Introduction to Automata Theory & Formal Languages

MCA II Semester

1. **Unit-I:**

Basic concepts of Automata Theory: Alphabets, Strings and Languages, Deterministic Finite Automata (DFA) and Nondeterministic Finite Automata (NFA) – Definition, Representation using Transition Tables and State Diagrams, Language of DFA and NFA. NFA with ε-transitions, Language of NFA with ε-transitions, Equivalence of NFA and DFA

38.

2. **Unit – II:**
3. **Regular Expressions and Languages:** Introduction, Definition of regular expression, Kleen's Theorem, Equivalence of regular expression and Finite Automata, Pumping Lemma for regular Languages, Closure properties of Regular Languages, Decision properties of Regular Languages, Finite Automata with Output: Moore and Mealy Machine, Equivalence of Moore and Mealy Machines.
4.
5. **Unit – III:**
6. **Non-Regular Grammars:** Definition of Grammar, Classification of Grammars, Chomosky's Hierarchy. Context Free Grammars (CFG) and Context Free Languages (CFL) - Definition, Examples, Derivation trees, Ambiguous Grammars, Simplification of Grammars, Normal forms of CFGs: CNF and GNF, Closure properties of CFLs, Decision Properties of CFLs, Pumping lemma for CFLs. Push Down Automata (PDA): Definition and Description, Language of PDA and its applications.
7.
8. **Unit – IV:**
9. **Turing Machines:** Introduction, Basic Features of a Turing Machine, Language of a Turing Machine, Variants of Turing Machine: Multitapes, Nondeterministic Turing Machine, Universal Turing Machine. Turing Machine as Computer of Integer functions, Halting problem of Turing Machine, Church-Turing Thesis
10.
11. **Unit – V:**
12. **Undecidability:** Introduction, Undecidable problems about Turing Machines, Rice's Theorem, Post's Correspondence problem (PCP) and Modified PCP. Tractable and Intractable Problems: P and NP, NPComplete Problems, Introduction to recursive function theory
13.
14.
15. **Text Books:**
16. 1. Introduction to Automata theory, Languages and Computation, J.E.Hopcraft, R.Motwani, and Ullman. 2nd edition, Pearson Education Asia
17. 2. Introduction to languages and the theory of computation, J Martin, 3rd Edition, Tata McGraw Hill
18. 3. Elements and Theory of Computation, C Papadimitrou and C. L. Lewis, PHI
19. 4. Mathematical Foundation of Computer Science, Y.N.Singh, New Age International

Subject: Innovation and Entrepreneurship

MCA II Semester

(8 Hrs)

Unit-I: Innovation and Entrepreneurship

Ex-(2.8)
Page-(37) (KLP) Construct a DFA equivalent to $M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$.

Sol^n

Now, we find State table for DFA -

Present state	Next State a	b
$\rightarrow [q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$\boxed{[q_0, q_1, q_2, q_3]}$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$
$\boxed{[q_0, q_1, q_3]}$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[\phi]$	$[\phi]$	$[\phi]$

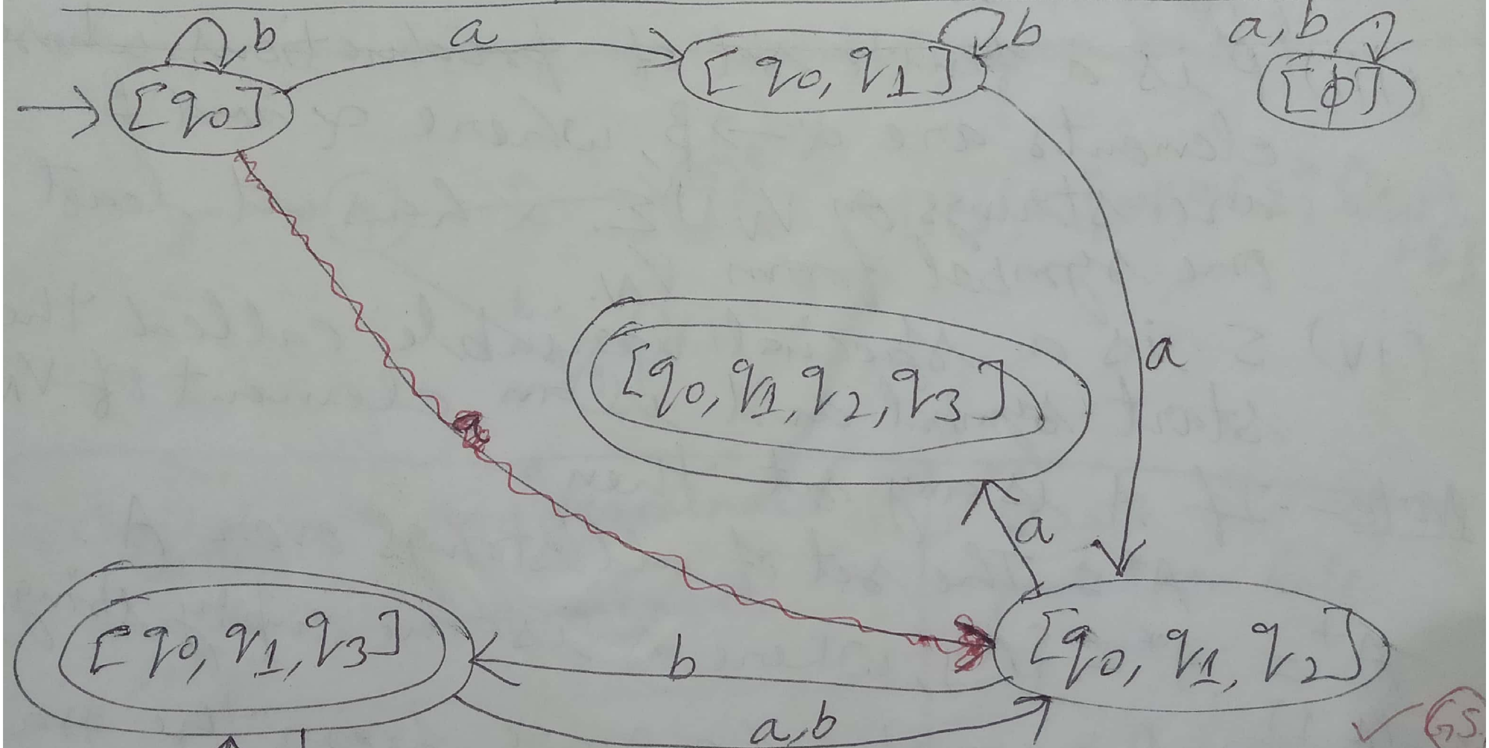


Fig.- Resulting DFA ✓ Ans.

Definition of a Grammar - A Grammar G can be written as a 4-tuple (V_N, Σ, P, S) where -

(i) V_N is a finite non-empty set of variables.

(ii) Σ is a finite non-empty set of elements /terminals.

(iii) P is a finite set of productions whose elements are $\alpha \rightarrow \beta$, where α and β are strings on $V_N \cup \Sigma$. α has at least one symbol from V_N.

(iv) S is a special variable called the start symbol and is an element of V_N.

Note- If A is any set, then

A^* = The set of all strings over A.

$A^+ = A^* - \{\Lambda\}$, where Λ is the empty string.

Ex-(4.1) Pragati(63) Derive the Language from the grammar G

$$S \rightarrow aSb$$
$$S \rightarrow \Lambda$$

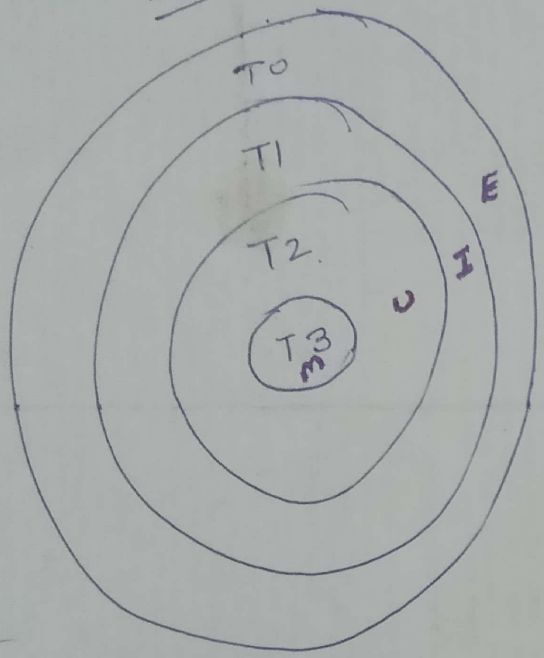
Sol.n

$$S \rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaa\Lambda bbb \Rightarrow aaabbb$$

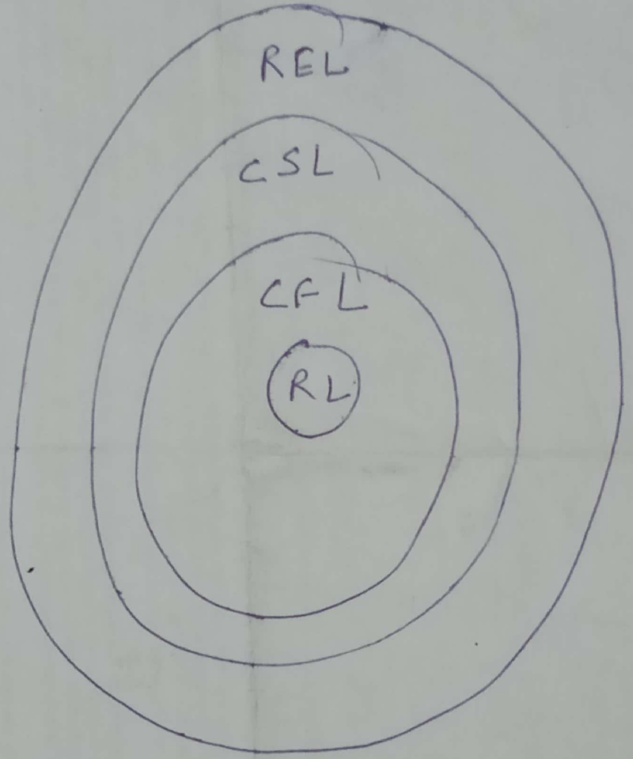
$L(G) = \{a^n b^n \mid n \geq 0\}$

GS.

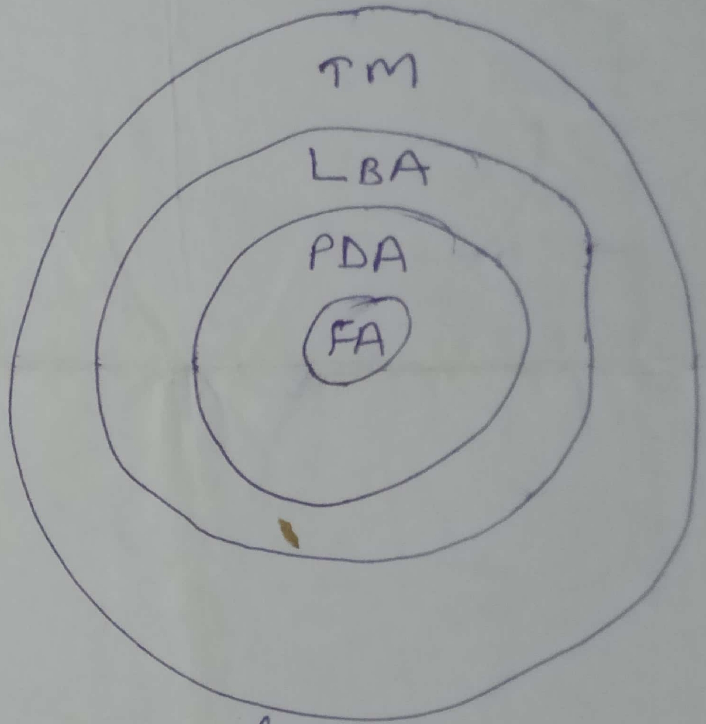
Ans.



Grammar



Language



Machine

__Note__- Grammar is a generator which generates a language and machine is an acceptor which accepts the language.

* **Type-0 Grammar** - (It is also called as Recursive enumerable Grammar/ Unrestricted Grammar)/ Phrase Structured Grammar. (Type-0 Grammar is used to generate recursive enumerable language (R.E.L.) which is accepted by Turing machine (T.M.).

$$G = (V_n, \Sigma, P, S)$$

$$\alpha \rightarrow \beta$$

$$\alpha \in (\Sigma \cup V_n)^* V_n (\Sigma \cup V_n)^*$$

$$\beta \in (\Sigma \cup V_n)^* \;)$$

R.E.G.
↓
R.E.L.
↓
T.M.

* **Type-1 Grammar** - (It is also called as context Sensitive Grammar)/ Length Increasing Grammar / Non-contracting Grammar. (Type-1 Grammar is used to generate context sensitive language (C.S.L.) which is accepted by Linear Bounded Automata (LBA).

$$G = (V_n, \Sigma, P, S)$$

$$\alpha \rightarrow \beta$$

$$\alpha \in (\Sigma \cup V_n)^* V_n (\Sigma \cup V_n)^*$$

$$\beta \in (\Sigma \cup V_n)^+$$

$$|\alpha| \leq |\beta| \;)$$

C.S.G.
↓
C.S.L.
↓
L.B.A.

$$(\alpha A \beta \rightarrow \alpha \delta \beta$$

$$\alpha \beta \in (\Sigma \cup V_n)^*$$

$$A \in V_n$$

$$\delta \in (\Sigma \cup V_n)^+ \;)$$

* Type-2 Grammar - It is also called Context Free Grammar. Type-2 Grammar generates Context Free language (C.F.L.) which is accepted by Push-down Automata (P.D.A.).

$G = (V_n, \Sigma, P, S)$

$\alpha \rightarrow \beta$

$\alpha \in V_n, \quad |\alpha| = 1$

$\beta \in (\Sigma \cup V_n)^*$

C.F.G.
↓
C.F.L.
↓
P.D.A

Note - The parsers for different Programming Languages are created using C.F.G..

* Type-3 Grammar - It is also called Regular Grammar. Type-3 Grammar generates Regular language which is accepted by Finite Automata (F.A.).

$G = (V_n, \Sigma, P, S)$

R.G.
↓
R.L.
↓
F.A.

(a.) Left-Linear Grammar -

$A \rightarrow a \mid Ba$

$A, B \in V_n, \quad |A| = |B| = 1$

$a \in \Sigma^*$

(b.) Right-Linear Grammar -

$A \rightarrow a \mid aB$

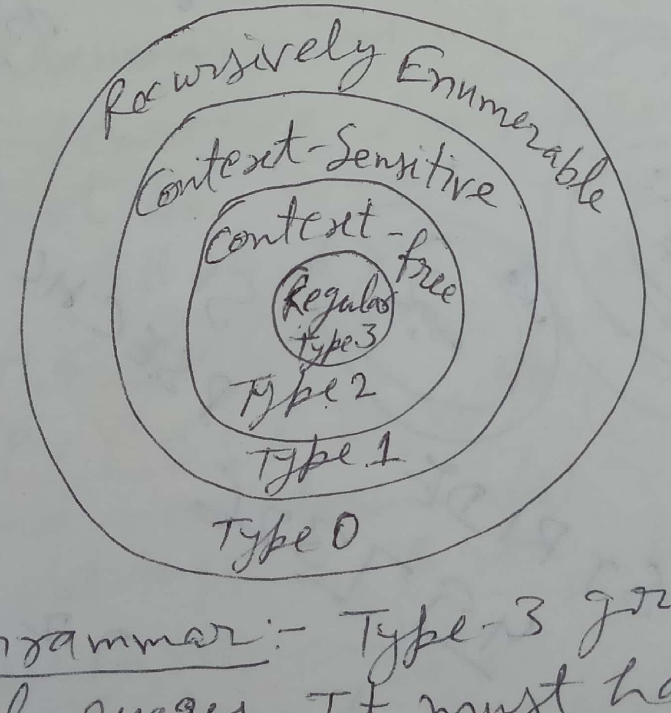
$A, B \in V_n, \quad |A| = |B| = 1$

$a \in \Sigma^*$

Chomsky classification of Grammars-

According to Noam chomsky, there are four types of grammars - Type 0, Type 1, Type 2, and Type 3.

Grammar Type	Grammar Accepted	Language Accepted	Automation
Type 0	Unrestricted Grammar	Recursively enumerable language	Turing Machine
Type 1	Context-sensitive Grammar	Context-sensitive language	Linear-bounded Automation
Type 2	Context-free Grammar	Context-free language	Pushdown Automation
Type 3	Regular Grammar	Regular Language	Finite State Automation



(a.) Type-3 Grammar:- Type-3 grammars generat regular languages. It must have a single non-terminal on the left-hand side an a right-hand side consisting of a single terminal or single terminal followed by a single non-terminal.

The productions must be in the form ㊻
$X \rightarrow a$ or $X \rightarrow aY$ where,
$X, Y \in V_N$ (set of variables)
$a \in \Sigma$ (set of input alphabet)
$S \rightarrow \epsilon$ is allowed if S does not appear on the right hand side of any rule.

(b.) Type-2 Grammar - Type-2 grammars generate context-free languages. The productions must be in the form $A \rightarrow \gamma$ where,
$A \in V_N$ (set of variables)
$\gamma \in (\Sigma \cup V_N)^*$ (String of terminals and non-terminals)

e.g. - $S \rightarrow Xa$, $X \rightarrow a$, $X \rightarrow aX$, $X \rightarrow abc$,
$X \rightarrow \epsilon$

(c.) Type-1 Grammar - Type-1 grammars generate context-sensitive languages. The productions must be in the form -
$\alpha A \beta \rightarrow \alpha \gamma \beta$ where,
$A \in V_N$ (set of variables)
$\alpha, \beta, \gamma \in (\Sigma \cup V_N)^*$ (String of terminals and non-terminals)

Note - The strings α, β may be empty, but γ must be non-empty.
The rule $S \rightarrow \epsilon$ is allowed if S does not appear on the right side of any rule

e.g. - $AB \rightarrow AbBc$, $A \rightarrow bcA$, $B \rightarrow b$

(4.) Type-0 Grammar - Type-0 grammars generate recursively enumerable languages. The productions can be in the form of $a \to \beta$ where a is a string of terminals and non-terminals with at least one non-terminal and a can not be null. β is a string of terminals and non-terminals.

e.g.- $S \to ACaB$, $Bc \to acB$, $CB - DB$, $aD \to Db$

Ex.(3.16) KLP(64) (a) $aAbcD \to abcDbcD$

It is a type-1 production. a is left context and bcD is right context. A is replaced by bcD.

(b) $AB \to AbBc$

It is a type-1 production. The left context is A, right context is Λ.

(c) $A \to abA$

It is a type-1 production. Here, both left and right contexts are Λ.

Ex (3.17) KLP(66) (a) $S \to Aa$, $A \to c|Ba$, $B \to abc$

Type-2 Type-3 Type-2 Type-2

So, highest type number is =2. ✓ GS Ans.

(b) $S \to ASB$, $S \to d$, $A \to aA$

Type-2 Type-3 Type-3

So, highest type number is =2. Ans ✓ GS

(c) $S \to aS$, $S \to ab$

Type-3 Type-2

So, highest type number is = 2 Ans ✓ GS

Chomsky Normal Form - (One kind of Normal form we can look for is one in which the no. of symbols on the right of a production are strictly limited. In particular, we can ask that the string on the right of a production consists of no more than two symbols. One Instance of this is the chomsky Normal Form.)

Example (A context free grammar is in Chomsky normal form if all productions are of the form.

$$A \rightarrow BC$$

or

$$A \rightarrow a$$

where A, B, C are in V, and a is in T.

Example The grammar

$$S \rightarrow AS \mid a$$
$$A \rightarrow SA \mid b$$

is in chomsky normal form.

The Grammar

$$S \rightarrow AS \mid AAS$$
$$A \rightarrow SA \mid aa$$

is not; both productions $S \rightarrow AAS$ and $A \rightarrow aa$.

Introduce new variables B_1 & B_2 to make G in CNF

$$\left[\begin{array}{ll} S \rightarrow AB_1 & A \rightarrow aB_2 \\ B_1 \rightarrow AS & B_2 \rightarrow a \end{array}\right]$$ Ans.)

Example Convert the grammar with productions

$$S \to ABa$$
$$A \to aab$$
$$B \to Ac$$

to Chomsky normal form.

Step1 we introduce new variables B_a, B_b, B_c and use the algo, to get.

$$S \to ABB_a$$
$$A \to B_a B_a B_b$$
$$B \to A B_c$$
$$B_a \to a$$
$$B_b \to b$$
$$B_c \to c$$

Step 2 we introduce additional variables to get the first two productions into normal form and we get the final result.

$$S \to AD_1$$
$$D_1 \to BB_a$$
$$A \to B_a D_2$$
$$D_2 \to B_a B_b$$
$$B \to AB_c$$
$$B_a \to a$$
$$B_b \to b$$
$$B_c \to c$$

Ex-(3.18)
KLP(68) Grammar G given by $S \to 0SA_12$, $S \to 012$, $2A_1 \to A_12$, $1A_1 \to 11$.
Test whether (a) $00112 \in L(G)$ and (b) $001122 \in L(G)$.

Sol.ⁿ

(a)

$$S \to 0\underline{S}A_12$$
$$\Rightarrow 00\underline{S}A_12A_12$$
$$\Rightarrow 00012\underline{A_1}\underline{2A_1}2$$
$$\Rightarrow 000\underline{1A_1}2\underline{A_1}22$$
$$\Rightarrow 00011\underline{1A_1}222$$
$$\Rightarrow 000111222$$

So, $00112 \notin L(G)$ Ans. ✓ G.S.

(b)

$$S \to 0\underline{S}A_12$$
$$\Rightarrow 001\underline{2A_1}2$$
$$\Rightarrow 00\underline{1A_1}22$$
$$\Rightarrow 001122$$

So, $001122 \in L(G)$ Ans. ✓ G.S.

Ex-(4.10)
Pragati(70) Find the Highest Type Number of the Grammars.

(i) $A \to Ba$, $B \to c \mid Ca$, $C \to abc$
Type-2 Type-3 Type-2 Type-2

So, Highest Type No. is = 2 Ans. ✓ G.S.

(ii) $B \to aB$, $A \to BAC \mid dab$
Type-3 Type-2 Type-2

So, Highest Type No. is = 2 Ans. ✓ G.S.

(iii) $A \to aA \mid ab$
Type-3 Type-2

✓ G.S.

So, Highest Type No. is = 2 Ans.

Ex-(4.2)
Pragati(63)

$G = (\{S\}, \{a, b\}, P, S)$ where
P is defined as -

$S \to a, \quad S \to b$

$S \to aS, \quad S \to bS$

Solⁿ-

$S \to aS$
$\Rightarrow aaS$
$\Rightarrow aaaS$
$\Rightarrow aaabS$
$\Rightarrow aaabbS$
$\Rightarrow aaabbbS$
$\Rightarrow aaabbba$

$S \to aS$
$\Rightarrow aaS$
$\Rightarrow aaaS$
$\Rightarrow aaabS$
$\Rightarrow aaabbS$
$\Rightarrow aaabbbS$
$\Rightarrow aaabbbb$

$S \to aS$
$\Rightarrow aa$

$S \to aS$
$\Rightarrow ab$

$S \to bS$
$\Rightarrow bb$

$S \to bS$
$\Rightarrow ba$

So, $L(G) = (a+b)^* - \{\Lambda\}$
$= (a+b)^+$ ✓ Ans. ✓ GS

Ex-(4.3)
Pragati(64)

$S \to aS \,|\, aA \,|\, a$
$A \to aAb \,|\, ab$

Solⁿ-

$S \to aS$
$\Rightarrow aaS$
$\Rightarrow aaaS$
$\Rightarrow aaaaA$
$\Rightarrow aaaaaAb$
$\Rightarrow aaaaaabb$

$S \to aS$
$\Rightarrow aa$

$L(G) = \{a^m b^n / m > n \geq 0\}$

✓ CSS. ✓ Ans

Ex.(4.4)
Pragati(64)

$S \to aABa, \quad A \to baABb$
$B \to Aab, \quad aA \to baa, \quad bBb \to abab$

Test whether the string $s = \underline{ba^2b^2a}ba^3b^2\underline{aba}$

Solⁿ $S \to aABa$
$\Rightarrow baaBa$
$\Rightarrow \underline{baa}A\underline{aba}$
$\Rightarrow \underline{baa}ba\underline{A}B\underline{aba}$
$\Rightarrow \underline{ba^2bba}aBb\underline{aba}$
$\Rightarrow \underline{ba^2b^2a}aAabb\underline{aba}$

So, we can

$\Rightarrow \underline{ba^2b^2abaaabbaba}$

$\Rightarrow ba^2b^2aba^3b^2aba$ ✓ GS.

So, $s \in L(G)$ Ans.

Ex-(4.5) Pragati(65) $S \rightarrow AB$, $A \rightarrow BB$, $B \rightarrow AA$. find the language generated by grammar G.

Sol.ⁿ- Since $\Sigma = \{\phi\}$, because all three productions have no terminals on the R.H.S.

So, $L(G) = \{\phi\}$ Ans. ✓ GS

Ex-(4.6) Pragati(65) $G = (\{S, A\}, \{a, b, c\}, P, S)$, where P Contains: $S \rightarrow aSAc$, $S \rightarrow abc$, $cA \rightarrow Ac$, $bA \rightarrow bb$. find the language generated by grammar G.

Sol.ⁿ- $S \rightarrow abc$, here $abc \in L(G)$.

$$\begin{aligned}
S &\rightarrow aSAc \\
&\Rightarrow a\underline{aSAc}Ac \\
&\Rightarrow \underline{aa}\,\underline{aSAc}AcAc \\
&\Rightarrow aaa\underline{abc}AcAcAc \\
&\Rightarrow \underline{aaaa}b\underline{A}cAcAcc \\
&\Rightarrow aaaa\overline{bb}\underline{cA}\,\underline{cA}cc \\
&\Rightarrow aaaabb\underline{Ac}\,\underline{A}ccc \\
&\Rightarrow \underline{aaaabbbA}cccc \\
&\Rightarrow aaaabbbbcccc \\
&\Rightarrow a^4b^4c^4
\end{aligned}$$

✓ GS

So, $L(G) = \{a^n b^n c^n \mid n \geq 1\}$ Ans.

Ex-(4.7) Pragati(65) Find the grammar, generating the language - $L(G) = \{a^n b a^n \mid n \geq 1\}$

Sol.ⁿ- for this language L(G), there must be a production of the form

starting symbol → (a) starting symbol (a)

$S \Rightarrow aSa$
$\Rightarrow aaSaa$
$\Rightarrow aaaSaaa$
$\Rightarrow aaaaSaaaa$

$S \rightarrow aba$, then
$\Rightarrow \underline{aaaa}b\underline{aaaa}$
$\Rightarrow \{a^n ba^n \mid n \geq 1\}$

So, Grammar is
$V_N = \{S\}$, $\Sigma = \{a, b\}$, ✓ G.S.
$P = \{S \rightarrow aSa, S \rightarrow aba\}$, ✓ Ans.
S = Start Symbol.

Ex-(4.8) Prajati(66) Construct the grammar G for set L of all even and odd length palindromes over $\{0, 1\}$.

Solⁿ:- For the construction of grammar G to generate the set of palindromes, these can be the cases:-

(i) Any string of zero length (Λ) is a palindrome.
(ii) Any single alphabet is palindrome (i.e., 0, 1)
(iii) If s is a palindrome then 0s0 and 1s1 will also be palindrome.

So, we can define productions as-

(i) $S \rightarrow \Lambda$, which will give palindrome of zero length.

(ii) $S \rightarrow 0|1$, which will give palindromes of one length.

(iii) $S \rightarrow 0S0|1S1$, which will give palindromes of two or more length

So, Grammar is $G = (\{S\}, \{0,1\}, P, S)$

G.S. Ans. $P = S \rightarrow \Lambda,\ S \rightarrow 0|1,\ S \rightarrow 0S0|1S1$

Ex-(3.2) KLP(57) $G = (\{S\}, \{0,1\}, \{S \rightarrow 0S1, S \rightarrow \Lambda\}, S)$, find $L(G)$.

Solⁿ- $S \rightarrow 0S1$

$\Rightarrow 00S11$

$\Rightarrow 000S111$

$\Rightarrow 000\Lambda 111$

$L(G) \Rightarrow \{0^n 1^n \mid n \geq 0\}$ Ans. G.S.

Ex-(3.3) KLP(57) $G = (\{S\}, \{a\}, \{S \rightarrow SS\}, S)$, find the language generated by G.

Solⁿ- $L(G) = \{\phi\}$, because the only production $S \rightarrow SS$ in G has no terminal on the R.H.S. Ans. G.S.

Ex-(3.4) KLP(57) $G = (\{S, C\}, \{a, b\}, P, S)$, where P consists of $S \rightarrow aCa$, $C \rightarrow aCa|b$. find $L(G)$.

Solⁿ- $S \rightarrow aCa$

$\Rightarrow aaCaa$

$\Rightarrow aaaCaaa$

$\Rightarrow aaabaaa$

$\Rightarrow a^n b a^n$

$S \rightarrow aCa$

$\Rightarrow aba$

$\Rightarrow a^n b a^n$

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

Ans. G.S.

① Answered but marked for review.

Ex (3.5)
KLP (58) $G = S \rightarrow aS \mid bS \mid a \mid b$, find $L(G)$. (41)

Solⁿ:-

$$S \Rightarrow aS$$
$$\Rightarrow aaS$$
$$\Rightarrow aaaS$$
$$\Rightarrow aaabS$$
$$\Rightarrow aaabbS$$
$$\Rightarrow aaabbbS$$
$$\Rightarrow \underline{aaa}\,\underline{bbb}\,a \quad \text{or} \quad \Rightarrow \underline{aaa}\,\underline{bbbb}$$

$$S \Rightarrow bS$$
$$\Rightarrow bbS$$
$$\Rightarrow bbbS$$
$$\Rightarrow bbbaS$$
$$\Rightarrow bbbaaS$$
$$\Rightarrow bbbaaaS$$
$$\Rightarrow \underline{bbb}\,\underline{aaa}\,a \quad \text{or} \quad \Rightarrow \underline{bbb}\,\underline{aaa}\,b$$

So, in the above grammar G; we have only two terminals a, b.

$$L(G) = \{a, b\}^* - \{\wedge\}$$
$$L(G) = \{a, b\}^+$$

because $S \rightarrow \wedge$ is not a production in G. <u>Ans.</u> ✓ (lzS.)

(3.7). P (59) Construct grammar generating $L = \{wcw^T \mid w \in \{a, b\}^*\}$.

<u>Solⁿ</u>:- Let $G = (\{S\}, \{a, b, c\}, P, S)$, where P is defined as $S \rightarrow aSa \mid bSb \mid c$. Because $c \in L$

nal $w \in \{a,b\}^*$. So, the productions are
tone i.e; $S \to aSa \mid bSb \mid \varepsilon$. Ans. ✓ G.S.

Ex-(3.8)
KLP(59) find Grammar generating $L = \{a^n b^n c^i \mid n \geq 1,\ i \geq 0\}$.

Sol^n–

$S \to A$, $A \to ab$, $A \to aAb$, $S \to Sc$

$S \to A$
$\Rightarrow ab$

$S \to A$
$\Rightarrow aAb$
$\Rightarrow aaAbb$
$\Rightarrow \underline{aaa}\,\underline{bbb}$

$S \to Sc$
$\Rightarrow Scc$
$\Rightarrow Sccc$
$\Rightarrow Accc$
$\Rightarrow aAbccc$
$\Rightarrow \underline{aa}\,\underline{bb}ccc$

So, $\boxed{a^n b^n c^0 \in L(G)}$ and so,
$\boxed{a^n b^n c^i \in L(G)}$ Ans. ✓ G.S.

Ex(3.9)
KLP(59) find Grammar generating $L = \{a^j b^n c^n \mid n \geq 1,\ j \geq 0\}$.

Sol^n– $S \to A$, $A \to bc$, $A \to bAc$, $S \to aS$

$S \to A$
$\Rightarrow bc$

$S \to A$
$\Rightarrow bAc$
$\Rightarrow bbAcc$
$\Rightarrow \underline{bbb}\,\underline{ccc}$

$S \to aS$
$\Rightarrow aA$
$\Rightarrow abc$

$S \to aS$
$\Rightarrow aA$
$\Rightarrow abAc$
$\Rightarrow a\underline{bb}\,\underline{cc}$

$S \to aS$
$\Rightarrow aaS$
$\Rightarrow aaA$
$\Rightarrow aa\underline{b}\,\underline{c}$

$S \to aS$
$\Rightarrow aaS$
$\Rightarrow aaA$
$\Rightarrow aabAc$
$\Rightarrow aa\underline{bb}\,\underline{cc}$

So, $a^0 b^n c^n \in L(G)$ and so,
$a^j b^n c^n \in L(G)$ Ans. ✓ G.S.

Ex-(3.10)
KLP(60) $G = (\{S, A_1\}, \{0,1,2\}, P, S)$, where P consists of $S \to 0SA_12$, $S \to 012$, $2A_1 \to A_12$, $1A_1 \to 11$. Show that

$$L(G) = \{0^n 1^n 2^n \mid n \geq 1\}$$

Sol^n
$S \to 012$ i.e, $012 \in L(G)$

$S \rightarrow 0SA_12$

$\Rightarrow 00SA_12A_12$

$\Rightarrow 0001\underline{2A_1}\underline{2A_1}2 \Rightarrow 0001A_122A_12$

$\Rightarrow 0001\underline{1A_1}\underline{2A_1}22 \Rightarrow 000112A_122$

$\Rightarrow 00011\underline{1A_1}222$

$\Rightarrow \underline{000}\underline{111}\underline{222}$ ✓ (GS)

$\Rightarrow 0^n1^n2^n \in L(G)$ Ans.

Ex-(3.11)
KLP (60) Construct a grammar G generating $\{a^nb^nc^n \mid n \geq 1\}$.

Soln:- $S \rightarrow aS\alpha$, $S \rightarrow a\alpha$

So, $S \rightarrow aS\alpha$

$\Rightarrow aaS\alpha\alpha$

$\Rightarrow aaaS\alpha\alpha\alpha$

$\Rightarrow \underline{aaaa}\underline{\alpha\alpha\alpha\alpha}$

$\alpha = BC$
To bring B's toget-
-her we introduc
$CB \rightarrow BC$

Let $G = (\{S, B, C\}, \{a, b, c\}, P, S)$, where P consist
of $S \rightarrow aSBC$, $S \rightarrow aBC$, $CB \rightarrow BC$, $aB \rightarrow ab$,
$bB \rightarrow bb$, $bC \rightarrow bc$, $cC \rightarrow cc$ Ans. ✓ (G

$S \rightarrow aSBC$

$\Rightarrow \underline{aa}S\underline{BCBC}$

$\Rightarrow \underline{aaa}S\underline{BCBCBC}$

$\Rightarrow \underline{aaaa}BCBCBCBC$

$\Rightarrow \underline{aaaab}BCBCBCC$

$\Rightarrow \underline{aaaabb}BCBCCC$

$\Rightarrow \underline{aaaabbb}BCCCC$

$\Rightarrow \underline{aaaa}\underline{bbbb}CCCC$

$\Rightarrow aaaabbbbcCCC$

$\Rightarrow aaaabbbbccCC$

$\Rightarrow aaaabbbbcccC$

$\Rightarrow \underline{aaaa}\underline{bbbb}cccc$

$\Rightarrow a^nb^nc^n \in L($

$S \rightarrow \underline{a}BC$

$\Rightarrow a\underline{b}C$

$\Rightarrow abc \in L(G$

So, $L(G) = \{a^nb^nc^n \mid n$

✓ Ans

Ex-(3.13) KLP(63) Let $G = (\{S, A_1, A_2\}, \{a, b\}, P, S)$, where P consists of $S \to aA_1A_2a$, $A_1 \to baA_1A_2b$, $A_2 \to A_1ab$, $aA_1 \to baa$, $bA_2b \to abab$.

Test whether $w = baabbabaaabbaba \in L(G)$

Sol.ⁿ-

$S \Rightarrow aA_1A_2a$

$\Rightarrow baaA_2a$

$\Rightarrow baaA_1aba$

$\Rightarrow baabaA_1A_2baba$

$\Rightarrow baabbaaA_2baba$

$\Rightarrow baabbaaA_1abbaba$

$\Rightarrow baabbabaaabbaba \in L(G)$

Ans. ✓ GS.

Ex-(3.14) KLP(63) $S \to aSa \mid bSb \mid aa \mid bb \mid \Lambda$. Show that

✓ (a) $L(G)$ has no strings of odd length.

(b.) any string in $L(G)$ is of length $2n$, $n \geq 0$.

(c.) the no. of strings of length $2n$ is 2^n.

Sol.ⁿ-

$S \to \Lambda$

$\Rightarrow \phi$ (0 length) i.e. even length

$S \to bSb$

$\Rightarrow bbSbb$

$\Rightarrow bbbSbbb$

$\Rightarrow bbbaSabbb$

$\Rightarrow bbbaaSaabbb$

$\Rightarrow bbbaaaSaaabbb$

$\Rightarrow bbbaaaaaabbb$

$\Rightarrow a^n b^n$ (i.e. even length)

$\Rightarrow bbbaaaaaaaabbb$ i.e. even length

$\Rightarrow bbbaaabbaaabbb$ i.e. even length

$S \to aSa$

$\Rightarrow aaSaa$

$\Rightarrow aaaSaaa$

$\Rightarrow aaabSbaaa$

$\Rightarrow aaabbSbbaaa$

$\Rightarrow aaabbbSbbbaaa$

$\Rightarrow aaabbbbbbaaa$

$\Rightarrow a^n b^n$ (i.e. even length)

$\Rightarrow aaabbbaabbbaaa$ i.e. even length

$\Rightarrow aaabbbbbbbbaaa$ i.e. even length

Ans. ✓ GS.

$S \rightarrow 0B$
$\Rightarrow 01$

$S \rightarrow 0B$
$\Rightarrow 01S$
$\Rightarrow 010B$
$\Rightarrow 0100BB$
$\Rightarrow 010011$

$S \rightarrow 0B$
$\Rightarrow 01S$
$\rightarrow 011A$
$\Rightarrow 0110$

Ans. ✓ 7.5

$S \rightarrow 0B$
$\Rightarrow 01S$
$\Rightarrow 011A$
$\Rightarrow 0110S$
$\Rightarrow 01101A$
$\Rightarrow 011011AA$
$\Rightarrow 01101100$

Q-2(b) KLP-(73) $\{0^n 1^m 0^m 1^n \mid m, n \geq 1\}$

Sol^n - $S \rightarrow 0S1$, $S \rightarrow 0A1$, $A \rightarrow 1A0$, $A \rightarrow 10$

$S \rightarrow 0S1$
$\Rightarrow 00S11$
$\Rightarrow 000A111$
$\Rightarrow 0001A0111$
$\Rightarrow 0001100111$
$\Rightarrow 0^n 1^m 0^m 1^n$

$S \rightarrow 0A1$
$\Rightarrow 01A01$
$\Rightarrow 011001$
$\Rightarrow 0^n 1^m 0^m 1^n$

Ans. ✓ 7.5

Q-2(c) KLP-(73) $\{0^n 1^{2n} \mid n \geq 1\}$

Sol^n - $S \rightarrow 0S11$, $S \rightarrow 011$

$S \rightarrow 0S11$
$\Rightarrow 00S1111$
$\Rightarrow 000111111$

Ans ✓ 7.5

Q-2(d) KLP-(73) $\{0^n 1^n \mid n \geq 1\} \cup \{1^m 0^m \mid m \geq 1\}$

Sol^n - $S \rightarrow 0A1$, $S \rightarrow 1B0$, $A \rightarrow 0A1$, $A \rightarrow \Lambda$, $B \rightarrow 1B0$, $B \rightarrow \Lambda$

$S \rightarrow 0A1$
$\Rightarrow 00A11$
$\Rightarrow 0011$
$\Rightarrow 0^n 1^n$

$S \rightarrow 1B0$
$\Rightarrow 11B00$
$\Rightarrow 1100$
$\Rightarrow 1^m 0^m$

Ans ✓ 7.5

Q(3)
KLP(73) Test whether the given strings are in the language

Language is S → 0S1, S → 0A, S → 0, S → 1B, S → 1, A → 0A, A → 0, B → 1B, B → 1

(a) 001100

S → 0S1
⇒ 00A1
⇒ 0001

So, 001100 is not accepted by the Language. Ans. ✓ GS

(b.) 001010

S → 0S1
⇒ 00S11
⇒ 001B11
⇒ 001111

So, 001010 is not accepted by the Language. Ans. ✓ GS

(c) 01010

S → 0S1
⇒ 01B1
⇒ 0111

So, 01010 is not accepted by the Language. Ans ✓ GS.

Q(4)
KLP-(73) G = ({A, B, S}, {0, 1}, P, S), where P consists of S → 0AB, A0 → S0B, A1 → SB1, B → SA, B → 01. Show that L(G) = φ.

Soln - S → 0AB
⇒ 0ASA
⇒ No Derivation possible further

S → 0AB
⇒ 0A01
⇒ 0S0B1
⇒ 00AB0B1
⇒ 00A010011
⇒ 00S0B10011
⇒ 000AB00110011
⇒ Repetitive Derivation

So, L(G) = φ is True. ✓ GS.

Q(5.)
KLP(73) Find the language generated by Grammar S → AB, A → A1 | 0, B → 2B | 3

Soln - S → AB
⇒ A1B
⇒ A11B
⇒ A112B
⇒ A1122B
⇒ 011223

$\Rightarrow \{01^m2^n3 \mid m,n \geq 0\}$

✓ GS Ans.

Q-(9.) KLP-(73) Construct a Grammar which generates all even integers up to 998.

Sol^n - $G = (\{S, S_1, A, B\}, \Sigma, P, S)$, where $\Sigma = \{0, 1, 2, ---, 9\}$ and P consists of

$S \rightarrow 0|2|4|6|8$, $S \rightarrow AS_1$, $A \rightarrow 1|2|---|9$ Ans

$S_1 \rightarrow 0|2|4|6|8$, $S \rightarrow ABS_1$, $B \rightarrow 1|2|---|9$ ✓ (G.S.)

Q-(10.) KLP-(73) Construct context-free Grammar

(a.) $\{0^m 1^n \mid m \neq n, m, n \geq 1\}$ ✓ (G.S.)

Sol^n- $S \rightarrow 0S1|0A|1B|0|1$, $A \rightarrow 0A|0$, $B \rightarrow 1B|1$ Ans

Q-(10.) KLP-(73) (b.) $\{a^l b^m c^n \mid$ one of l, m, n equals 1 and the remaining two are equal$\}$

Sol^n- $S \rightarrow aS_1$, $S_1 \rightarrow bS_1c$, $S_1 \rightarrow bc$ ✓ (G.S.)

$S \rightarrow S_2c$, $S_2 \rightarrow aS_2b$, $S_2 \rightarrow ab$ Ans.

$S_3 \rightarrow b$, $S \rightarrow aS_3c$, $S_3 \rightarrow aS_3c$

Q-(10.)
KLP-(73) (c.) $\{0^m 1^n | 1 \le m \le n\}$

Soln:- S → 0S1, S → 01, S → 0A1, A → 1A,
A → 1

Ans.

S → 0S1
⇒ 00S11
⇒ 000A111
⇒ 0001A111
⇒ 00011111

S → 01

S → 0A1
⇒ 01A1
⇒ 011A1
⇒ 01111

S → 0A1
⇒ 011

S → 0S1
⇒ 0011

Q-(9.)
KLP(73) Soln:- S → 0 | S → 2 | S → 4 | S → 6 | S → 8

S → AS_1
⇒ $1S_1$
⇒ 10

S → AS_1
⇒ $2S_1$
⇒ 22

S → AS_1
⇒ $9S_1$
⇒ 98

S → AS_1
⇒ $8S_1$
⇒ 80

S → ABS_1
⇒ $1BS_1$
⇒ $19S_1$
⇒ 198

S → ABS_1
⇒ $9BS_1$
⇒ $99S_1$
⇒ 998 Ans

(10.)
LP(73) (a.) S → 0 | S → 1 | S → 0S1
Soln:-

⇒ 00S11
⇒ 000S111
⇒ 0000111
or
⇒ 0001111
or
⇒ 0000A111
⇒ 00000A111
⇒ 000000111
or
⇒ 0001B111
⇒ 00011B111
⇒ 000111111

S → 0A
⇒ 00A
⇒ 000A
⇒ 0000

S → 1B
⇒ 11B
⇒ 111B
⇒ 1111

Q-(9.)
Tamil(161.) Construct a C.F.G. for the language $L = \{w \mid w \in \{a,b\}^+\}$ consists of an equal number of a's and b's.

Sol^n:- $S \to aB$, $S \to bA$, $A \to a$, $A \to aS$, $A \to bAA$, $B \to b$, $B \to bS$, $B \to aBB$

$S \to a\underline{B}$
$\Rightarrow ab$

$S \to a\underline{B}$
$\Rightarrow ab\underline{S}$
$\Rightarrow aba\underline{B}$
$\Rightarrow abaa\underline{BB}$
$\Rightarrow abaab\underline{S}B$
$\Rightarrow abaabb\underline{A}B$
$\Rightarrow abaabba\underline{B}$
$\Rightarrow abaabbab$

$S \to bA$
$\Rightarrow ba$

$S \to b\underline{A}$
$\Rightarrow ba\underline{S}$
$\Rightarrow bab\underline{A}$
$\Rightarrow babb\underline{AA}$
$\Rightarrow babba\underline{S}A$
$\Rightarrow babbaa\underline{B}A$
$\Rightarrow babbaab\underline{A}$
$\Rightarrow babbaaba$

Ans.

Q-(11.)
KLP(73.) Construct regular grammars to generat the following:

(a.) $\{a^{2n} \mid n \geq 1\}$

Sol^n:- $S \to aS_1$, $S_1 \to aS$, $S \to aS_2$, $S_2 \to a$

$S \to a\underline{S_1}$
$\Rightarrow aa\underline{S}$
$\Rightarrow aaa\underline{S_2}$
$\Rightarrow aaaa$

$S \to aS_2$
$\Rightarrow aa$

$S \to a\underline{S_1}$
$\Rightarrow aa\underline{S}$
$\Rightarrow aaa\underline{S_1}$
$\Rightarrow aaaa\underline{S}$
$\Rightarrow aaaaa\underline{S_2}$
$\Rightarrow aaaaaa$

Ans.

(b.) The set of all strings over $\{a,b\}$ ending i

Sol^n:- $S \to aS$, $S \to bS$, $S \to a$

$S \to a$

$S \to a\underline{S}$
$\Rightarrow aa\underline{S}$
$\Rightarrow aab\underline{S}$
$\Rightarrow aabb\underline{S}$
$\Rightarrow aabba$

$S \to b\underline{S}$
$\Rightarrow bb\underline{S}$
$\Rightarrow bba\underline{S}$
$\Rightarrow bbaa\underline{S}$
$\Rightarrow bbaaa$

Ans.

Ex-(3.6) KLP(58) Let L be the set of all palindromes over $\{a, b\}$. Construct G generating L.

Sol^n- For any string to be the palindrome, it can be like the following:-

(i) $S \rightarrow \Lambda$, (ii) $S \rightarrow a \mid b$, (iii) $S \rightarrow aSa \mid bSb$

$S \rightarrow \Lambda$ $\Rightarrow \phi$ (String of Length zero)	$S \rightarrow a$ $S \rightarrow b$	$S \rightarrow aSa$ $\Rightarrow aaSaa$ $\Rightarrow aaaSaaa$ $\Rightarrow aaaaaaa$ or $\Rightarrow aaabaaa$ or $\Rightarrow aaaaaa$

$S \rightarrow bSb$
$\Rightarrow bbSbb$
$\Rightarrow bbbSbbb$
$\Rightarrow bbbbbbb$ or $\Rightarrow bbbabbb$ or $\Rightarrow bbbbbb$

So, $L(G) = \{a, b\}^*$ ✓ Ans.

Q-(15.)
KLP(74.) Show that {abc, bca, cab} can be generated by a regular grammar whose terminal set is {a, b, c}.

Sol^n:- $S \to aS_1$, $S_1 \to bS_2$, $S_2 \to c$, $S \to bS_3$, $S_3 \to cS_4$, $S_4 \to a$, $S \to cS_5$, $S_5 \to aS_6$, $S_6 \to b$

$S \to aS_1$	$S \to bS_3$	$S \to cS_5$
$\Rightarrow abS_2$	$\Rightarrow bcS_4$	$\Rightarrow caS_6$
$\Rightarrow abc$ ✓	$\Rightarrow bca$ ✓	$\Rightarrow cab$ ✓

Ans.

KLP(72) Find the language generated by the (49)
Grammar.

(a.) $S \rightarrow 0S1$, $S \rightarrow 0A1$, $A \rightarrow 1A$, $A \rightarrow 1$

Sol^n –

$S \rightarrow 0S1$
$\Rightarrow 00S11$
$\Rightarrow 000S111$
$\Rightarrow 0000A1111$
$\Rightarrow 00001A1111$
$\Rightarrow 0000111111$

$L(G) \Rightarrow 0^n 1^m$ Ans. GS.
$\Rightarrow \{0^n 1^m : m > n \geq 1\}$

$S \rightarrow 0S1$
$\Rightarrow 00A11$
$\Rightarrow 001A11$
$\Rightarrow 001111$
$L(G) \Rightarrow 0^n 1^m$ Ans. GS.

$S \rightarrow 0S1$
$\Rightarrow 00A11$
$\Rightarrow 00111$
$L(G) \Rightarrow 0^n 1^m$ Ans. GS.

(1)(KLP-72)
(b.) $S \rightarrow 0S1$, $S \rightarrow 0A$, $S \rightarrow 0$, $S \rightarrow 1B$, $S \rightarrow 1$
$A \rightarrow 0A$, $A \rightarrow 0$, $B \rightarrow 1B$, $B \rightarrow 1$

ol^n – $S \rightarrow 0S1$
$\Rightarrow 00S11$
$\Rightarrow 000A11$
$\Rightarrow 0000A11$
$\Rightarrow 0000011$

$S \rightarrow 0S1$
$\Rightarrow 00S11$
$\Rightarrow 001B11$
$\Rightarrow 0011B11$
$\Rightarrow 0011111$

$S \rightarrow 0A$
$\Rightarrow 00A$
$\Rightarrow 000$

$S \rightarrow 0$

$S \rightarrow 1$

$S \rightarrow 1B$
$\Rightarrow 11B$
$\Rightarrow 111$

$L(G) =$ ~~[illegible]~~ Ans. GS.
$(0+1)^+$

~~(1)(KLP-72)~~
C.) $S \rightarrow 0SBA$, $S \rightarrow 01A$, $AB \rightarrow BA$, $1B \rightarrow 11$,
$1A \rightarrow 10$, $0A \rightarrow 00$

ol^n – $S \rightarrow 0SBA$
$\Rightarrow 00SBABA$
$\Rightarrow 0001ABABA$
$\Rightarrow 00010BBAA$
$\Rightarrow$ No further derivation

$S \rightarrow 0SBA$
$\Rightarrow 001ABA$
$\Rightarrow 001BAA$
$\Rightarrow 0011AA$
$\Rightarrow 00110A$
$\Rightarrow 001100$

$S \rightarrow 01$
$\Rightarrow 01$

So, $L(G) = \{0^m 1^n : m > n \geq 1\}$ Ans. GS (d.) $S \rightarrow 0S1,\ S \rightarrow 0A1,\ A \rightarrow 1A0,\ A \rightarrow 10$ (50)

Solⁿ:-

$S \rightarrow 0S1$
$\Rightarrow 00S11$
$\Rightarrow 000A111$
$\Rightarrow 0001A0111$
$\Rightarrow 00011A00111$
$\Rightarrow 000111000111$

$S \rightarrow 0A1$
$\Rightarrow 01A01$
$\Rightarrow 011001$

So, $L(G) = \{0^n 1^n : n \geq 2\}$
Ans. (nS.)

~~Q-1-(KLP-72)~~

(e) $S \rightarrow 0A,\ S \rightarrow 1S,\ S \rightarrow 0,\ S \rightarrow 1,\ A \rightarrow 1A,$
$A \rightarrow 1S,\ A \rightarrow 1$

Solⁿ:-

$S \rightarrow 0A$
$\Rightarrow 01A$
$\Rightarrow 011S$
$\Rightarrow 0110A$
$\Rightarrow 01101$

$S \rightarrow 1S$
$\Rightarrow 11S$
$\Rightarrow 111S$
$\Rightarrow 1111$

$S \rightarrow 1S$
$\Rightarrow 11S$
$\Rightarrow 110$

$S \rightarrow 0$

$S \rightarrow 1$

So, $L(G) = (0+1)^*$ (GS)
Ans

Q-2(a) KLP-(73) Construct the Grammar, accepting the following -
The set of all strings over $\{0, 1\}$ consisting of equal number of 0's and 1's.

Solⁿ:- $G = (\{S, A, B\}, \{0, 1\}, P, S)$, where P consists of $S \rightarrow 0B,\ S \rightarrow 1A,\ A \rightarrow 0,\ A \rightarrow 0S,\ A \rightarrow 1AA,\ B \rightarrow 1,$
$B \rightarrow 1S,\ B \rightarrow 0BB$

* **Context Free Languages** - The language generated by the context-free grammars are called context-free languages. An important example for context-free languages is the syntax of programming languages. Context-free languages are also applied in the design of parser. It is also useful for describing block structure in programming languages.

* **Derivation Tree / Parse Tree** - The strings generated by a context-free grammar (V_n, Σ, P, S) can be represented by a hierarchical structure called tree. A hierarchical structure is that structure in which elements have some relationship with each other, e.g., if a node A is parent node of B than B is a child node of A.

A parse tree also called as derivation tree / syntax tree / generation tree / production tree for a context-free grammar G has the following characteristics:-

(i) Every vertex of parse tree has a label which is either a variable (upper case letter / Non-terminal) or a terminal (lower case letter) or Λ (null).

* Simplification of C

(ii) The root of parse tree labelled as S, is the start symbol.

(iii) The label of an internal vertex is always a variable.

(iv) A vertex n is a leaf if its label is a ∈ Σ or Λ; n is the only son of its father if its label is Λ (null).

e.g; we have the productions

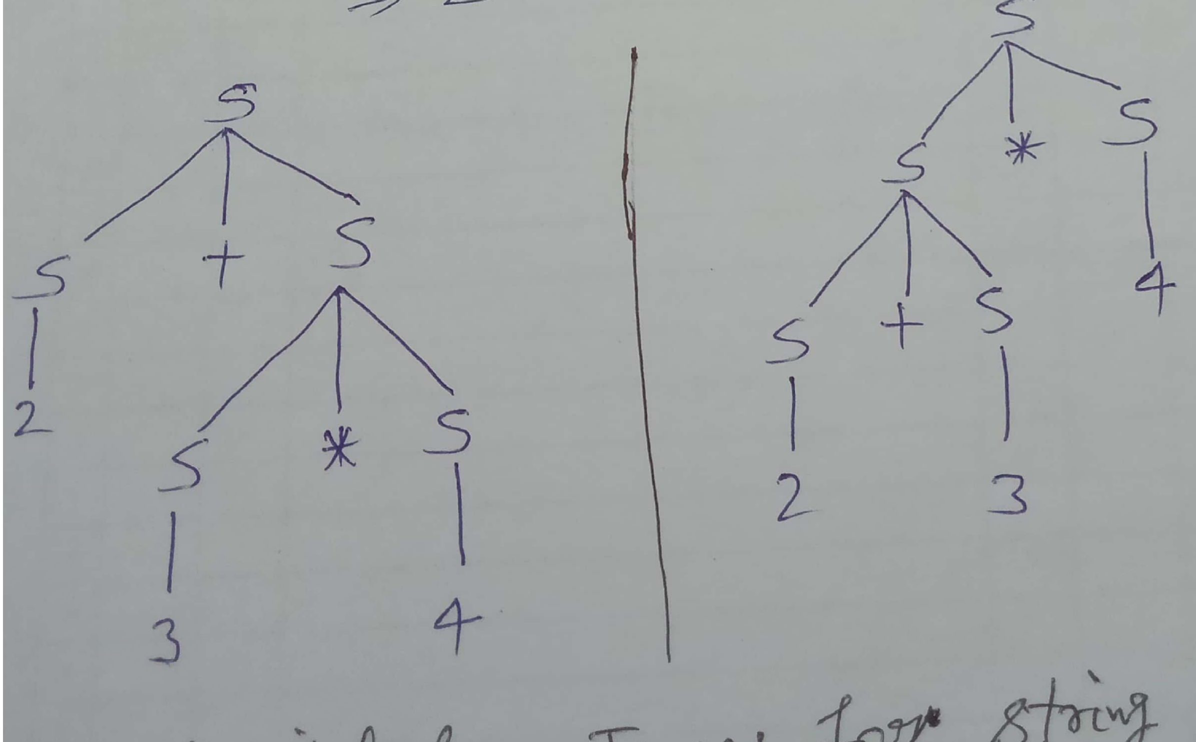
$S \rightarrow S+S \mid S*S \mid N$

Generate the string 2+3*4

Sol:-

$S \rightarrow S+S$
$\Rightarrow S+S*S$
$\Rightarrow N+N*N$
$\Rightarrow 2+3*4$

$S \rightarrow S*S$
$\Rightarrow S+S*S$
$\Rightarrow N+N*S$
$\Rightarrow 2+3*4$



Fig;- Possible Parse Trees for string 2+3*4

* Ambiguous Grammar

\- A context-free grammar G is ambiguous if there is atleast one string in L(G) having two or more distinct derivation trees

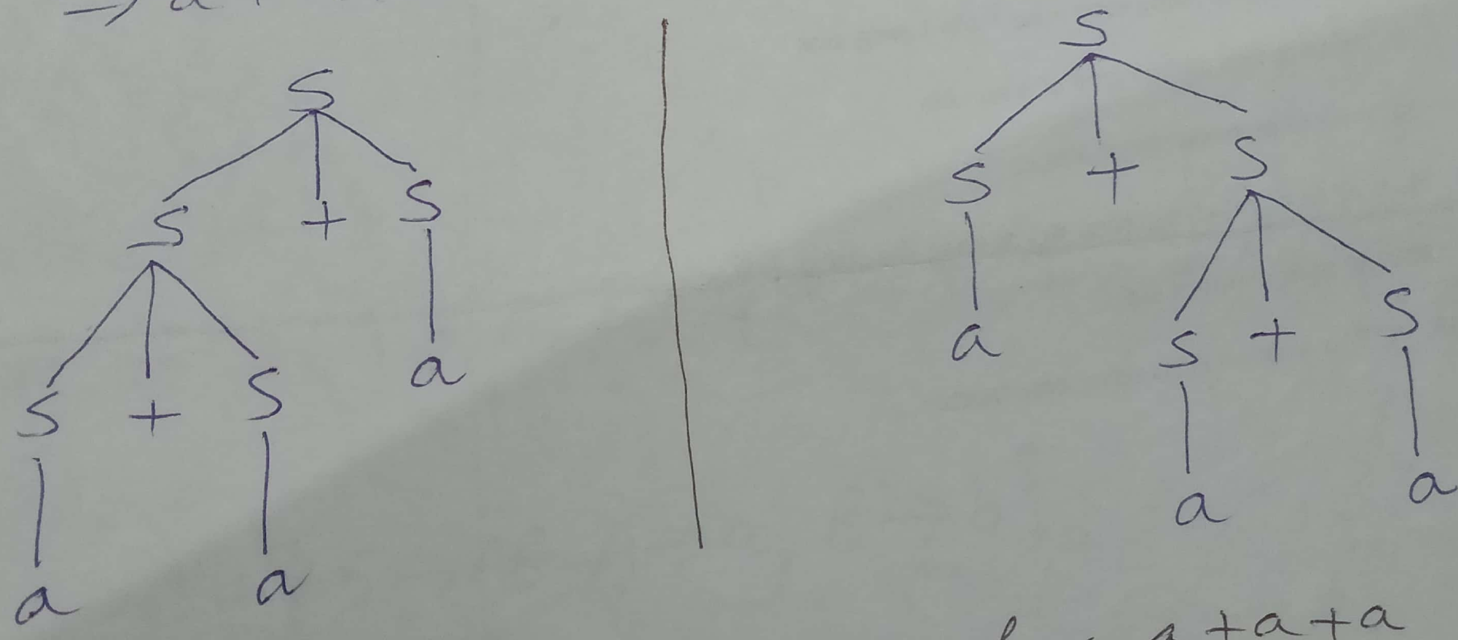
e.g; we have the productions

$$S \rightarrow S+S \mid a$$

Sol^n -

$$S \rightarrow \underline{S}+S$$
$$\Rightarrow S+S+S$$
$$\Rightarrow a+a+a$$

$$S \rightarrow S+S$$
$$\Rightarrow a+S$$
$$\Rightarrow a+S+S$$
$$\Rightarrow a+a+a$$



Fig;- Two Derivation Trees for a+a+a

* Simplification of Grammar - In a CFG, it may not be necessary to use all the symbols in $V_N \cup \Sigma$, or all the productions in P for deriving sentences. So, when we study a C.F.L. L(G), we try to eliminate symbols and productions in G which are not useful for derivation of sentences.

e.g.; $G = (\{S, A, B, C, E\}, \{a, b, c\}, P, S)$

where $P = \{S \rightarrow AB, A \rightarrow a, B \rightarrow b, B \rightarrow C, E \rightarrow c|\Lambda\}$

$S \rightarrow AB$
$\Rightarrow aB$
$L(G) \Rightarrow \{ab\}$

Let $G' = (\{S, A, B\}, \{a, b\}, P', S)$

where
$P' = \{S \rightarrow AB, A \rightarrow a, B \rightarrow b\}$
$S \rightarrow AB$
$\Rightarrow aB$
$L(G') \Rightarrow ab$

So, $L(G) = L(G')$

We have eliminated the symbols C, E and c and the productions ~~which~~ $B \rightarrow C$, $E \rightarrow c|\Lambda$ because

(i) C does not derive any terminal string

(ii) E and c do not appear in any sententia form.

(iii) $E \rightarrow \Lambda$ is a null production.

(iv) $B \rightarrow C$ simply replaces B by C.

* <u>Greibach Normal Form (GNF)</u> - A C.F.G. is in GNF if every production of the grammar is of the form:-

$$A \rightarrow a\alpha$$

where, $\alpha \in V_N^*$ and $a \in \Sigma$

<u>Basic Idea</u>:- The R.H.S. of every production must start with a terminal and only one terminal.

<u>Ques</u>.- Check, whether the following grammar is in G.N.F. If not, convert it into a G.N.F.

$$A \rightarrow a\underset{\textcircled{1}}{BD} \mid b\underset{\textcircled{2}}{DB} \mid \underset{\textcircled{3}}{\epsilon} \mid \underset{\textcircled{4}}{AB} \mid \underset{\textcircled{5}}{AD}$$

<u>Solⁿ</u>:- clearly, productions ①, ② and ③ conform to G.N.F. but ④ & ⑤ do not. So, we convert ④ & ⑤ into GNF.

<u>Approach</u>:- In order to get a terminal beginning for the productions on their R.HS in ④ & ⑤, we substitute ①, ② & ③ in them.

<u>Note</u>:- We can not substitute any one of them, we need to substitute all, one by one.

So, <u>Starting with ④</u> -

$A \rightarrow AB$

$A \rightarrow aBDB \mid bDBB \mid \epsilon B$

<u>Similarly, for ⑤</u> -

$A \rightarrow AD$

$A \rightarrow aBDD \mid bDBD \mid \epsilon D$

So, the final set of productions are:-

$A \rightarrow aBD \mid bDB \mid \epsilon \mid aBDB \mid bDBB \mid \epsilon B \mid aBDD \mid bDBD \mid \epsilon D$

<u>Ans.</u>

Ex-(1)

Pushdown Automata

aabb $a^n b^n$

$\delta(q_0, \epsilon, \epsilon) = (q_0, Z_0)$

Present state input tos (top of stack)

✓ $\delta(q_0, a, Z_0) = (q_1, aZ_0)$

$\delta(q_1, a, a) = (q_1, aaZ_0)$

$\delta(q_1, b, a) = (q_2, aZ_0)$

$\delta(q_2, b, a) = (q_3, Z_0)$

$\delta(q_3, \epsilon, Z_0) = (q_f, Z_0)$

Ans.

Push Down Automata (PDA)-

Definition:- A PDA $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F$

consists of 7-tuples.

(i) a finite non-empty set of states denote

by Q.

(ii) a finite non-empty set of input sym

denoted by Σ.

(iii) a finite non-empty set of pushdown sy

denoted by Γ.

(iv) a special state called the initial s

denoted by q_0.

(v) a special pushdown symbol called the

② initial symbol on the pushdown sto
denoted by Z_0.

(vi) the set of final states, a subset of Q denoted by F.

(vii) the transition function $\underline{\delta}$ from $Q \times (\Sigma \cup \{\Lambda\}) \times \Gamma$ to the set of finite subsets of $Q \times \Gamma^*$.

Ex-(6.6)
KLP(162) Construct a PDA A accepting the set of all strings over $\{a, b\}$ with equal no. of a's and b's.

Sol^n:- aaabbb

$\delta(q_0, a, Z_0) = (q_1, aZ_0)$

$\delta(q_1, a, a) = (q_1, aaZ_0)$

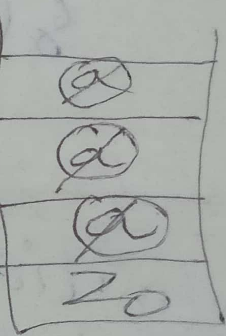
$\delta(q_1, a, a) = (q_1, aaaZ_0)$

$\delta(q_1, b, a) = (q_2, aaZ_0)$

$\delta(q_2, b, a) = (q_3, aZ_0)$

$\delta(q_3, b, a) = (q_4, Z_0)$

$\delta(q_4, \epsilon, Z_0) = (q_f, Z_0)$ Ans:



$\delta(q_0, a, Z_0) = (q_1, aZ_0)$ (a^n b^n by PDA) ③

$\delta(q_1, a, a) = (q_1, aaZ_0)$

$\delta(q_1, b, a) = (q_2, aZ_0)$

$\delta(q_2, b, a) = (q_3, Z_0)$

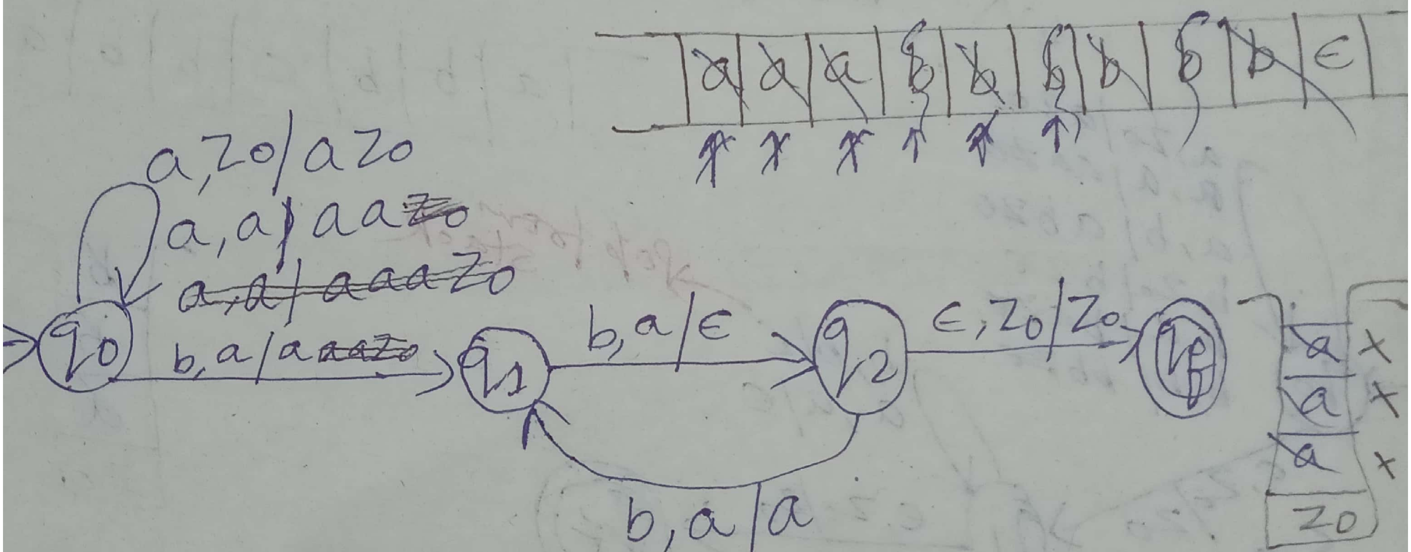
Acceptance by null stack (empty)

Acceptance by final stat

$\delta(q_3, \epsilon, Z_0) = (q_4, \epsilon)$

$\delta(q_3, \epsilon, Z_0) = (q_f, Z$

(2.) $L = \{a^n b^{2n} \mid n \geq 1\}$ by PDA



$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aaZ_0)$

$\delta(q_0, a, a) = (q_0, aaaZ_0)$

$\delta(q_0, b, a) = (q_1, aaaZ_0)$

$\delta(q_1, b, a) = (q_2, aaZ_0)$

$\delta(q_2, b, a) = (q_1, aaZ_0)$

$\delta(q_1, b, a) = (q_2, aZ_0)$

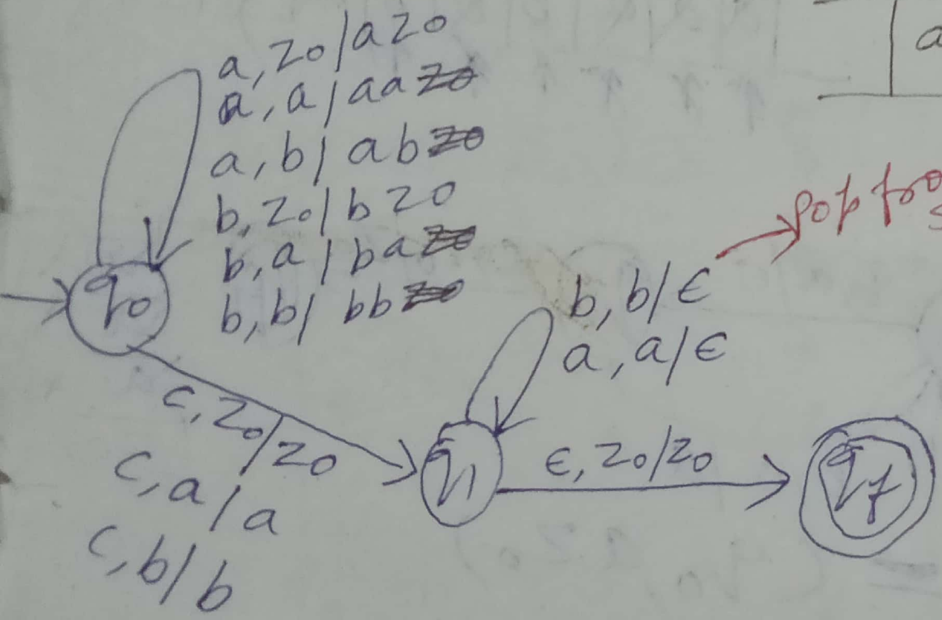
$L = \{wcw^R \mid w \in \{a,b\}^*\}$ acceptibility by PDA.

Sol^n_

$w = abc$

$w^R = cba$ (Reverse of string)

a	b	b	c	b	b	a	∈



~~b~~
~~b~~
~~a~~
Z0

Ans.

$\delta(q_0, a, Z_0) = (q_1, aZ_0)$

$\delta(q_1, b, a) = (q_1, baZ_0)$

$\delta(q_1, b, b) = (q_1, bbaZ_0)$

$\delta(q_1, c, b) = (q_2, bbaZ_0)$

$\delta(q_2, b, b) = (q_3, baZ_0)$

$\delta(q_3, b, b) = (q_4, aZ_0)$

$\delta(q_4, a, a) = (q_5, Z_0)$

$\delta(q_5, \epsilon, Z_0) = (q_f, Z_0)$ Accepted by Final State

or

$\delta(q_5, \epsilon, Z_0) = (q_5, Z_0)$,, ,, Null Stack (Empty)