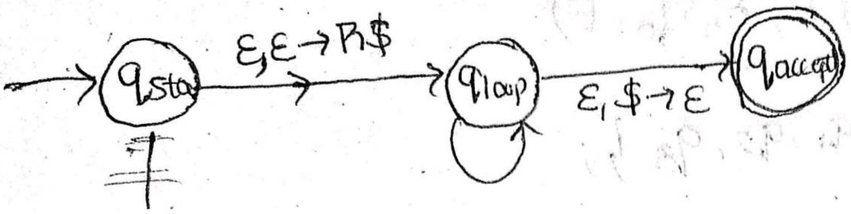
$\varepsilon, S \rightarrow aTb$ $\qquad$ $\varepsilon, X \rightarrow b$

$\varepsilon, S \rightarrow bTa$ $\qquad$ $a, a \rightarrow \varepsilon$

$\varepsilon, T \rightarrow XTX$ $\qquad$ $b, b \rightarrow \varepsilon$

$\varepsilon, T \rightarrow X$

So, equivalent PDA is:-



Problem35:- Let $\Sigma = \{0,1\}$ and let B be collection of strings that contain atleast one 1 in their second half. In other words, $B = \{uv \mid u \in \Sigma^*, v \in \Sigma^* 1 \Sigma^*$ and $|u| \geq |v|\}$.

(a) Give a PDA that recognizes B.

(b) Give a CFG that generates B.

Answer:-

Here to construct PDA that recognises B firstly 'u' string is pushed on reading related alphabets that is either 0's or 1's into stack.

Now, machine matches the letters in stack with letters in v.

Here, the bridge between u to v is solved on allowing $ empty string reading to next (V)

Conditions:- $|u| \geq |v|$

no. of 1's in input atleast 1 in second half, accept entire string.

Thus, PDA machine M that recognizes B

is $(Q, \Sigma\frac{1}{2}, \Gamma, \delta, q_0, F)$.

$Q = \{q_0, q_1, q_2, q_3\}$,

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

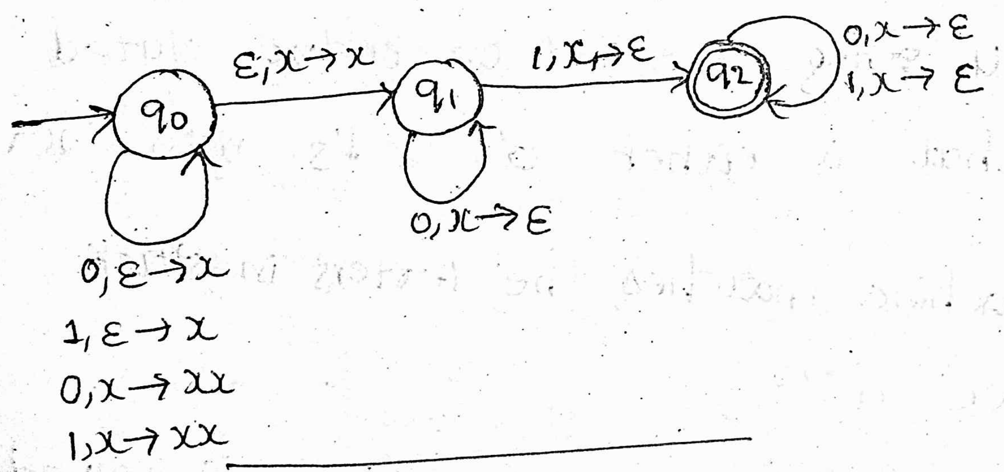
$\Gamma = \{x\}$

$F = \{q_3\}$

Thus on Non derteministic push down automata

δ is =

Input	0		1		ε	
Stack	x	ε	x	ε	x	ε
q_0	$\{(q_0, xx)\}$	$\{(q_0, x)\}$	$\{(q_0, xx)\}$	$\{(q_0, x)\}$	$\{(q_1, x)\}$	
q_1	$\{(q_1, \varepsilon)\}$		$\{(q_2, \varepsilon)\}$			
q_2	$\{(q_2, \varepsilon)\}$		$\{(q_2, \varepsilon)\}$			



(b) So, for this a contex-free grammar is language B

as $B = UV$ we have $N \to UV$

u can have any no of one's so,

$U \rightarrow XY$

v which is second half has one atleast one 1 so,

$V \rightarrow X|X| X|Y |X|U | Y|U | U|U$

X is not restricted with no of zeroes but atleast one

$X \rightarrow 00^*|\varepsilon$

Y is atleast one 1 or more than one

$Y \rightarrow 11^*|\varepsilon$

Thus CFG that generates B is obtained.

Question 36. - Given $C = \{x\#y \mid x, y \in \{0,1\}^*$ and $x \neq y\}$. Prove C is a context-free language

Solution So, from given, C is a string that in form $x\#y$ and x, y are different letters.

So, the possible context-free grammar is

$N \rightarrow A\#P \mid P\#A$

$A \rightarrow SAS \mid 0$

$P \rightarrow SAS \mid 1$

$S \rightarrow 0 \mid 1$

Hence the language defined in terms of CFG, the language is CFL. Hence proved.

Question 37:- There is language $L = \{w \mid w$ has even number of a's and each a is followed by atleast one $b\}$. Construct DFA.

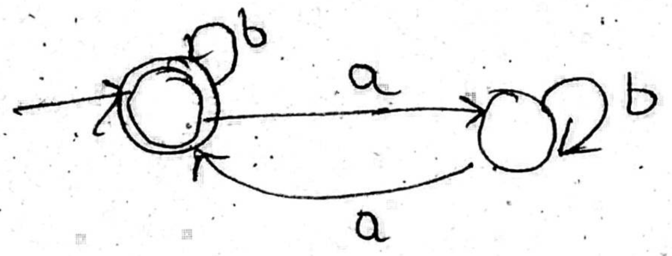
Answer:-

M_1, M_2 be DFA's recognizes L_1, L_2 where

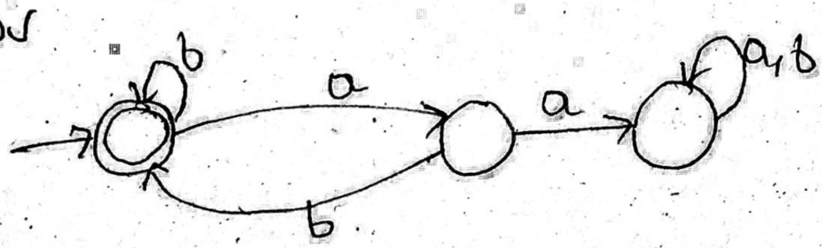
$L_1 = \{w \mid w$ has even no of a's$\}$

$L_2 = \{w \mid w$ has each a is a followed by atleast $b\}$

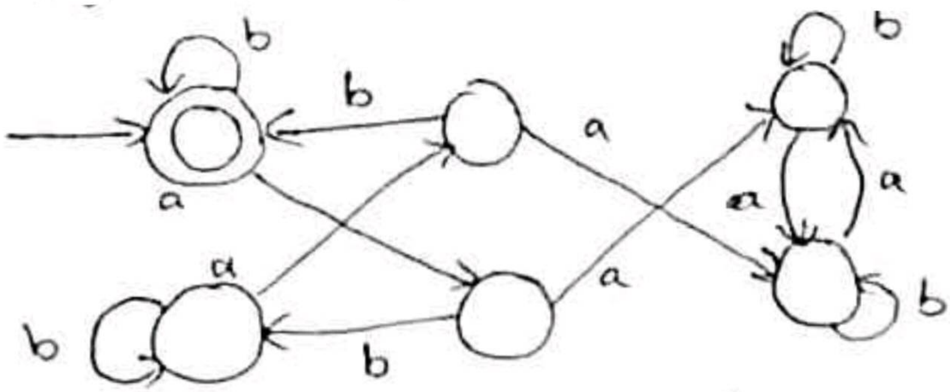
Now, state diagram of M_1 that recognizes L_1 is as follows where $L_1 = \{w \mid w$ has even no of a's$\}$



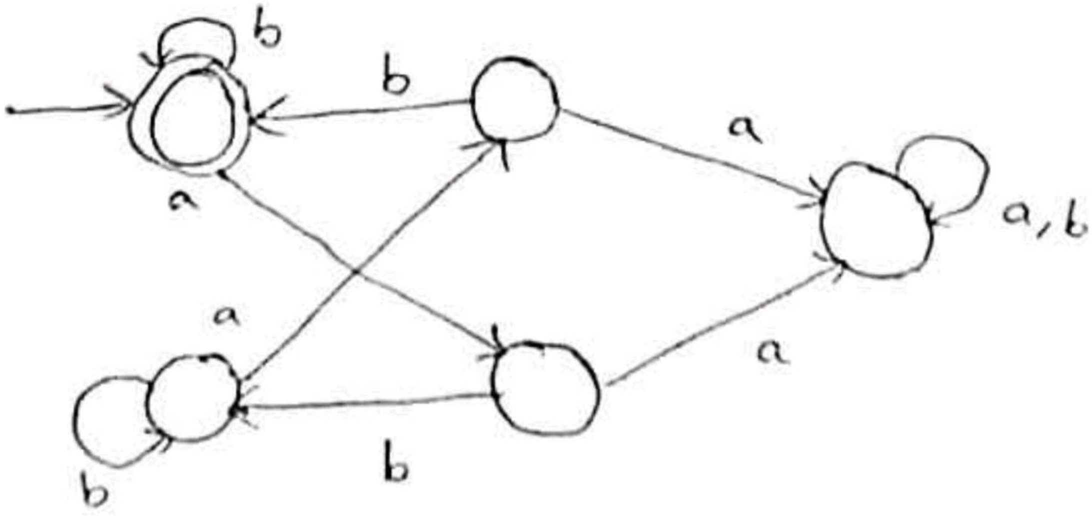
state diagram of M_2 that recognises L_2 is as follows



The machine M will accept the input if and only if both M_1 & M_2 accept it. The state diagram of M that recognizes L is :-



Combine some states to get more simplified form as follows



Problem 38:- Using pumping lemma, show the following $\{0^n 1^n 0^n 1^n \mid n \geq 0\}$ is not context free lang.

Solution:- As, $L = \{0^n 1^n 0^n 1^n \mid n \geq 0\}$

let s be a string where it is splitted into form (using pumping lemma).

Now, show s can't be pumped thus, proves L is not CFL, since s is from L.

$s = 0^p 1^p 0^p 1^p$

Now, observe, $s = uvxyz$ where

$s = uv^2xy^2z$ ($\because$ v, y are same type of alphabets)

then, $s \notin L$ ($\because$ symbols order is missed)

Thus, on violating pumping lemma condition.

L is proved to not a CFL.

Problem 39:- Show that every DCFG in an unambigous CFG.

Solution:- As, CFG is proper superset of DCFG.

Thus CFG is used to generate DCFL ($\because$ CFG is derived from DFA)

We know that,

Claim:- DCFG's are always shows an unambigous behaviour and an unambogous CFG is super

Class of DCFG's:

Proof:- let say a PDA P

equivalent CFG C

Then languages recognised by P are be

L\$ =) C generate L\$

But P deterministic $\Rightarrow$ C is an unambiguos.

Thus, replacing special character with empty ε

we generate L. $(G \Rightarrow G)$

Therefore, proved the given statement every DCFG is an unambigous CFG.

Probelem 40:- Describe the DPDA that recognises the language of G, where G is the following grammar

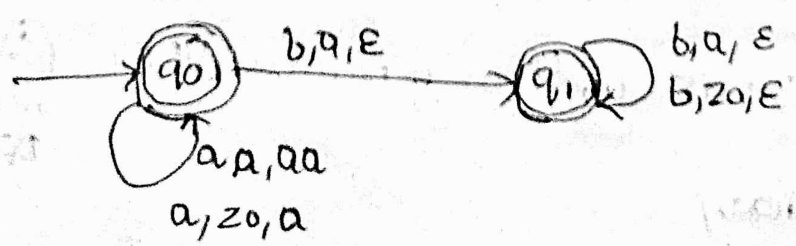
$P \rightarrow Q$

$Q \rightarrow QaQb \mid QbQa \mid \varepsilon$

Answer:- from given ~~CF~~ Grammar,

terminal states a, b non-terminal P, Q

Now, DPDA that recognises L(G) is



Problem 41:- Prove the language L is not DCFL, where $L = \{a^i b^j c^k \mid i,j,k \geq 0 \text{ and } i=j \text{ or } i=k\}$

Solution:-

From given,

Say L is Context Free language

also L is DCFL then there exists $\bar{L} = \{a,b,c\}^* - L$ is dCFL.

Then $L_0 = \{a^i b^j c^k \mid i \neq j \ \& \ j \neq k\}$ is not CFL

Here, $\bar{L} \cap a^* b^* c^*$ be a language equal to L_0

Then, it concludes as $\bar{L}$ is DCFL, then L_0 is DCFL, thus L_0 is CFL.

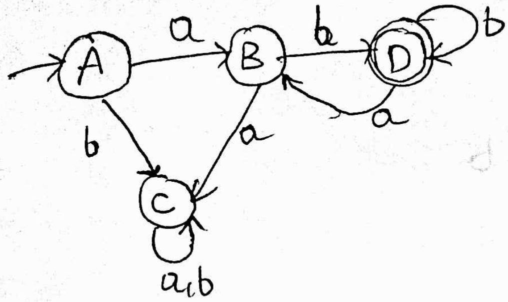
But it contradicts L_0 is not CFL. therefore, L is not DCFL.

Problem 42:- Construct a DFA for the language L, where $L = \{w \mid w$ is any string not in $(ab+1)^*\}$

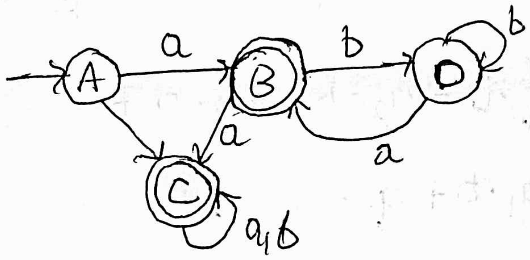
Ans:-

Now, L complement be $\bar{L} = \{w \mid w$ string not in $(ab^*)^*\}$

DFA that recognised L is:-



Now, DFA that recognises 'I' is :-

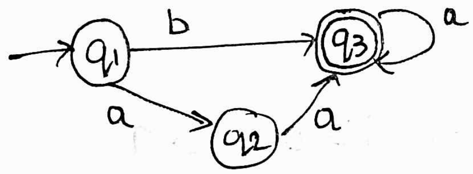


—*—

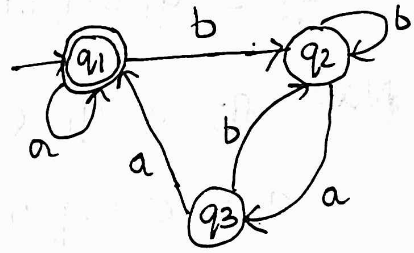
Question 43:-

Find the regular expression of following DFA's using Arden's Theorem :-

(a)



(b)



Answer:-

(a) Firstly, write the equation to reach each state

$q_1 = \epsilon$ (∵ initial state)

$q_2 = q_1 \cdot a$ (transistion function to reach q_2)

$q_3 = q_1 \cdot b + q_2 \cdot a + q_3 \cdot a$

Now, $P = S + PR$ is the form to get regular expression

Thus, $q_2 = \epsilon \cdot a$ (∵ substituting q_1 in q_2)

$= a$

$q_3 = q_1 \cdot b + q_2 \cdot a + q_3 \cdot a$

$= \cancel{\epsilon} \cdot b + a \cdot a + q_3 \cdot a$

$= (b + a \cdot a) + q_3 \cdot a$

Thus, $(b+aa)a^*$ is the regular expression for given DFA.

(b) $q_1 = \epsilon + q_1 \cdot a + q_3 \cdot a$

$q_2 = q_1 \cdot b + q_2 \cdot b + q_3 \cdot b$

$q_3 = q_2 \cdot a$

Thus, on substituting the q_3 value in q_2 and

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

$= q_1 \cdot b + q_2 \cdot (b + a \cdot b)$

$= q_1 \cdot b \cdot (b + ab)^*$ (using Arden's theorem)

Now, q_2 in q_3 gives,

$q_3 = q_1 \cdot b \cdot (b + a \cdot b)^* \cdot a$

Now, q_3 in q_1 gives

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

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

Then, using Arden's theorem q_1 can be

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

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

Thus, DFA his regular expression $(a + b \cdot (b + ab)^* aa)^*$

Question 44:- Give a CFG generating the language of string with twice as many a's and b's. Prove that your grammar is correct. $\Sigma = \{a, b\}$

Answer:-

So, from given we can clearly say the frequency of a's, b's matters most than their order.

Possible CFG for this language is:-

$N \rightarrow Paab \mid aPab \mid aaPb \mid aabP \mid Paba \mid aPba \mid abPa \mid abaP \mid Pbaa \mid bPaa \mid baPa \mid baaP$

$P \rightarrow N \mid \varepsilon$

Proof:- We can prove CFG is correct using induction

say a string $s \in \{aab, aba, baa\}$ (set of small strings)

Then, $f_a(s) = 2 f_b(s)$ ($\because f_i$ is frequency of i in s)

Here any small string obeys this, Thus we can write as follows:- $f_a(s_n) = 2 f_b(s_n)$ — ① ($\because n$ is length of string)

Now, if we can prove this for $n+1$ then obtained grammar is correct.

Claim:- $f_a(s_{n+1}) = 2 f_b(s_{n+1})$

So, s_{n+1} can be 0 a's and 0 b's (or) 2 a's and 1 b's.

If ε i.e 0 a's & 0 b's are inserted then,

$$f_a(s_{n+1}) = f_a(s_n) + f_a(\varepsilon)$$
$$= f_a(s_n) \quad (\because f_a(\varepsilon) = 0)$$

$$f_b(s_{n+1}) = f_b(s_n) + f_b(\varepsilon)$$
$$= f_b(s_n)$$

Now, $f_a(s_{n+1}) = 2 f_b(s_{n+1})$ (in ①)

If 2 a's 1 b is inserted

$f_a(s_{n+1}) = f_a(s_n) + 2$ | $f_b(s_{n+1}) = f_b(s_n) + 1$

Here s is aab (or) aba (or) baa

$$f_a(s_{n+1}) = 2 f_b(s_n) + 2 \quad (\text{from ①})$$
$$= 2(f_b(s_n) + 1)$$

So then from $f_b(S_{n+1}) = f_b(S_n)+1$

we have now, $f_a(S_{n+1}) = 2 f_b(S_{n+1})$

Therefore both cases proved $f_a(S_{n+1}) = 2 f_b(S_{n+1})$

Then obtained grammar is correct.

Problem 45:- Show that class of DCFLs is not closed under '$*$' operation.

Solution:- If given language $L = \{d\} \cup \{d\} L_1 \cup L_2$

where $L_1 = \{a^i b^j c^k : i, j, k \geq 0, i \neq j\}$
$L_2 = \{a^i b^j c^k : i, j, k \geq 0, i \neq k\}$ > DCFLs

Then, $L^* \cap \{d\}^* (\{a\}^* \{b\}^* \{c\}^* - \{\wedge\}) = \{d\} L_1 \cup \{d\} L_2$

Here, if L^* is DCFL then $\{d\} L_1 \cup \{d\} L_2$ should be in DCFL. But here it is not followed.

Thus, DCFL is not closed under $(*)$.

Problem 46:- Class of DCFLs is not closed under concatenation. Prove this.

Solution:- say $L_3 = \{d\} L_1 \cup L_2$ where

$L_1 = \{a^i b^j c^k : i, j, k \geq 0, i \neq j\}$
$L_2 = \{a^i b^j c^k : i, j, k \geq 0, i \neq k\}$ > DCFLs

Now, L_3 and $\{d\}^*$ are DCFL. To show $\{d\}^* L$ is not DCFL.

Consider, $\{d\}^* L_3 \cap \{d\} \{a\}^* \{b\}^* \{c\}^* = \{d\} L_1 \cup \{d\} L_2$

Then, $\{d\} L_1 \cup \{d\} L_2$ is DCFL, if $\{d\}^* L$ is DCFL

Here but say S words and K languages then are DCFL when $\{S\} K$ is DCFL.

But as $L_1 \cup L_2$ is not DCFL ($\because$ complement contradiction)

Then $\{d\}(L_1 \cup L_2)$ is not DCFL.

Hence proved DCFL is not closed under concatenation.

Problem 47 :- Let CUT of language $CUT(A) = \{yxz \mid xyz \in A\}$ show class of CFL is not closed under CUT.

Ans :- CFL has few defined context free languages one of them cyclic shift of CFL.

Here proving that doesn't happen proves CUT is not.

Say, three strings L_1, L_2, L_3 be x, y, z

Assume, $L_1 \circ L_2$ exist in CFL then it must be cyclic shift too. say. $s = xyz \in CUT(A)$

That means concatenation $X \& Y \& Z$, $x \in X$, $y \in Y$, $z \in Z$

But $CUT(A)$ defined as $yxz \in A$ & $xyz \in A$

So closed under cyclic shift means $xy \in A$ $yx \in A$

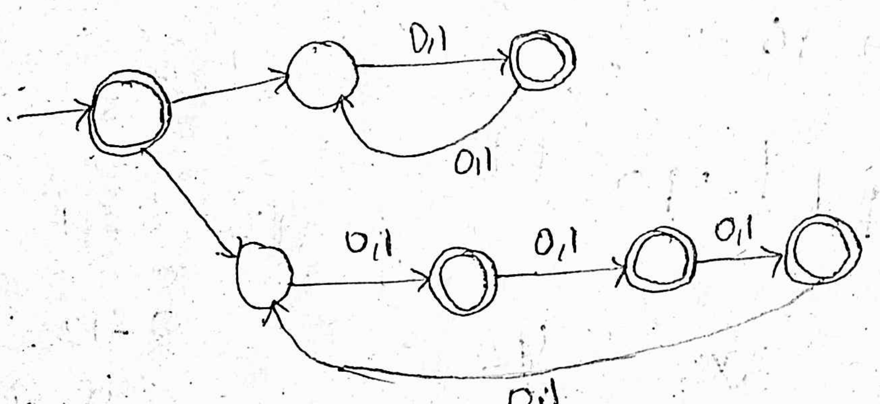
But contradicts. Thus proved.

Problem 48:- N be NFA with K states that recognises language A, and $\bar{A}$ is its complementary language. Now show that if $\bar{A}$ is nonempty, $\bar{A}$ contains some string of length at most K is not necessarily true (through eg)

Solution:- Assume, $\Sigma = \{0,1\}$, N be NFA with $k = 8$ states

Then, A be language recognized by N.

State diagram of N can be



which says ~~a~~ empty string is accepted by N.

for any nonempty string s, N will reject s if and only if length s is divisible by 2 and 5.

Then $\bar{A}$ consists of all non empty strings of length divisible by 10.

shortest in $\bar{A}$ has $10 > k$ length and $\bar{A}$ is nonempty.

Hence $\bar{A}$ contradicts fact $\bar{A}$ is nonempty, $\bar{A}$ contains some string of length at most K.

Thus, not necessarily true.

Problem 49:- Let B be any language over alphabet Σ. Prove that $B = B^*$ iff $BB \subseteq B$.

Case(a) If $B = B^*$ then $BB \subseteq B$.

Since for every language $BB \subseteq B^*$ then we can get $BB \subseteq B$ $(\because B = B^*)$

proved $BB \subseteq B$ iff $B = B^*$

case(b) If $BB \subseteq B$ then $B = B^*$

we know $\boxed{BB \subseteq B^*}$ for every language.

Assume s be string, that $s \in B^*$ then $s \in B$

If $s \in B^*$ then s can be split into $x_1, x_2, x_3, \ldots, x_k$

where $s = x_1, x_2, \ldots, x_k$ for $x_i \in B$, $k \geq 1$.

$BB \in B$ $(\because x_1, x_2 \in B)$

^{lly} $BB \subseteq B$ $(\because x_3 \in B$ Thus, $x_1, x_2, x_3 \in B)$

continued the B. $s \in B$ $(\because x_1 x_2 x_3 \ldots x_k \in B)$

So, If $(x_1, x_2 \ldots x_k) \in B$ then string $s \in B$.

Thus $\boxed{B^* \subseteq B}$

So, $B = B^*$

proved $B = B^*$ iff $BB \subseteq B$

Thus whole proved for any language B, $B = B^*$ iff $BB \subseteq B$.

*

~~Prove~~ Question 50:- Let A be an infinite regular language. Prove A can be split into two infinite disjoint regular subsets.

Answer:- To prove this, assume string w where $w \in A$ and $w = xyz$ (x, y, z substrings of w)

We know from pumping lemma,

$w = xy^i z$ ($i \geq 0$) $\in A$ ($\because A$ is regular)

If say $A_1 = \{xy^{2i}z, i \geq 0\}$ then $xy^{2i}z$ must be in A ($\because xy^i z \in A$). Then says $A_1 \subseteq A$

So, the regex is $x(yy)^*z$

Clearly A_1 is infinite language ($\because i \geq 0$)

Assume $A_2 = \bar{A}_1 \cap A$ ($\because$ closed under complement $\bar{A}_1$ regular)

A_2 is also regular ($\because$ regular lang. closed under intersection)

As A_1, A are infinite then A_2 is infinite lang.

And we know A_2, ~~is~~ A_1 are disjoint sets.

Thus, $A = A_1 \cup A_2$

Thus A into two infinite disjoint regular subsets