# UNIT 4 IMPORTANT QUESTIONS
## SHORT QUESTIONS

I. What is the use of normalization?

A:- → Context Free grammar can be written in some standard forms is known as Normal Forms.

→ In Context Free grammar left side only non-terminals, right side. combination of terminals & non-terminals. then we need to normalize such grammar.

→ These normal forms certain reductions on the production of CFG. then CFG contain fixed no. of terminals and non-terminals.

2 Define Turing Machine?

A: Turing machine is a mathematical model which consists one infinite length tape is divided into cell on which input is given.

→ Turing machine is used to accept recursively enumerable language.

Formal Definition: There are 7 tuples

$(Q, \Sigma, \Gamma, \delta, q_0, b, F)$

$Q \rightarrow$ set of finite states

$\Sigma \rightarrow$ input symbols

$\Gamma \rightarrow$ Tape alphabets

$q_0 \rightarrow$ initial state

$b \rightarrow$ blank symbol $(b \in \Gamma)$

$F \rightarrow$ final state $(F \subseteq Q)$

$\delta \rightarrow$ Transition Function

$Q \times x \longrightarrow Q \times x \times \{$ left (or) right (or) no more $\}$.

3. Define pumping lemma for the CFG

A: Pumping Lemma is used to prove that the given language is not CFL

Let 'L' is a CFL, there is a pumping length 'n' such that any string $z \in L$ then the length of the string $|z| \geq n$

Here 'n' is the natural number

Here we can break $z = uvwxy$

given conditions

$|vx| \geq 1$

$|vwx| \leq n$

$uv^kwx^ky \in L$ for all $k \geq 0$

4. What is chomsky Normal form?

A: A Context Free grammar is in Chomsky Normal Form if all production rules satisfy one of the following conditions:

→ Start symbol generating $\epsilon$

Ex: $A \rightarrow \epsilon$.

→ A Non-terminal generating two non-terminals

Ex: $S \rightarrow AB$.

→ A non-terminal generating a terminal

Ex: $S \rightarrow a$

5 Discuss the decision properties of CFL's

A. 1) Emptiness Test.
   → Generating test
   → Reachability test

2) Membership Test
   → PDA acceptance

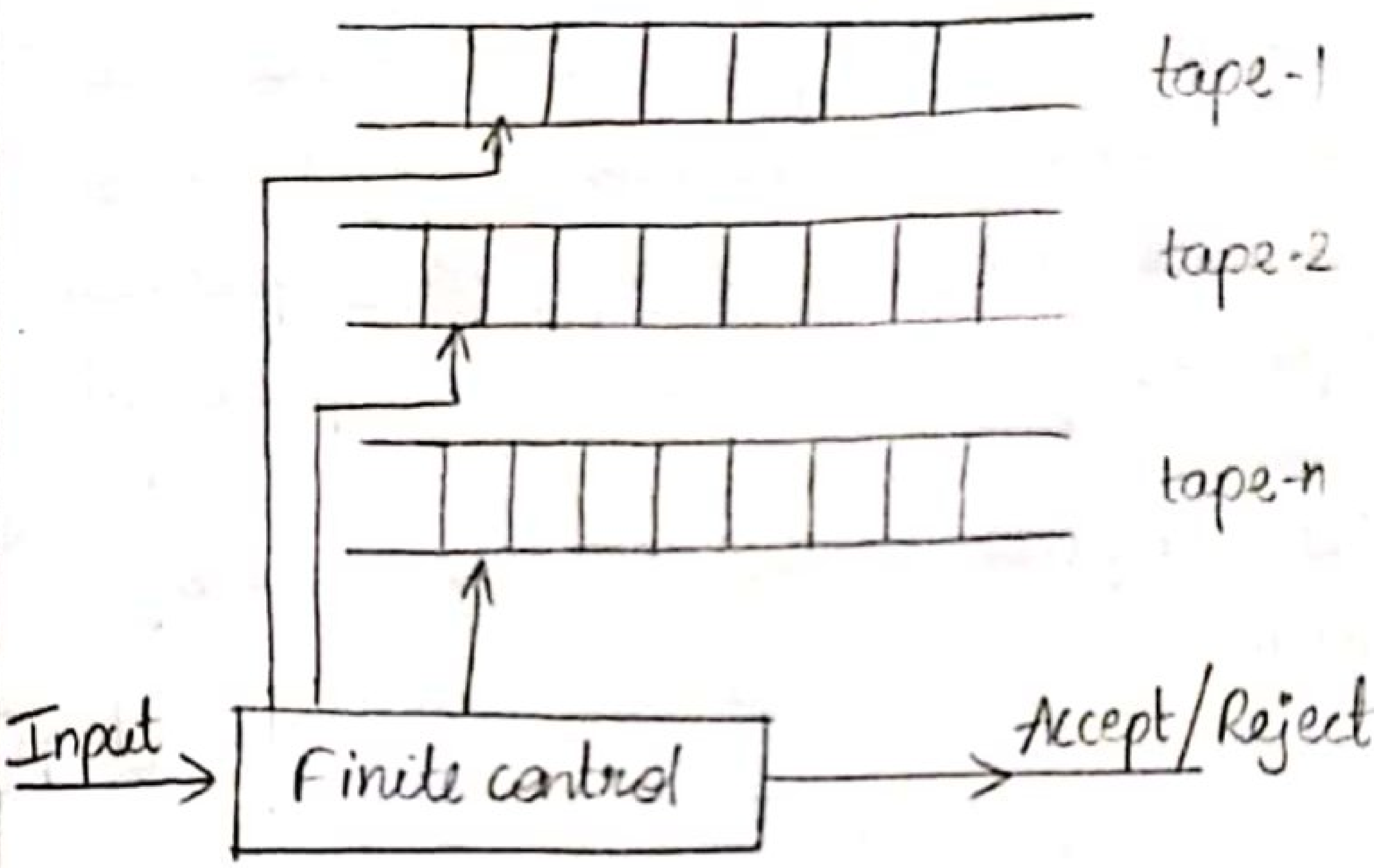
3) Finiteness Test

4) Infiniteness Test

6 Define multi-tape turing machine

A A Turing machine with more than one tape and each tape having its own independent read/write head is said to be multi-tape TM

The language accepted by an n-tape turing machine can be accepted by one-tape TM




$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

$$\delta : Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{L, R\}^k$$

k is the number of tapes

7. What do you mean by instantaneous Description of Turing machine?

A:- The Instantaneous Description of a turing machine is $(q, xaz)$ where

→ $q$ is the current state

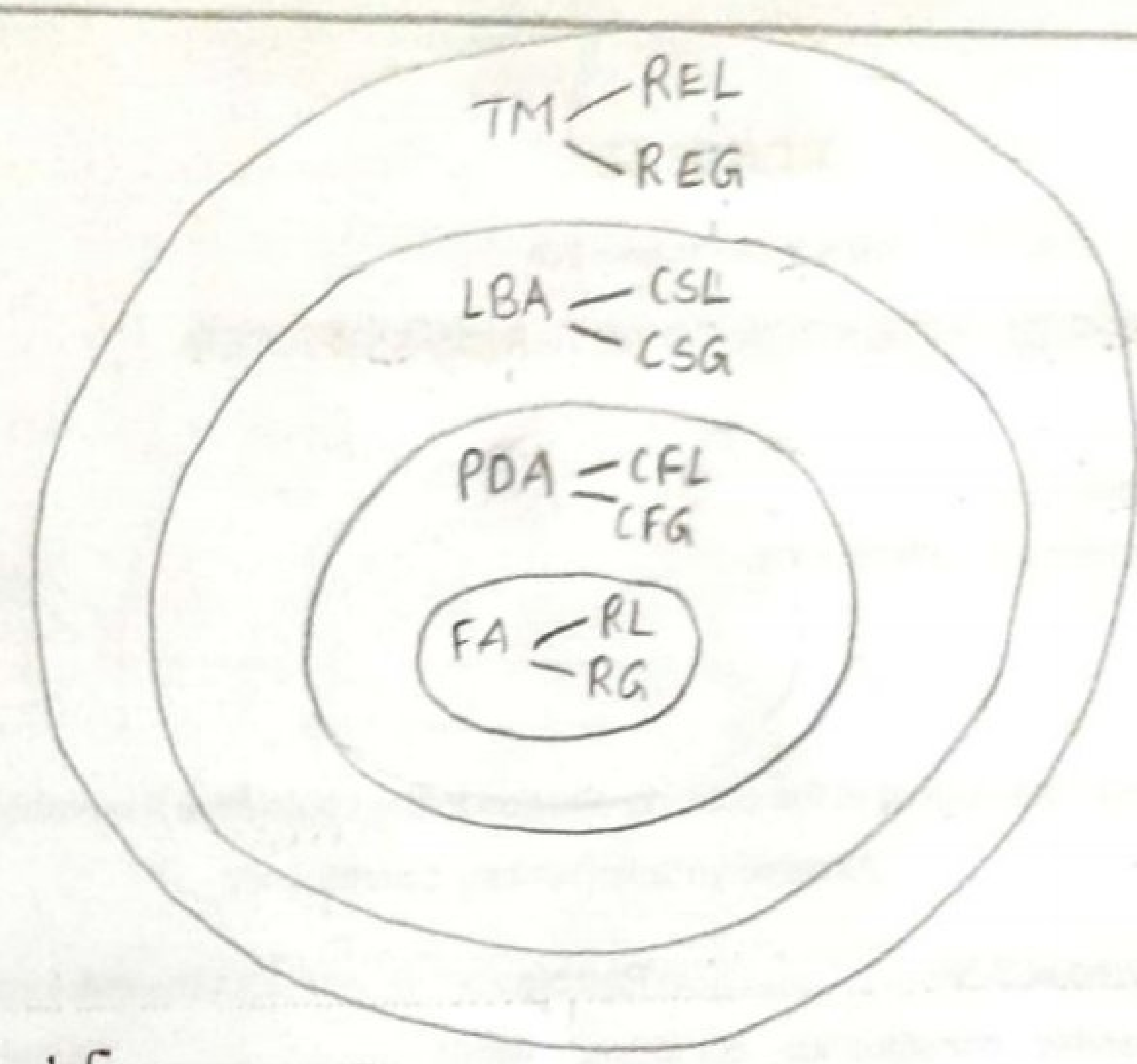
→ $a$ is the character under the reading head, as indicated by underline.

→ $x$ is the string to the left of $a$ on the tape

→ $z$ is the string to the right of $a$ on the tape

8. Briefly explain about chomsky hierarchy of languages

A:




Level-1 [Type-3]:

By using Regular language [RL] we are going to generate Regular Grammar [RG] to accept the regular language we are designing finite automata [FA]

Level-2 [Type-2]:

By using Context free language (CFL) we are going to generate Context free grammar to accept CFL we are going to draw Pushdown automata (PDA)

Level-3 [Type-1]:

By using Context sensitive language [CSL] we are going to generate Context sensitive Grammar [CSG] to accept the CSL we are going to design linear Bounded Automata(LBA).

Level-4 [Type-0]:

By using Recursively enumerable language ~~to~~[REL] we are going to generate Recursively enumerable Grammar [REG] we are going to design a Turing machine (TM)

9 Change the grammar to CNF

$S \rightarrow aSa / aa$

$S \rightarrow bSb / bb$

$S \rightarrow a / b$

A:- $S \longrightarrow aSa$

$S \longrightarrow C_a S C_a \quad [\because C_a = a]$

$S \longrightarrow C_a C_b \quad [\because SC_a = C_b]$

$S \rightarrow aa$

$S \rightarrow C_a C_a$ [$\because C_a = a$]

$S \rightarrow bSb$

$S \rightarrow C_c S C_c$ [$\because C_c = b$]

$S \rightarrow C_c C_d$ [$\because S C_c = C_d$]

$S \rightarrow bb$

$S \rightarrow C_c C_c$

$S \rightarrow a$

$S \rightarrow b$

$P' \Longrightarrow S \rightarrow C_a C_b / C_a C_a$

$S \rightarrow C_c C_d / C_c C_c$

$S \rightarrow a/b$

10 List types of TM

A: Types of Turing Machine :

1) Multiple track
2) Shift over Turing machine
3) Nondeterministic
4) Two way Turing machine
5) Multitape turing machine
6) Multidimensional Turing machine
7) Composite Turing machine.
8) Universal Turing machine.

# LONG QUESTIONS

**11** Define closure properties of context free grammar with proof?

A:- 1. Closed under Union:

→ If $L_1$ & $L_2$ are context Free languages then $L = L_1 \cup L_2$ is also context free language i.e. CFL's are closed under union.

Proof: We will consider two languages $L_1$ & $L_2$ which are CFL.

→ We can give these languages using CFG's $G_1$ & $G_2$ such that $G_1 \in L_1$, $G_2 \in L_2$.

The $G_1$ can be given as

$G_1 = \{V, \Sigma, P_1, S_1\}$ where $P_1$ can be given as

$$P_1 = \{S_1 \rightarrow A_1 S_1 A_1 / B_1 S_1 B_1 / \epsilon$$
$$A_1 \rightarrow a$$
$$B_1 \rightarrow b\}$$

Here $V_1 = \{S_1, A_1, B_1\}$ and $\Sigma = \{a, b\}$ and $S_1$ is start symbol

Similarly, $G_2 = (V_2, \Sigma, P_2, S_2)$

$V_2 = \{S_2, A_2, B_2\}$ and $S_2$ is start symbol

$P_2 = \{S_2 \to aA_2A_2 / bB_2B_2$

$A_2 \to b$

$B_2 \to a\}$

Now, $L = L_1 \cup L_2$ gives $G \in L$

This G can be written as $G = \{V, \Sigma, P, S\}$

$V = \{S_1, A_1, B_1, S_2, A_2, B_2\}$

$P = \{P_1 \cup P_2\}$, $\Sigma = \{a, b\}$

then S is start symbol

$P = \{S \to S_1 / S_2$

$S_1 \to A_1S_1A_1 / B_1S_1B_1 / \epsilon$

$A_1 \to a$

$B_1 \to b$

$S_2 \to aA_2A_2 / bB_2B_2$

$A_2 \to b$

$B_2 \to a\}$

∴ Thus, G is CFG.

2) If $L_1$ & $L_2$ are two CFL then $L_1L_2$ is CFG that means CFL's are closed under concatenation

Proof: Let $L_1$ is CFL which can be represented by a CFG is $G_1$ such that $G_1 \in L_1$ and $G_1 = \{V_1, \Sigma, P_1, S_1\}$

$V_1 = \{S_1, A_1, B_1\}$, $\Sigma = \{a, b\}$

$S_1$ is a start symbol and

$P_1$ is a set of production rules

$P_1 = \{S_1 \to A_1S_1A_1 / B_1S_1B_1 / \epsilon$

$A_1 \to a$

$B_1 \to b\}$

Similarly, $L_2$ is CFL which can be represented by a CFG is $G_2$ such that $G_2 \in L_2$ and $G_2 = \{V_2, \Sigma, P_2, S_2\}$

$V_2 = \{S_2, A_2, B_2\}$ $\Sigma = \{a, b\}$
$S_2$ is a start symbol
$P_2$ is set of production rules

$P_2 = \{ S_2 \rightarrow aA_2A_2 \,/\, bB_2B_2$
$A_2 \rightarrow b$
$B_2 \rightarrow a \}$

Now $L = L_1 \cdot L_2$ can be obtained by $G$ such that $G = G_1 \cdot G_2$
Therefore, $G = \{V, \Sigma, P, S\}$
$V = \{S, S_1, A_1, B_2, S_2, A_2, B_1\}$
where $S$ is a start symbol. Then the production rule $P$ can be given as

$P = \{ S \rightarrow S_1 / S_2$
$S_1 \rightarrow A_1S_1A_1 \,/\, B_1S_1B_1 \,/\, \epsilon$
$A_1 \rightarrow a$
$B_1 \rightarrow b$
$S_2 \rightarrow aA_2A_2 \,/\, bB_2B_2$
$A_2 \rightarrow b$
$B_2 \rightarrow a \}$

As grammar 'G' is CFG the language 'L' produced by 'G' is CFL.
Hence, G CFL's are closed under concatenation

If $L_1$ is CFL
Concatenation: $L_1 \rightarrow G_1$
$L_2 \rightarrow G_2$

$L_1 \cdot L_2 \rightarrow G = \{V, \Sigma, P, S\}$
$G_1 = \{ S_1 \rightarrow aA$
$A \rightarrow a \}$
$G_2 = \{ S_2 \rightarrow bB$
$B \rightarrow b \}$
$S \rightarrow S_1 \cdot S_2$

$S_1 \rightarrow aA$

$A \rightarrow a$

$S_2 \rightarrow bB$

$B \rightarrow b$

3) If $L_1$ is CFL then $L_1^*$ is also CFL that means CFL is closed Kleen closure

$G_1 = \{V_1 \cup (S), T_1 \cup \Sigma, S_1, P_1 \cup S \rightarrow S_1 S / \epsilon\}$

Proof:- Let L be CFL represented by $G_1$ such that $G_1 \rightarrow \epsilon L_1$

The CFG $G_1$ can be given as

$G_1 = \{V_1, \Sigma, P_1, S_1\}$ where $S_1$ is start symbol

$V_1 = \{S_1, A_1, B_1\}$ $\Sigma = \{a, b\}$

$P_1 = \{S_1 \rightarrow A_1 S_1 A_1 / B_1 S_1 B_1 / \epsilon$

$A_1 \rightarrow a$

$B_1 \rightarrow b\}$

Now $L = L_1^*$ can be represented by a Grammar G such that

$G = \{V, \Sigma, P, S\}$

$V = \{S_2, S_1, A_1, B_1\}$ and $P = \{S \rightarrow S_1 S / \epsilon$

$S_1 \rightarrow A_1 S_1 A_1 / B_1 S_1 B_1$

$A_1 \rightarrow a$

$B_1 \rightarrow b\}$

Thus grammar G is a CFG and language produced by G is also CFG.

Hence CFL are closed under Kleen closure.

4) If $L_1$ & $L_2$ are two CFL'S then $L = L_1 \cap L_2$ may be CFL (or) may not be CFL that means L is not closed under intersection

Let $L_1 = \{a^n b^n c^m / n, m \geq 0\}$

$S_1 \rightarrow S_1 C$

$S_1 \rightarrow a S_1 b / \epsilon$

$C \rightarrow cC / \epsilon$

$L_2 = \{a^n b^m c^m / n, m \geq 0\}$

$S \rightarrow S_2 A$

$S_2 \rightarrow S_2 bc / \epsilon$

$A \rightarrow aA / \epsilon$

then $L_1 \cap L_2 = \{a^n b^n c^n / n > 0\}$ is not CFL

$\therefore$ CFL are not closed under intersection

5) If $L_1$ is CFL. then $L_1'$ may (or) may not be CFL that means CFL is not closed under complement.

i.e. $L_1 \cap L_2 = \overline{L_1}$

$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

$L_1$ & $L_2$ are CFL. $\overline{L_1}$ & $\overline{L_2}$ may not be CFL.

12 Design a Turning machine for unary multiplication

A: Let us consider $2 \times 2 = 4$

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | # | 0 | 0 | # | ${b\!\!\!/}$ | ${b\!\!\!/}$ | ${b\!\!\!/}$ | ${b\!\!\!/}$ | ${b\!\!\!/}$ | ${b\!\!\!/}$ |

$0 \rightarrow {b\!\!\!/}, R$

${b\!\!\!/}\ \underset{\uparrow}{0}\ \#\ 0\ 0\ \#\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}$

$0 \rightarrow 0, R$ , $\# \rightarrow \#, R$

${b\!\!\!/}\ 0\ \#\ \underset{\uparrow}{0}\ 0\ \#\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}$

$0 \rightarrow x, R$

${b\!\!\!/}\ 0\ \#\ x\ \underset{\uparrow}{0}\ \#\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}$

$0 \rightarrow 0, R$

$\# \rightarrow \#, R$

${b\!\!\!/}\ 0\ \#\ x\ 0\ \#\ \underset{\uparrow}{{b\!\!\!/}}\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}$

${b\!\!\!/} \rightarrow 0, L$

${b\!\!\!/}\ 0\ \#\ x\ 0\ \underset{\uparrow}{\#}\ 0\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}$

$\# \rightarrow \#, L$

$0 \rightarrow 0, L$

${b\!\!\!/}\ 0\ \#\ \underset{\uparrow}{x}\ 0\ \#\ 0\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}$

$x \rightarrow x, R$

$0 \rightarrow x, R$

${b\!\!\!/}\ 0\ \#\ x\ x\ \underset{\uparrow}{\#}\ 0\ {b\!\!\!/}\ {b\!\!\!/}\ {b\!\!\!/}$

$\# \rightarrow \#, R$

$0 \rightarrow 0, R$

${b\!\!\!/}\ 0\ \#\ x\ x\ \#\ 0\ \underset{\uparrow}{{b\!\!\!/}}\ {b\!\!\!/}\ {b\!\!\!/}$

${b\!\!\!/} \rightarrow 0, L$

${b\!\!\!/}\ 0\ \#\ x\ x\ \#\ \underset{\uparrow}{0}\ 0\ {b\!\!\!/}\ {b\!\!\!/}$

$0 \rightarrow 0, L$

#→#,L

ƀ0#xx#00ƀƀ

x→0,L

#→#,L

ƀ0#00#00ƀƀ

0→ƀ,R

#→#,R

ƀƀ#00#00ƀƀ

0→x,R

0→0,R

#→#,R

ƀƀ#x0#00ƀƀ

ƀ→0,L

ƀƀ#x0#000ƀ

0→0,L

#→#,L

ƀƀ#x0#000ƀ

x→x,R

0→x,R

ƀƀ#xx#000ƀ

#→#,R

0→0,R

ƀƀ#xx#000ƀ

ƀ→0,L

ƀƀ#xx#0000

0→0,L

#→0,L

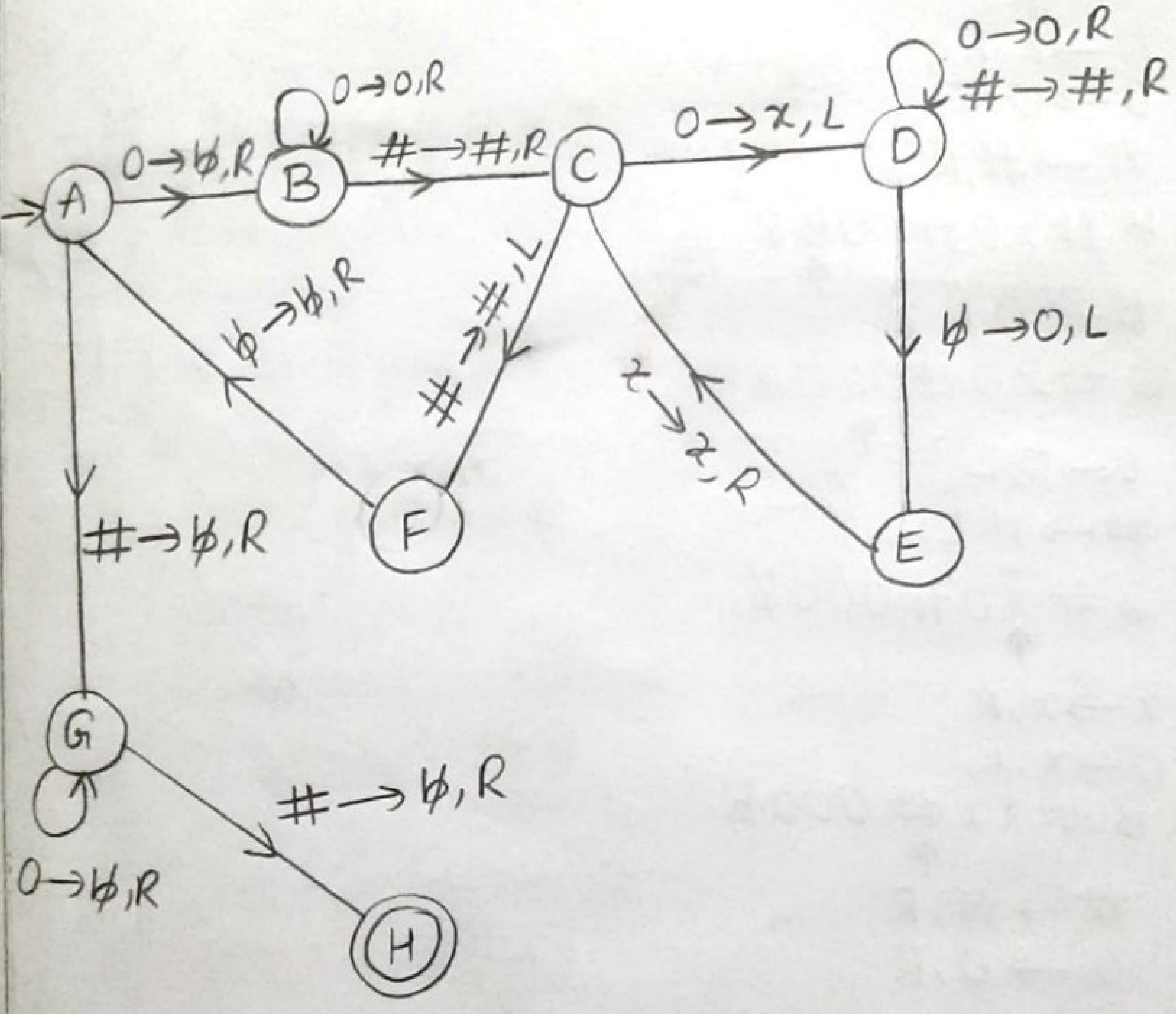
x→0,L

b̸ b̸ #00#0000
↑
b̸ → b̸, R
# → b̸, R

b̸ b̸ b̸ 00#0000
↑
0 → b̸, R
# → b̸, R

b̸ b̸ b̸ b̸ b̸ b̸ 0000

= 4



13 Define Chomsky's normal form?
Change following grammar G into CNF where
$G = (\{S, A, B, D\}, \{a, b, d\}, P, S)$

P: $S \rightarrow aAD$
$A \rightarrow aB \mid bAB$
$B \rightarrow b$
$D \rightarrow d$

A:- A Context free grammar is in Chomsky Normal Form if all production rules satisfy one of the following conditions:

→ Start Symbol generating ∈
Ex:- $A \rightarrow \epsilon$

→ A Non-terminal generating two non-terminals
Ex:- $S \rightarrow AB$

→ A Non-terminal generating a terminal
Ex: $S \rightarrow a$

Conversion from CFG to CNF:

Step-I: Eliminate useless symbols, ∈-productions and unit productions

Step-II: Eliminate terminal symbols in Right hand side.

Step-III: Restrict Number of Non-terminals on the right hand side.

Given $G = (\{S, A, B, D\}, \{a, b, d\}, P, S)$

$S \rightarrow aAD$

$S \rightarrow C_a AD \quad [\because C_a = a]$

$\boxed{S \rightarrow C_a C_c} \quad [\because AD = C_c]$

$A \rightarrow aB$

$\boxed{A \rightarrow C_a B} \quad [\because C_a = a]$

$A \rightarrow bAB$

$A \rightarrow C_b AB \quad [\because C_b = b]$

$\boxed{A \rightarrow C_b C_d} \quad [\because C_d = AB]$

$B \rightarrow b$

$D \rightarrow d$

$P'$: $B \rightarrow b$

$D \rightarrow d$

$S \rightarrow C_a C_c$

$A \rightarrow C_a B / C_b C_d$

14. Construct a Transition diagram for Turing Machine to accept the language

$L = \{W\#W^R / W \in (a+b)^*\}$

A:- $L = \{W\#W^R / W \in (a+b)^*\}$

$L = \{\#, ab\#ba, aba\#aba \ldots\}$

Let us consider a string $ab\#ba$

$ab\#ba$ (head at $a$)

$a \to \not b, R$

$\not b\, b\#ba$ (head at $b$)

$b \to b, R$

$\# \to \#, R$

$a \to a, R$

$\not b\, b\#ba\not b$ (head at last $\not b$)

$\not b \to \not b, L$

$\not b\, b\#ba\not b$ (head at $a$)

$a \to \not b, L$

$b \to b, L$

$\# \to \#, L$

$\not b\, b\#b\not b\not b$ (head at first $\not b$)

$\not b \to \not b, R$

$\not b\, b\#b\not b\not b$ (head at $b$)

$b \to \not b, R$

$\# \to \#, R$

$b \to b, R$

$\not b\not b\#b\not b\not b$ (head at fifth symbol $\not b$)

$\not b \to \not b, L$

$\not b\not b\#b\not b\not b$ (head at $b$)

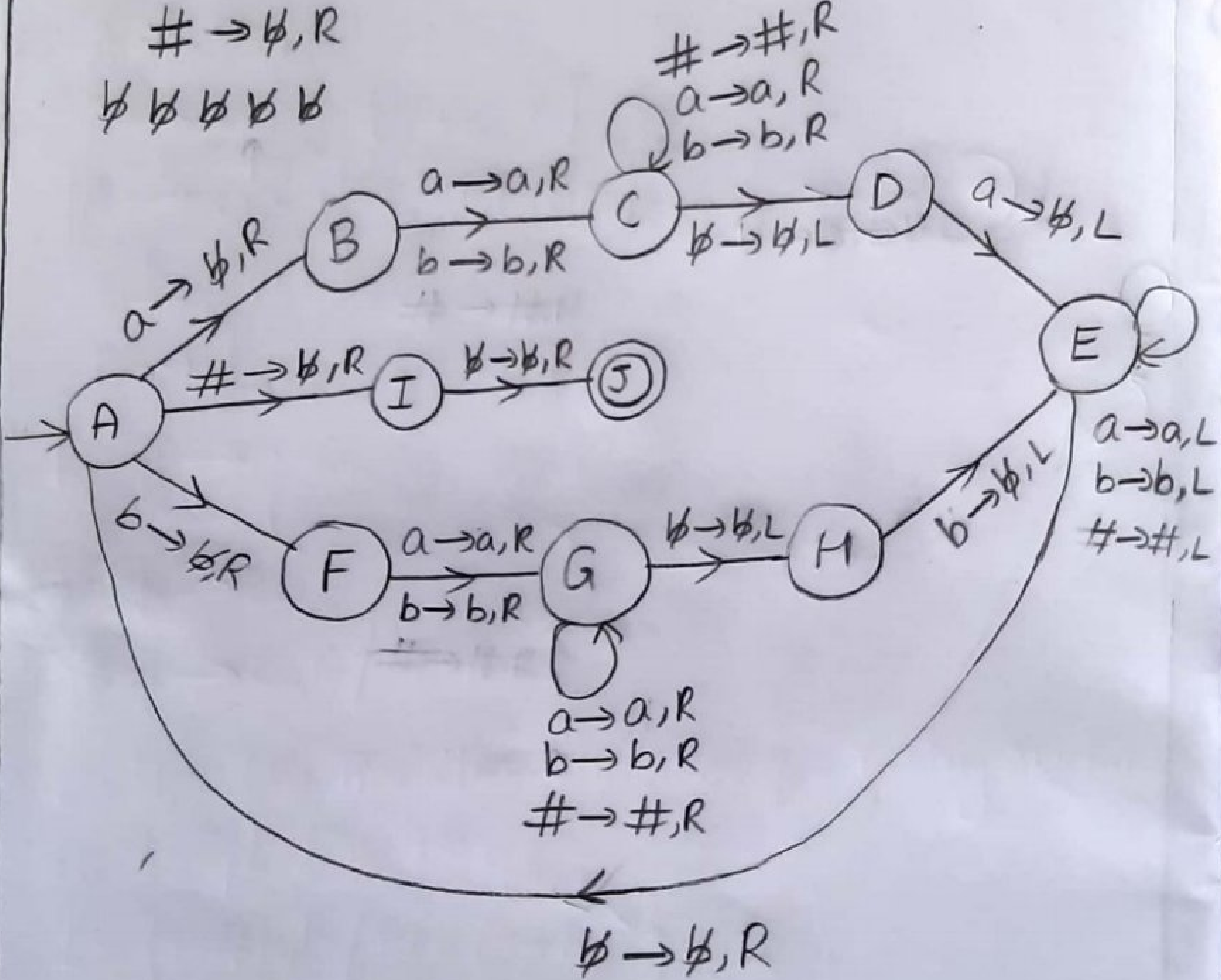
$b \to \not b, L$

b̸ b̸ # b̸ b̸ b̸
↑
# → #, L

b̸ b̸ # b̸ b̸
↑
b̸ → b̸, R

b̸ b̸ # b̸ b̸
↑
# → b̸, R

b̸ b̸ b̸ b̸ b̸



15 Define Greibach Normal form. change the following CFG into GNF $S \rightarrow AA/a$, $A \rightarrow SS/b$

A: Greibach Normal Form [GNF] :

A CFG is in GNF if all the production rules satisfy the these conditions

(1) Start symbol generating '$\epsilon$'

$S \rightarrow \epsilon$

(2) A Non-terminal generating a terminal

$A \rightarrow a$

(3) A Non-terminal generating a terminal which is followed by any no. of Non-terminals

$A \rightarrow aBCD$

Given grammar:

$$S \rightarrow AA / a$$
$$A \rightarrow SS / b$$

Step-1: In the given grammar no useless symbols, no null productions and no unit productions

Step-2: The given grammar is already in the form of CNF.

Step-3: Replace the non-terminals with the new non-terminals i.e.

$S \rightarrow A_1$

$A \rightarrow A_2$ replace these new non-terminals in the given productions i.e.

$$A_1 \rightarrow A_2 A_2 / a$$
$$A_2 \rightarrow A_1 A_1 / b$$

Step-4: The given productions is in the form of $A_i \rightarrow A_j X$

→ For the first production

$$A_1 \rightarrow A_2 A_2$$

Here $i=1, j=2$

$i<j \Rightarrow 1<2$

it satisfies the given condition then their is no problem with the production

→ Second production:

$$A_2 \rightarrow A_1 A_1$$

Here $i=2, j=1$

$i>j \Rightarrow 2>1$

then replace the $A_1$ with the $A_2$ production

$$A_1 \rightarrow A_2A_2 / a$$

After replacing $A_1$ in the $A_2$ with this production

$$A_2 \rightarrow A_2A_2A_1 / aA_1 / b$$

but in the above production $i = j$

then there is a possibility of left recursion

Step-5:

Eliminate the left recursion from the $A_2$ production

$$\underbrace{A_2}_{A} \rightarrow \underbrace{A_2}_{A}\underbrace{A_2A_1}_{\alpha} / \underbrace{aA_1}_{\beta_1} / \underbrace{b}_{\beta_2}$$

A-productions:

$A_2 \rightarrow aA_1 / b$

$A_2 \rightarrow aA_1z / bz$

$[\because$ Rules: $A \rightarrow \beta_1 / \beta_2$

$A \rightarrow \beta_1 z / \beta_2 z]$

z-productions:

$z \rightarrow A_2A_1$

$z \rightarrow A_2A_1z$

$[\because$ Rules

$z \rightarrow \alpha_1$

$z \rightarrow \alpha_1 z]$

Finally, after removing left recursion $A_2$ productions are in the form of GNF but $A_1$ production, z production are not in the form of GNF then replace $A_2$ productions in the $A_1$ and z productions

After replacing

$A_1 \rightarrow A_2A_2 / a$

$A_2 \rightarrow aA_1 / b$

$A_2 \rightarrow aA_1Z / bZ$

$A_1 \rightarrow aA_1A_2 / bA_2 / aA_1ZA_2 / bZA_2 / a$

replacing $A_2$ in z production

$Z \rightarrow aA_1A_1 / bA_1 / aA_1ZA_1 / bZA_1$

$Z \rightarrow aA_1A_1Z / bA_1Z / aA_1ZA_1Z / bZA_1Z$

finally the grammar

$G' = (Q', T, P', A_1)$

$Q' = \{A_1, A_2, Z\}$

$T = \{a, b\}$

$P' \Rightarrow A_1 \rightarrow aA_1A_2 / bA_2 / aA_1ZA_2 / bZA_2 / a$

$A_2 \rightarrow aA_1 / b$

$A_2 \rightarrow aA_1Z / bZ$

$Z \rightarrow aA_1A_1 / bA_1 / aA_1ZA_1 / bZA_1$

$Z \rightarrow aA_1A_1Z / bA_1Z / aA_1ZA_1Z / bZA_1Z$

16 Design a Turing machine for $L = \{a^n b^n / n \geq 1\}$

A:- → Turing machine is used to accept recursively enumerable language.

→ Turing machine consists a head which reads the input tape

→ This Head moves in both directions and also it doesn't accept epsilon.

Formal Definition of Turing machine:

Here, it contains 7 tuples

$(Q, \Sigma, \Gamma, \delta, q_0, b, F)$

$Q$ = set of all states

$\Sigma$ = input alphabets

$\Gamma$ = tape alphabets

$q_0$ = initial state

$b$ = Blank symbol $(b \in \Gamma)$

$F$ = final state $(F \subseteq Q)$

$\delta$ = Transition function

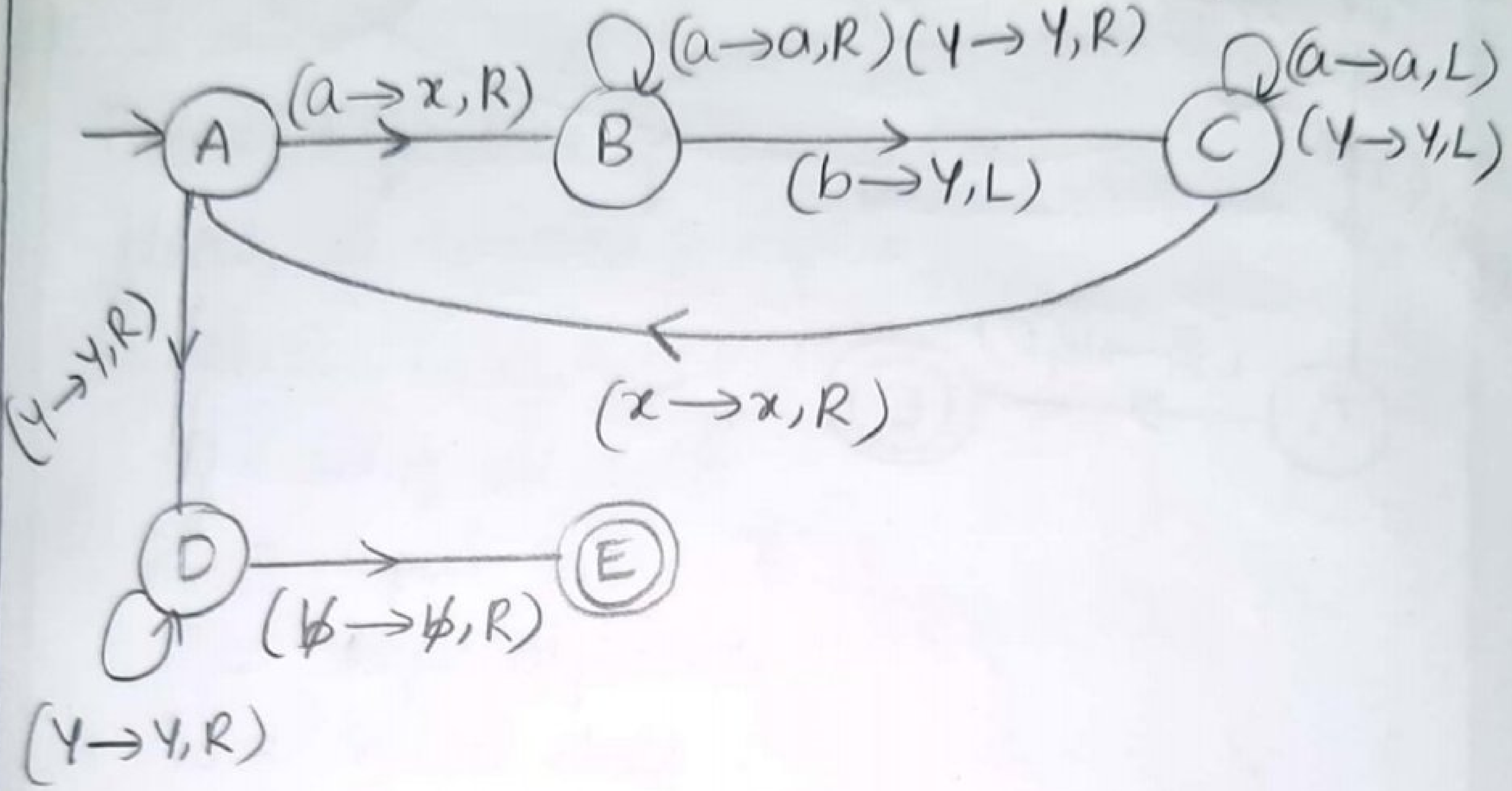
$Q \times x \rightarrow Q \times x \times \{$left (or) right (or) no move$\}$

Here $x$ = tape symbol.

Given,

$L = \{a^n b^n \mid n \geq 1\}$

L = {ab, aabb, aaabbb, ...... }




aabb ⊢ xabb ⊢ xabb ⊢ xaYb ⊢ xaYb ⊢ xaYb
↑ A ↑ B ↑ B ↑ C ↑ C ↑ A

⊢ xxYb ⊢ xxYb ⊢ xxYY ⊢ xxYY ⊢ xxYY ⊢
↑ B ↑ B ↑ B ↑ C ↑ A

xxYY ⊢ xxYYb̸ ⊢ xxYYb̸b̸
↑ D ↑ D ↑ E

The string aabb is halted at E . E is the final state . then the string is accepted .

17. Prove pumping lemma for Context Free Language and Prove $L=\{a^i b^i c^i / i \geq 1\}$ is not CFL

$L = \{abc, aabbcc, aaabbbccc, aaaabbbbcccc, \ldots\}$

$|z| \geq n$

$|9| \geq 7$

$z = aaabbbccc$

Case-1: v, x each contain only one type of symbols

$z=aaabbbccc$

$u = a$

$v = aa$

$w = b$

$x = bb$

$y = ccc$

$k=2$

$uv^2wx^2y$

$aaaaabbbbbccc \Rightarrow a^5b^5c^3 \notin L$

Case-2: v, x has more than one kind of symbols

$z = aaabbbccc$

$u = aa$
$v = ab$
$w = b$
$x = bc$
$y = cc$

$k = 2$

$uv^2wx^2y$

aaababbbcbccc

$a^3bab^3cbc^3 \notin L$

18. Construct a Transition diagram for Turing machine to accept the language $L = \{ ww^R \mid w \in (a+b)^* \}$

A: $L = \{ \epsilon, aa, bb, abba, baab, abaaba, aabbbbaa$
- - - - - - - - - - - - - $\}$

Consider a string aabbbbaa

a a bb bbaa
↑

$a \rightarrow b\!\!\!/, R$

b̸ a bbbbaa
↑

$a \rightarrow a, R$
$b \rightarrow b, R$
$a \rightarrow a, R$

b̸ abbbbaa b̸
↑

$b\!\!\!/ \rightarrow b\!\!\!/, L$
$a \rightarrow b\!\!\!/, L$

b̸abbbba b̸ b̸
↑

$a \rightarrow a, L$
$b \rightarrow b, L$
$b\!\!\!/ \rightarrow b\!\!\!/, R$

b̸ abbbbab̸ b̸
↑

$a \rightarrow b\!\!\!/, R$
$b \rightarrow b, R$
$a \rightarrow a, R$

b̸ b̸ bbbba b̸ b̸
↑

$b\!\!\!/ \rightarrow b\!\!\!/, L$
$a \rightarrow b\!\!\!/, L$

b̸ b̸ bbbb b̸ b̸ b̸
↑

$b \rightarrow b, L$

b̸ b̸ b b b b b̸ b̸

↑

b̸ → b̸, R

b → b̸, R

b̸ b̸ b̸ b b b b̸ b̸

↑

b → b, R

b̸ b̸ b̸ b b b b̸ b̸

↑

b̸ → b̸, L

b → b̸, L

b̸ b̸ b̸ b b b̸ b̸ b̸

↑

b → b, L

b̸ b̸ b̸ b b b̸ b̸ b̸

↑

b̸ → b̸, R

b → b̸, R

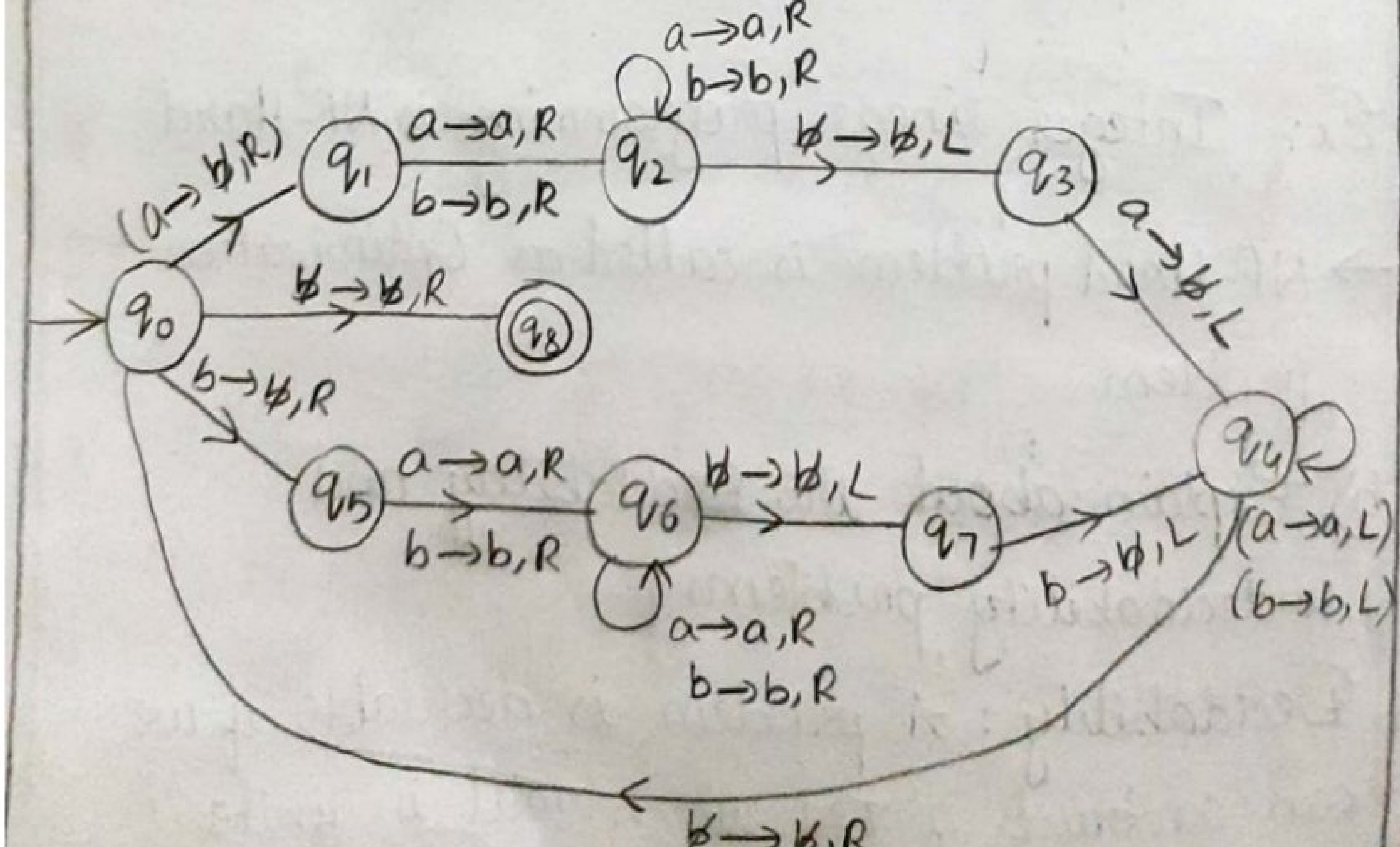
b̸ b̸ b̸ b̸ b b̸ b̸ b̸

↑

b̸ → b̸, L

b → b̸, L

b̸ b̸ b̸ b̸ b̸ b̸ b̸ b̸ b̸



19. Design a CFG for the language $L = \{a^{4n} / n \geq 1\}$ and convert the CFG into CNF form

$n = 1$ $L = \{a^{4n} / n \geq 1\}$

$L = \{a^4\} = \{a, a, a, a\}$

$n = 2$ $L = \{a^8\} = \{a, a, a, a, a, a, a, a\}$

CFG: $S \rightarrow aaaaS / \epsilon$

$S \rightarrow aaaa$

$S \longrightarrow aaaa\,S / \epsilon$

$S \longrightarrow \epsilon$

$S \longrightarrow aaaaS$ $\quad [\because C_a = aa]$

$S \longrightarrow C_aC_aS$ $\quad (\because C_aS = C_b]$

$S \longrightarrow C_a C_b$

$S \longrightarrow aaaa$

$S \longrightarrow C_aC_a$

$P' \Longrightarrow$ $S \rightarrow \epsilon$

$S \longrightarrow C_aC_b$

$S \longrightarrow C_aC_a$

20) Design a TM for Unary Subtraction

A:- $5-3=2$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | # | 0 | 0 | 0 |

$0 \to \not b, R$ , $0 \to 0, R$ $\# \to \#, R$

$\not b\ 0000\ \#\ 000\ \not b$

$\not b \to \not b, L$

$\not b\ 0000\ \#\ 000$

$\uparrow$

$0 \to \not b, L$

$\not b\ 0000\ \#\ 00\ \not b$

$0 \to 0, L$ , $\# \to \#, L$

$\not b \to \not b, R$

$\not b\ 0000\ \#\ 00\ \not b$

$\uparrow$

$0 \to \not b, R$ , $0 \to 0, R$ , $\# \to \#, R$

$\not b\ \not b\ 000\ \#\ 00\ \not b$

$\uparrow$

$\not b \to \not b, L$

$\not b\ \not b\ 000\ \#\ 00\ \not b$

$\uparrow$

$0 \to \not b, L$ $0 \to 0, L$ $\# \to \#, L$

$\not b \to \not b, R$

$0 \to \not b, R$

$\not b\ \not b\ \not b\ 00\ \#\ 0\ \not b\ \not b$

$\not b \to \not b, L$ , $0 \to \not b, R$

$\not b\ \not b\ \not b\ 00\ \#\ \not b\ \not b\ \not b$

$\not b\ \not b\ \not b\ \not b\ 0\ \#\ \not b\ \not b\ \not b$

$\not b \to \not b, L$

$\# \to 0, R$

(or) $\Rightarrow$ $\not b\ \not b\ \not b\ 00\ \#\ \not b\ \not b\ \not b$

$\Rightarrow$ 2 is output

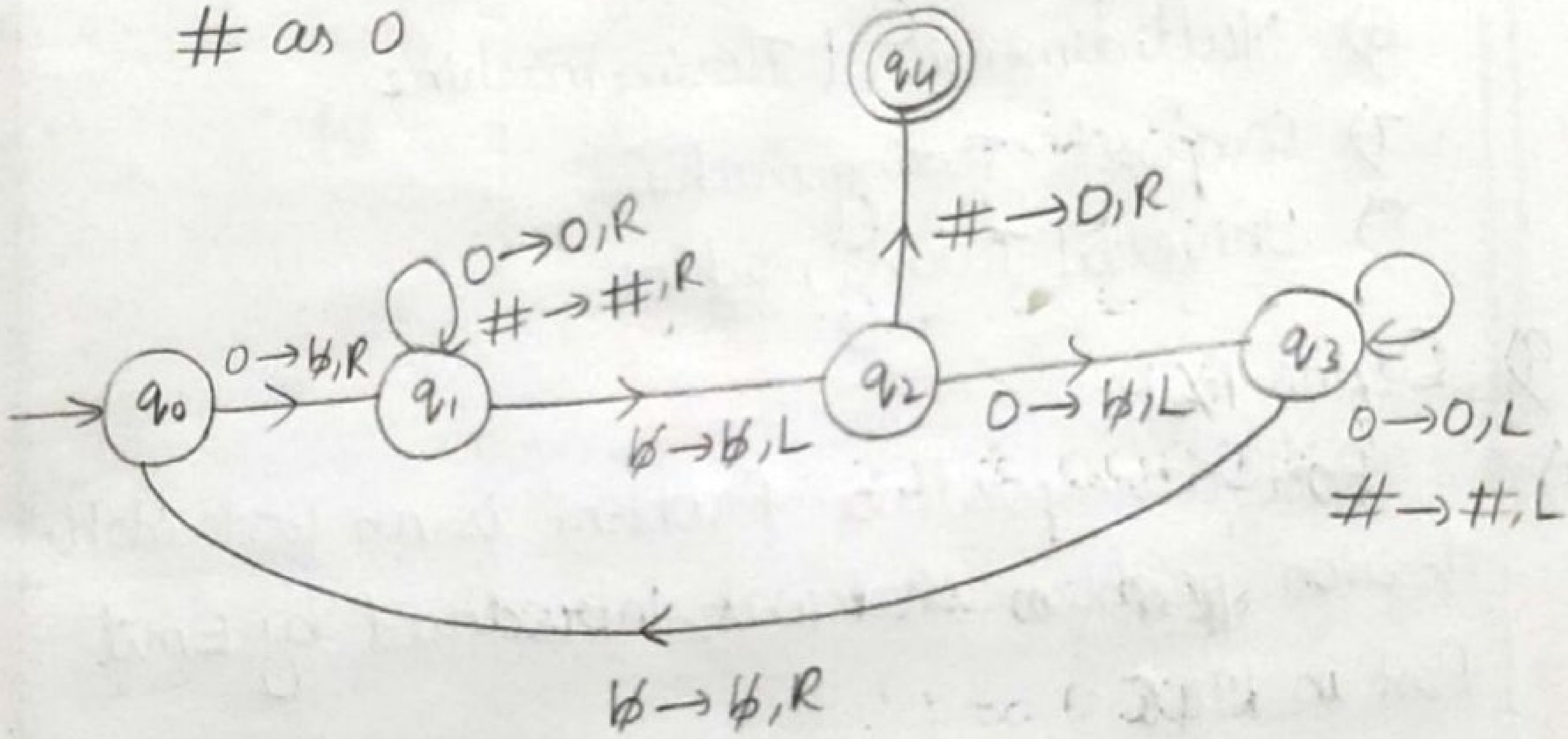
b b b b 0 0 b b b

⇒ 2 is the Output

Here, we are having 2 options

1st: When we making blank for the zero there is no pair when we can't make them as blank

2nd: When we making zero as blank then make # as 0




Here m>n

if m≤n ⇒ 0