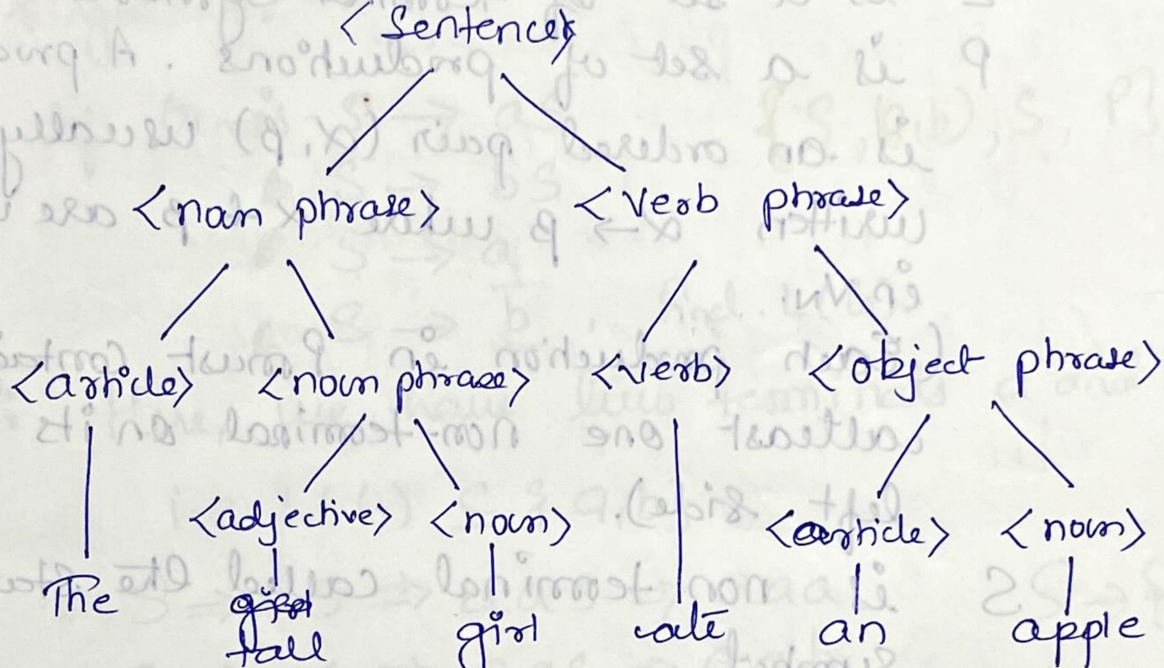
GRAMMAR :-

The set of rules used to define the language is called grammar.

Grammatical construction of a specific sentence-



There are :-

1) Various variables eg:- <Sentence>, <noun phrase>...

2) Various terminal words - The, an, ...

3) A beginning variable <Sentence> and

4) Various substitutions or productions eg:-

<Sentence> → <noun phrase> <Verb phrase>

<Object phrase> → <article><noun>

<noun> → apple

The final sentence contains only terminals, although both variables and terminals appear in its construction by the production.

Hence,
wi

A phrase structure grammar (or simpley grammar) G is defined by 4-tuples:

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

where

V_N is a set of non terminal symbols.

Σ is a set of terminal symbols.

P is a set of productions. A production is an ordered pair (α, β) usually written $\alpha \to \beta$ where α & β are words in V_N.

(Each production in P must contain atleast one non-terminal on its left side).

S is non terminal called the start symbol.

Example:-

Let $G = (\{S, C\}, \{a, b\}, P, S)$;

where P consists of

$S \to aCa$

$C \to aCa$

$C \to b$

Find the language described by grammar G.

Soln:- Let S is the start symbol.

$S \Rightarrow aCa \Rightarrow aba \in L(G)$

$S \Rightarrow aCa \Rightarrow aaCaa \Rightarrow aaaCaaa \Rightarrow a^nCa^n$

$\Rightarrow a^nba^n$

Hence, the language defined by above production will be;

$L(G) = a^n b a^n$ where $n \geq 1$.

Hence, this language consist of equal no. of leading and trailing a's separated by single b.

(2) If G is $S \to aS$ $\{S, (a,b), S, P\}$
$S \to bS$
$S \to a$
$S \to b$, find L(G).

Solⁿ: Here, we have two terminals a and b.

i.e $L(G) \subseteq \{a,b\}$

$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaa$ $S \to \{a,b\}^+$

$S \Rightarrow aS \Rightarrow aaS \Rightarrow aab$

$S \Rightarrow aS \Rightarrow aa$ $S \to aS|bS|a|b$

$S \Rightarrow aS \Rightarrow ab$

$S \Rightarrow bS \Rightarrow ba$

$S \Rightarrow bS \Rightarrow bb$

$S \Rightarrow bS \Rightarrow bb$

$S \Rightarrow bS \Rightarrow baS \Rightarrow bab$
$\Rightarrow baa$

$S \Rightarrow bS \Rightarrow bbS \Rightarrow bbb$
$\Rightarrow bba$

$L(G) = \{a,b\}^+$
because, $S \to \phi$ is not present.

③ Let L be the set of all palindromes over {a, b}. Construct generating L.

Soln:

To, generate the required grammar, we use the recursive definition,

i) ϕ is a palindrome.

ii) a, b are palindromes.

iii) If z is a palindrome, then aza, bzb are palindromes.

Let $G = \{\{S\}, \{a, b\}, P, S\}$

then,

$S \to \phi$

$S \to a$

$S \to b$

$S \to asa$ ~~bsb~~

$S \to bsb$

$\Rightarrow S \to \phi \mid a \mid b \mid asa \mid bsb$

④ Construct a grammar generating

$L = \{wcw^T ; w = \{a, b\}^*\}$

Soln:

$G = \{\{S\}, \{a, b, c\}, P, S\}$

where P consists of,

$S \to aSa \mid bSb \mid c$

Find a grammar generating
$L = \{a^n b^n c^i ; n \geq 1, i \geq 0\}$

Solⁿ: $S \to A$ $\qquad S \Rightarrow A \Rightarrow ab$ $\qquad ab, abc, aabbc, abcc.$

$A \to ab \mid aAb$ $\qquad S \Rightarrow Sc \Rightarrow Ac \Rightarrow abc$

$S \to Sc$

we defined P as $S \to Sc \mid A$
$A \to ab \mid aAb$

⑥ Construct grammar, accepting each of the following sets:

(i) $\{0^n 1^m 0^m 1^n ; m, n \geq 1\}$ (ii) $\{0^n 1^{2n} : n \geq 1\}$

Solⁿ: i) $\{0^n 1^m 0^m 1^n ; m, n \geq 1\}$ $\qquad$ RL / CFL / Re.L

$m, n = 1$ $\quad 0101$

$m = 2, n = 1$ $\quad 011001$

$m = 1, n = 2$ $\quad 001011$

$m = 3, n = 1$ $\quad 01110001$

$S \Rightarrow 0S1 \Rightarrow 00A11 \Rightarrow 001011$

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

$S \to 0S1 \mid 0A1$

$A \to 10 \mid \cancel{0A0}\ 1A0$

$S \to 0S1 \mid 0A1$

$A \to 10 \mid 1A0$

ii> $\{0^n 1^{2n} : n \geq 1\}$

$n=1$ 011 $n=3$ 00011111

$n=2$ 001111

$S \to 0S11 \mid 011$

⑦ Find a grammar generating
$\{a^j b^n c^n : n \geq 1,\ j \geq 0\}$

Soln: Let $G = \{\{S, A\}, \{a, b, c\}, P, S\}$

$n=1, j=0$ bc $n=3$ $j=0$ bbbccc

$n=1, j=1$ abc $n=3$ $j=1$ abbbccc

$n=2, j=1$ abbcc

$n=2, j=2$ aabbcc $n=3$ $j=2$ aabbbccc

$S \to aS$

$S \to A$

$A \to bc \mid bAc$

The production P will be,

$S \to aS \mid A$

$A \to bc \mid bAc$

Construct a grammar G generating $\{a^n b^n c^n : n \geq 1\}$

Soln:

$n=1$ abc

$n=2$ aabbcc

$n=3$ aaabbbccc

$S \to ASBC \mid aBC$, $CB \to BC$

$A \to a$

$B \to b$

$C \to c$.

$S \Rightarrow ASBC \Rightarrow AASBCBC$

$\Rightarrow AAaBCBCBC$

$\Rightarrow AAaBBCBCC$

$\Rightarrow AAaBBBCCC$

$\Rightarrow aaabbbccc$

$S \Rightarrow ASBC \Rightarrow AASBCBC \Rightarrow AAASBCBCBC$

$\Rightarrow AAAaBCBCBCBC$

$\Rightarrow AAAaBBCBCBCC$

$\Rightarrow AAAaBBBCBCCC$

$\Rightarrow AAAaBBBBCCCC$

$\Rightarrow aaaabbbbcccc$

So, the production P is

$S \to ASBC \mid aBC$

$CB \to BC$

$A \to a$

$B \to b$

$C \to c$

⑧ Give grammar G to represent language L with words consisting of,

a) equal no. of a's & b's
b) palindromes of odd length
c) palindromes of even length
d) Palindromes
e) atleast two a's
f) without consecutive occurence of b's.
g) atleast one occurence of 'aaa'.

Soln: ⓐ $S \rightarrow aB \mid bA$

$A \rightarrow a \mid aS \mid bAA$

$B \rightarrow b \mid bS \mid aBB$

ⓑ $S \rightarrow a \mid b \mid aSa \mid bSb$

ⓒ $S \rightarrow \phi \mid aSa \mid bSb$

ⓓ $S \rightarrow \phi \mid a \mid b \mid aSa \mid bSb \mid$

ⓔ $S \rightarrow AaAaA$

$A \rightarrow aA \mid bA \mid a \mid b \mid \phi$

ⓕ $S \rightarrow aS \mid bA \mid a \mid b \mid \wedge$

$A \rightarrow a \mid aS \mid \phi$

ⓖ $S \rightarrow AB$

$B \rightarrow aaa$

$A \rightarrow aA \mid bA \mid \phi$

S → A | aA | bA

A → aaa | aA

B →

Q. Find the language generated by the following grammars!

(a) $S \to 0S1 | 0A | 0 | 1B | 1$

$A \to 0A | 0$

$B \to 1B | 1$

Soln:- $S \Rightarrow 0S1 \Rightarrow 00A1 \Rightarrow 000A1 \Rightarrow 00001$

$00A1 \Rightarrow 0001$

$0S1 \Rightarrow 001$

$0S1 \Rightarrow 01B1 \Rightarrow 011B1 \Rightarrow 01111$

$01B1 \Rightarrow 0111$

$0S1 \Rightarrow 011$

$L(G) = 0^m 1^n$ where $m, n \geq 1$.

(b) $S \to 0S1 | 0A1$

$A \to 1A0 | 10$

Soln:- $S \Rightarrow \cancel{0S1}\ 0S1 \Rightarrow 00A11 \Rightarrow 001A011 \Rightarrow 00110011$

$00A11 \Rightarrow 001011$

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

$0A1 \Rightarrow 011001$

$S \Rightarrow 0S1 \Rightarrow 00A11 \Rightarrow 001A011$

$\cancel{\Rightarrow 0011A001}$

$\cancel{\Rightarrow 0011100011}$

$\Rightarrow 0011A0011$

$\Rightarrow \underbrace{00}_{m}\underbrace{11}_{n}1\underbrace{00}_{n}\underbrace{11}_{m}$

$L(G) = 0^m 1^n 0^n 1^m$ where $m, n \geq 1$

Q: Find the language for the specified grammar:-

$G = (V_N, \Sigma, P, S)$ where $\Sigma = \{a\}$, $V_N = \{S, N, Q, R\}$ and P consists of,

$S \to QNQ$

$QN \to QR$

$RN \to NNR$

$RQ \to NNQ$

$N \to a$

$Q \to \phi$

$\Rightarrow S \Rightarrow QNQ$

$\Rightarrow QRQ$

$\Rightarrow QNNQ$

$\Rightarrow QRNQ$

$\Rightarrow QNNRQ$

$\Rightarrow QNNNNQ$

$\Rightarrow aaaa$

$S \Rightarrow QNQ \Rightarrow a$

$S \Rightarrow QNQ$

$\Rightarrow QRQ$

$\Rightarrow QNNQ$

$\Rightarrow aa$

$S \Rightarrow \cancel{QNQ}$

~~$S \Rightarrow QNNNNQ$~~

~~$\Rightarrow QRNNNQ$~~

~~$\Rightarrow QNNRNNQ$~~

~~$\Rightarrow$~~

$$\boxed{L(G) = a^{2^n} \text{ where } n \geq 0}$$

Introduction	TM
R.L	Undecidability & Recursively Enumerable lang.
FA	Comparison of scope of languages & machines
RL	
CFG	
PDA	

Grammar

A grammar **G** can be formally written as a 4-tuple (N, T, S, P) where −

- **N** or V_N is a set of variables or non-terminal symbols.
- **T** or Σ is a set of Terminal symbols.
- **S** is a special variable called the Start symbol, $S \in N$
- **P** is Production rules for Terminals and Non-terminals. A production rule has the form $\alpha \rightarrow \beta$, where α and β are strings on $V_N \cup \Sigma$ and least one symbol of α belongs to V_N.

Example

Grammar G1 −

$(\{S, A, B\}, \{a, b\}, S, \{S \rightarrow AB, A \rightarrow a, B \rightarrow b\})$

Here,

- **S, A,** and **B** are Non-terminal symbols;
- **a** and **b** are Terminal symbols
- **S** is the Start symbol, $S \in N$
- Productions, $\mathbf{P : S \rightarrow AB, A \rightarrow a, B \rightarrow b}$

Example

Grammar G2 −

$((\{S, A\}, \{a, b\}, S, \{S \rightarrow aAb, aA \rightarrow aaAb, A \rightarrow \varepsilon\})$

Here,

- **S** and **A** are Non-terminal symbols.
- **a** and **b** are Terminal symbols.
- ε is an empty string.
- **S** is the Start symbol, $S \in N$
- Production $\mathbf{P : S \rightarrow aAb, aA \rightarrow aaAb, A \rightarrow \varepsilon}$

$S \Rightarrow aAb$
$\Rightarrow ab \quad (\because A \rightarrow \epsilon)$

$S \Rightarrow aAb$
$\Rightarrow aaAbb \, (aA \rightarrow aaAb)$
$\Rightarrow aabb$

$S \Rightarrow aAb$
$\Rightarrow a\underline{aA}bb \quad (\because A \rightarrow aaAb)$
$\Rightarrow a\underline{aaAb}bb$
$\Rightarrow aaabbb$

Derivations from a Grammar

Strings may be derived from other strings using the productions in a grammar. If a grammar **G**has a production $\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta}$, we can say that $\mathbf{x\ \alpha\ y}$ derives $\mathbf{x\ \beta\ y}$ in **G**. This derivation is written as −

$$x\,\alpha\,y \Rightarrow G\, x\,\beta\,y$$

Example

Let us consider the grammar –

G2 = ({S, A}, {a, b}, S, {S → aAb, aA → aaAb, A → ε })

Some of the strings that can be derived are –

S ⇒ aAb using production S → aAb

⇒ aaAbb using production aA → aAb

⇒ aaaAbbb using production aA → aAb

⇒ aaabbb using production A → ε

S ⇒ aAb

⇒ aaaAbb (aA → aaAb)

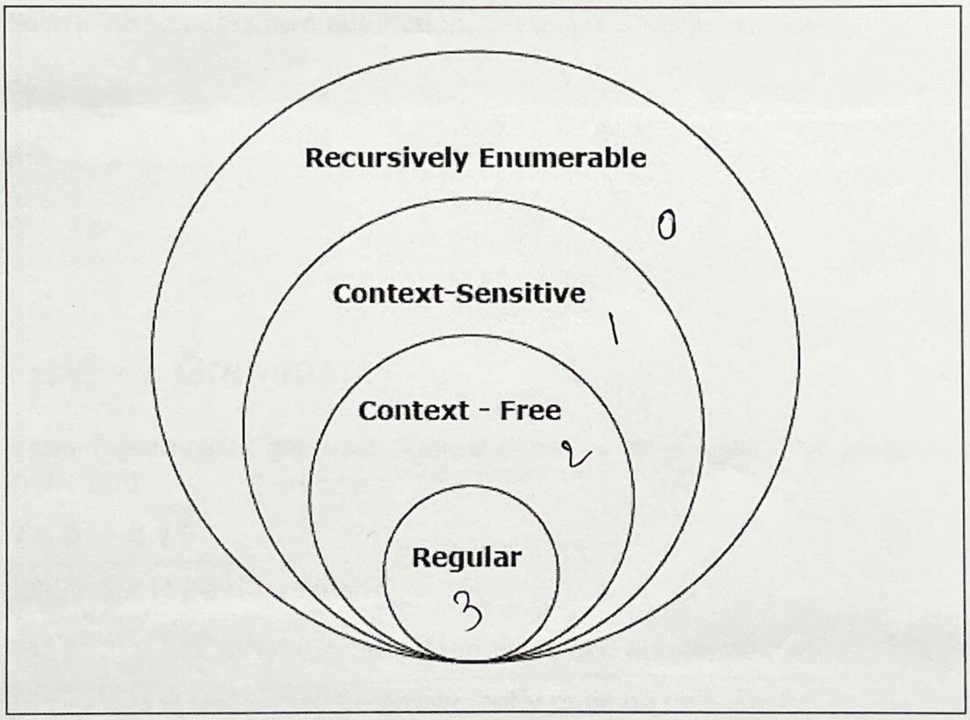
⇒ aaaAbb (aA → aaAb)

⇒ aaa

According to Noam Chomosky, there are four types of grammars − Type 0, Type 1, Type 2, and Type 3. The following table shows how they differ from each other −

Grammar Type	Grammar Accepted	Language Accepted	Automaton
Type 0	Unrestricted grammar	Recursively enumerable language	Turing Machine
Type 1	Context-sensitive grammar	Context-sensitive language	Linear-bounded automaton
Type 2	Context-free grammar	Context-free language	Pushdown automaton
Type 3	Regular grammar	Regular language	Finite state automaton

Take a look at the following illustration. It shows the scope of each type of grammar −



Type - 3 Grammar

Type-3 grammars generate regular languages. Type-3 grammars must have a single non-terminal on the left-hand side and 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 **X → a or X → aY** where **X, Y ∈ N** (Non terminal)

and **a ∈ T** (Terminal)

The rule **S → ε** is allowed if **S** does not appear on the right side of any rule.

Example

```
X → ε
X → a | aY
Y → b
```

Type - 2 Grammar

Type-2 grammars generate context-free languages.

The productions must be in the form **A → γ**

where **A ∈ N** (Non terminal)

and **γ ∈ (T ∪ N)*** (String of terminals and non-terminals).

These languages generated by these grammars are be recognized by a non-deterministic pushdown automaton.

Example

```
S → X a
X → a
X → aX
X → abc
X → ε
```

Type - 1 Grammar

Type-1 grammars generate context-sensitive languages. The productions must be in the form

α A β → α γ β

where **A ∈ N** (Non-terminal)

and **α, β, γ ∈ (T ∪ N)*** (Strings of terminals and non-terminals)

The strings **α** and **β** may be empty, but **γ** must be non-empty.

The rule **S → ε** is allowed if S does not appear on the right side of any rule. The languages generated by these grammars are recognized by a linear bounded automaton.

Example

```
AB → AbBc
```

```
A → bcA
B → b
```

Type - 0 Grammar

Type-0 grammars generate recursively enumerable languages. The productions have no restrictions. They are any phase structure grammar including all formal grammars.

They generate the languages that are recognized by a Turing machine.

The productions can be in the form of **α → β** where **α** is a string of terminals and nonterminals with at least one non-terminal and **α** cannot be null. **β** is a string of terminals and non-terminals.

Example

```
S → ACaB
Bc → acB
CB → DB
aD → Db
```

Q: CONTEXT FREE GRAMMARS

A Context free grammar (CFG) is defined as 4- tuples (V_N, Σ, P, S)

where P is a set of production rules of the form:-

One Non Terminal → finite String of terminals and/or non-terminals

The languages generated by the Context-free grammars are called Context-free languages. Context free languages are applied in Parser design. CFG's are useful for describing arithmetic expressions, with arbitrary nesting of balanced parenthesis.

Every regular grammar is context-free so a regular language is also context free one. But context-free languages are more complicated than regular languages. We did not define a 'deterministic' 'regular language'

Family of regular languages is a proper subset of the family of the context free languages.

Q: $a^n b^m$ ~~where (n)~~ in any order of a & b
where ~~a+b~~ $n \neq m$.

Generate String:-
aabaaabbbaaa

Sol^n: Case 1:- $n > m$ Case 2! $n < m$

$S \to U \mid T$

$U \to aUb \mid bUa \mid aU \mid Ua \mid a \mid UU$

$T \to aTb \mid bTa \mid bT \mid Tb \mid b \mid TT$

Q: CFG for arithmetic expressions - Derive string $(x+20)*y/(z-6.0)$

$S \to S+S \mid S*S \mid S-S \mid S/S \mid (S) \mid \text{variable} \mid \text{Constant}$

$S \Rightarrow S*S \Rightarrow (S)*S$

$\Rightarrow (S+S)*S$

$\Rightarrow (x+S)*S \Rightarrow (x+20)*S$

$\Rightarrow (x+20)*S/S \Rightarrow (x+20)*y/S$

$\Rightarrow (x+20)*y/(S)$

$\Rightarrow (x+20)*y/(S-S)$

$\Rightarrow (x+20)*y/(z-S)$

$\Rightarrow (x+20)*y/(z-6.0)$

Q: $L = \{aa, ab, ba, bb\}$

$S \to aa \mid ab \mid ba \mid bb$

$\underbrace{(a+b)}_{A} \cdot \underbrace{(a+b)}_{A}$

$S \to AA$
$A \to a \mid b$

Q: $a^n \mid n \geq 1$ or a^+

$a, aa, aaa, \ldots$

$S \to aA \mid a$
$A \to a \mid aA$

Q: $a^n \mid n \geq 0$ or a^*

$\lambda, a, aa, aaa, \ldots$

$S \to aA \mid a \mid \epsilon$
$A \to a \mid aA$

$A \to aA \mid \epsilon$

Q: $(a+b)^*$

$S \to aS \mid bS \mid \epsilon$

Q: Atleast two symbols.

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

$S \to AAB$
$A \to a \mid b$ $\quad B \to aB \mid bB \mid \epsilon$

Q: Atmost two symbols.

$(a+b+\epsilon)(a+b+\epsilon)$

$S \to AA$
$A \to a \mid b \mid \epsilon$

Question Bank

Q: Construct a CFG to generate any no. of a's. $\Sigma = \{a\}$.

Soln: $S \rightarrow aS \mid \lambda$

Q: Construct a CFG for $L = (a^n b^n)$ $n \geq 0$.

Soln: $S \rightarrow aSb \mid \lambda$

Q: Construct a CFG for the set of all even palindromes over $\Sigma = \{a, b\}$. (ww^R)

Generate String

~~aaabbb~~

aabbaa

Soln: $S \rightarrow aSa \mid bSb \mid \lambda$

Q: Construct a CFG for the set of all odd palindromes over $\Sigma = \{a, b\}$. waw^R

Generate String

aabbbbbaa

Soln: $S \rightarrow aSa \mid bSb \mid a \mid b$

Generate String

aaabbbb

Q: ~~$a^n b^n$ $n \neq$~~ $a^n b^{n+1}$ $n \geq 0$

$S \rightarrow aSb \mid b$

Q: $a^n b^{2n}$

$S \rightarrow aSbb \mid \lambda$

Generate String

aabbbb

Q: first symbol a and last symbol b, any symbol in middle

$a(a+b)^* b$

$S \rightarrow aAb$

$A \rightarrow aA \mid bA \mid \epsilon$

Generate String :- aaabb

Q:- $a^n b^m$ $n > m$. eg:- aab, aaabb, aa

$S \to aSb \mid aS \mid a$

Q:- $a^n b^m$ $n \neq m$

Generate String aaabbbbb

Case 1:- where $n > m$

Case 2:- where $n < m$

$S \to T \mid U$

$T \to aTb \mid aT \mid a$

$U \to aUb \mid Ub \mid b$

Q:- $a^n b^m c^K$ where $K = n+m$

Generate String aabbb ccccc

eg:- $n=2$, $m=3$, $K=5$

aabbb ccccc

$a^n b^m c^K = a^n b^m c^m c^n$

$S \to aSc \mid T$

$T \to bTc \mid \lambda$

Q:- $a^n b^n$ in any order of a & b | for every a there will be b.

eg:- ab, abab, aababb

Generate string aababb

$S \to aSb \mid SS \mid \lambda$

Q: $a^n b^n$ in any order of a & b.

eg:- ab, abba, abbaab

$S \to aSb \mid bSa \mid SS \mid \lambda$

aaSbaSbb

| $S \to aSbS \mid bSaS \mid \lambda$

Q1:- Construct a context-free grammar G generating all ~~strings~~ integers (with sign).

Soln: $G = (V_N, \Sigma, P, S)$

$V_N = \{S, \langle sign \rangle, \langle integer \rangle\}$

$\Sigma = \{0, 1, 2, 3, \ldots 9, +, -\}$

Production rules:-

$P \rightarrow \langle sign \rangle \langle integer \rangle$

$\langle integer \rangle \rightarrow \langle integer \rangle \langle integer \rangle \mid \langle integer \rangle$

$\langle integer \rangle \rightarrow 0|1|2|3|4|5|6|7|8|9$

$L(G)$ is the set ~~of~~ all integers.

Q2:- Construct a CFG for the language

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

Soln: Let G be the CFG.

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

$V_N = \{S\}, \Sigma = \{a, b\}$

Possible strings for the given language are:-

P: 1) $S \rightarrow aSb$

$S \rightarrow ab$

Q:- Different start and last symbol $\Sigma = \{a, b\}$.

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

$S \rightarrow aAb \mid bAa$

$A \rightarrow aA \mid bA \mid \epsilon$

Q:- Construct a CFG for the language
$L = \{ w c w^R : w \in (a,b)^* \}$

Soln:- $G = (V_N, \Sigma, P, S)$

$V_N = \{S\}$

$\Sigma = \{a, b\}$

Production Rule P: $S \to aSa$

$S \to bSb$

$S \to c$

Q:- Write a CFG, which generates string of balanced parenthesis.

Soln:- $V_N = \{S\}$ $\Sigma = \{(,)\}$

Production Rule P :-

$S \to SS$

$S \to (S)$

$S \to \Lambda$

Q:- Write a CFG, which generates palindrome for binary number.

Soln:- i.e 00, 11, 0, 1, 010, 101, 01010, 0110

$S \to 0S0$

$S \to 1S1$

$S \to 0|1|\epsilon$

Q:- Write a CFG, which generates strings having equal no. of a's & b's.

$S \rightarrow aSbS$

$S \rightarrow bSaS$

$S \rightarrow \wedge$

Q:- Design a CFG which can generate string, having any combination of a's & b's except null string.

Solⁿ:- $V_N = \{S\}$ $\Sigma = \{a, b\}$

P:- $S \rightarrow aS$

$S \rightarrow bS$

$S \rightarrow a$

$S \rightarrow b$

Q:- Design a CFG for the language $L(G) = \{ww^R : w \in \{0,1\}^*\}$

Solⁿ:- $G = \{V_N, \Sigma, P, S\}$

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

P:-

$S \rightarrow 0S0$

$S \rightarrow 1S1$

$S \rightarrow \wedge$

Q:- Design a CFG for the language $L = \{a^n b^m : n \neq m\}$

Solⁿ:- If $n \neq m$ then there are two possible cases-

i) $n > m$	aab, aaabb, aaab
ii) $m > n$	abb, aabbb, abbb

Case 1:- $n > m$ then $L_1 = \{a^n b^m : n > m\}$

Suppose G_1 is the CFG.

$G_1 = (V_N', \Sigma', P', S')$

$V_N' = \cancel{\{S_1, A, B\}}\ \{S_1, A, S'\}$

| aab
| aaabb
| aaab

~~A → BSq~~

$S' \rightarrow AS_1$

$S_1 \rightarrow aS_1b \mid \Lambda$

$A \rightarrow a \mid aA$

Case 2:- $m > n$ then let $L_2 = \{a^n b^m : m > n\}$

Suppose G_2 be the CFG.

$G_2 = (V_N'', \Sigma'', P'', S'')$

$V_N'' = \{S_2, B, S''\}$

| abb
| aabbb
| abbb
| aaabbbb

$S'' \rightarrow S_2 B$

$S_2 \rightarrow aS_2b \mid \Lambda$

$B \rightarrow bB \mid b$

By Combining G_1 & G_2, we can write CFG for the given language L,

i.e

$$\left.\begin{aligned}
&S \rightarrow S' \mid S'' \\
&S' \rightarrow AS_1 \\
&S_1 \rightarrow aS_1b \mid \Lambda \\
&A \rightarrow aA \mid a \\
&S'' \rightarrow S_2 B \\
&S_2 \rightarrow aS_2b \mid \Lambda \\
&B \rightarrow bB \mid b
\end{aligned}\right\} \text{CFG}$$

Q:- Write the CFG for the language
$L = \{a^{2n} b^m : n \geq 0, m \geq 0\}$

Generate string
aaaaaa bbbb

Solⁿ:-

$S \to AB \mid \lambda$

$A \to aaA \mid \lambda$

$B \to bB \mid \lambda$

Q:- Find Context free grammars for the following languages with $n \geq 0$, $m \geq 0$.

i) $L = \{a^n b^m : n \leq m+3\}$ ii) $L = \{a^n b^m : 2n \leq m \leq 3n\}$

Solⁿ:- i) $L = \{a^n b^m : n \leq m+3\}$

Let us consider for $n = m+3$ then add b's.

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

Generate string
aaaaa bbb

where, $V_N = \{S, A, B\}$

$\Sigma = \{a, b, \lambda\}$

Productions are:-

$S \to aaaA$

$A \to aAb \mid B$

$B \to Bb \mid \lambda$

But this production will create atleast 3 a's.

So, we will consider the case when $n = 1, 2$ or 0.

$S \to aA \mid aaA \mid aaaA \mid \lambda$

$A \to aAb \mid B$

$B \to Bb \mid \lambda$

ii) $S \to \lambda \mid aSbb \mid aSbbb$

$n = 0 \quad m = 0$

$n = 1 \quad m = 2, 3$

$n = 2 \quad m = 4, 5, 6 \quad 4 \leq m \leq 6$

Q:- Construct a CFG for the language containing all the strings of different first + last symbol over $\Sigma = \{0,1\}$.

Solⁿ:- $S \to 0A1 \mid 1A0$

$A \to 0A \mid 1A \mid \wedge$

Q:- Construct CFG for the language containing atleast one occurence of a double a. over $\{a,b\}$

$(a+b)^* \, aa \, (a+b)^*$

Solⁿ:- $S \to BAB$

$A \to aa$

$B \to aB \mid bB \mid \wedge$

Q:- Construct CFG for the language in which there are no cosecutive b's, the string may or may not have consecutive a's. Generate string aabaaba

Solⁿ:

~~$S \to ABA$~~

~~$B \to b$~~

~~$A \to Aa \mid bA \mid a \mid \wedge$~~

~~$S \Rightarrow ABA$~~

~~$\Rightarrow AbA$~~

~~$\Rightarrow AabAa \Rightarrow bAbbA$~~

~~$\Rightarrow bAabbAa \Rightarrow$~~

~~$\Rightarrow bab$~~

$S \to aS \mid bA \mid a \mid b \mid \wedge$

$A \to aS \mid a \mid \wedge$

$S \Rightarrow aS$

$\Rightarrow aaS$

$\Rightarrow aabA$

$\Rightarrow aabaS$

$\Rightarrow aabaaS$

$\Rightarrow aabaabA$

$\Rightarrow aabaaba$

Q: Find CFG for each of the following :-

i) ab^*

$S \to aB$

$B \to \epsilon \mid bB$

ii) $a^* b^*$

$S \to AB$

$A \to aA \mid a \mid \wedge$

$B \to bB \mid b \mid \wedge$

iii) $(baa + abb)^*$

$S \to AS \mid BS \mid \wedge$

$A \to baaA \mid baa$

$B \to abbB \mid abb$

$S \Rightarrow AS$
$\Rightarrow baaAS$
$\Rightarrow baabaaS$
$\Rightarrow baabaa$

$S \Rightarrow BS$
$\Rightarrow BAS$
$\Rightarrow BABS$
$\Rightarrow abbbaaabb$

$S \Rightarrow BS$
$\Rightarrow abbBS$
$\Rightarrow abbS$
$\Rightarrow abbAS$
$\Rightarrow abbbaaS$
$\Rightarrow abbbaaBS$
$\Rightarrow abbbaaabb$

Q: The Context free Grammar is given as $S \to aSb \mid ab$. Find the Context free language generated by the above grammar.

Sol^n: $S \to aSb \mid ab$

$S \Rightarrow ab$

$S \Rightarrow aSb$
$\Rightarrow aaSbb$
$\Rightarrow aaabbb$

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

Q: Find the CFL associated with the CFG given below:-

$S \to aB \mid bA$
$A \to a \mid aS \mid bAA$
$B \to b \mid bS \mid aBB$

Sol^n:

$S \Rightarrow aB$
$\Rightarrow ab$

$\Rightarrow aB$
$\Rightarrow abS$
$\Rightarrow abbA$
$\Rightarrow abbaS$
$\Rightarrow abbaaB$
$\Rightarrow abbaab$

$S \Rightarrow bA$
$\Rightarrow bbAA$
$\Rightarrow bbaSA$
$\Rightarrow bbaaBA$
$\Rightarrow bbaabSA$
$\Rightarrow bbaabbAA$
$\Rightarrow bbaabbaa$

All the strings have equal no. of a's & b's.

$L = \{ x : x$ containing equal no. of a's & b's. $\}$

PARSE TREE or DERIVATION TREE

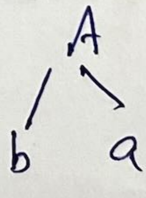
The strings generated by a context free grammar (V_N, Σ, P, S) can be represented by a hierarchical structure called tree. Such trees representing derivations are called derivation trees or parse tree or syntax tree.

Characteristics of a Parse Tree :-

A Parse Tree for a context free grammar G has the following characteristics:-

1) Every vertex of a parse tree has a label which is a variable or terminal or Λ.
2) The root of a parse tree has a label S.
3) The label of an internal vertex is a variable.
4) If a vertex A has K children with labels $A_1, A_2, \ldots A_k$, then $A \to A_1, A_2 \ldots A_k$ will be a production in CFG.
5) A vertex 'n' is a leaf if its label is $a \in \Sigma$ or Λ.
6) 'n' is the only son of its father if its label is Λ.

eg: $A \to ba$ Parse Tree

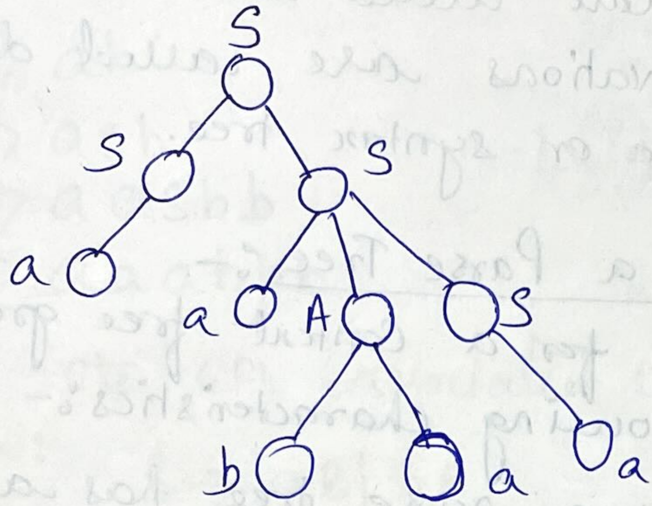


Eg:

$G = \{\{S, A\}, \{a, b\}, P, S\}$

$S \rightarrow aAS \mid a \mid SS$

$A \rightarrow SbA \mid ba$



$S \Rightarrow SS$
$\Rightarrow aS$
$\Rightarrow aaAS$
$\Rightarrow aabaS$
$\Rightarrow aabaa$

String generated by this derivation tree is

aabaa.

Yield of a derivation tree:- Yield of a derivation tree is the concatenation of the labels of leaf nodes in the left-to-right ordering without repetition.

Subtree of a derivation tree:- A subtree of a derivation tree is a tree having following characteristics:-

i) the root of subtree is some vertex v of tree.

Derivation:-

Leftmost Derivation:- A derivation $A \overset{*}{\Rightarrow} w$ is called leftmost derivation if we apply a production only to the leftmost variable at every step.

Rightmost Derivation:- A derivation $A \overset{*}{\Rightarrow} w$ is called rightmost derivation if we apply a production only to the rightmost variable at every step.

Eg:- $(S \to aSS \mid b)$ String:- aababbb

LMD	RMD
$S \Rightarrow aSS$	$S \Rightarrow aSS$
$\Rightarrow aaSSS$	$\Rightarrow aSb$
$\Rightarrow aabSS$	$\Rightarrow aaSSb$
$\Rightarrow aabaSSS$	$\Rightarrow aaSaSSb$
$\Rightarrow aababSS$	$\Rightarrow aaSaSbb$
$\Rightarrow aababbS$	$\Rightarrow aaSabbb$
$\Rightarrow aababbb$	$\Rightarrow aababbb$

Σ, V_N, P, S

Q:- A CFG given by the production is

$S \to a \mid aAS$

$A \to bS$

Obtain the derivation tree of the word abaabaa.

$\Rightarrow S \Rightarrow aAS$

$\Rightarrow abSS$

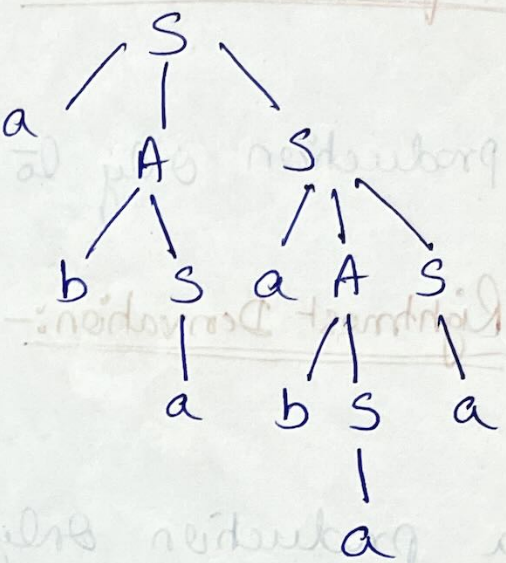
$\Rightarrow abaaS$

$\Rightarrow abaaAS$

$\Rightarrow abaabSS$

$\Rightarrow abaabaS$

$\Rightarrow abaabaa$

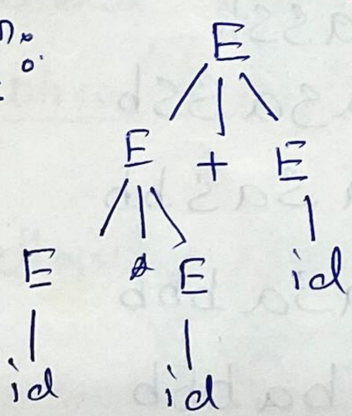


Q:- Given the grammar $G = (V_N, \Sigma, P, S)$; where $V_N = \{E\}$, $S = E$, $\Sigma = \{id, +, *, (\}$ and P Consists of $E \to E+E \mid E*E \mid (E) \mid id$.

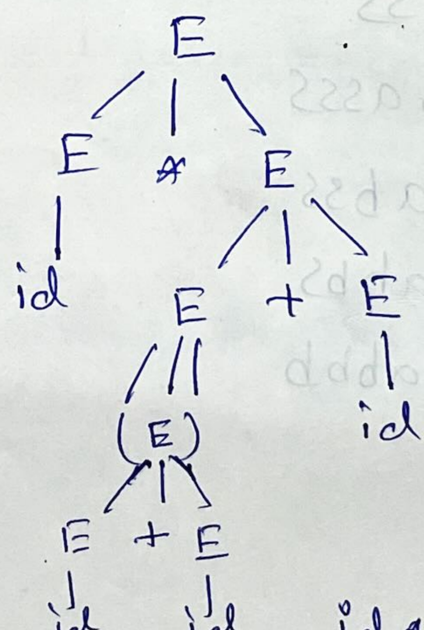
Give derivation tree.

id * (id *id) +id

Sol^n:-



(id * id) + id



id * (id+id) + id

$S \rightarrow aAS \mid a$

$A \rightarrow SbA \mid SS \mid ba$

Derive aabbaa using LMD and RMD.

Sol^n:-

$S \Rightarrow aAS$ using $S \rightarrow aAS$

$\Rightarrow aSbAS$ using $A \rightarrow SbA$

$\Rightarrow aabAS$ using $S \rightarrow a$

$\Rightarrow aabbaS$ using $A \rightarrow ba$

$\Rightarrow aabbaa$ using $S \rightarrow a$

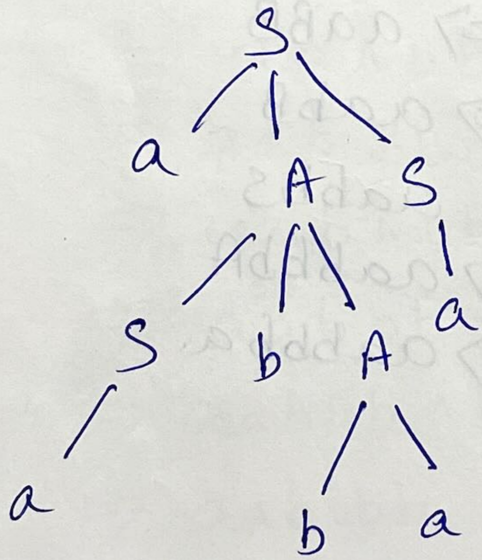
$S \Rightarrow aAS$ using $S \rightarrow aAS$

$\Rightarrow aAa$ using $S \rightarrow a$

$\Rightarrow aSbAa$ using $A \rightarrow SbA$

$\Rightarrow aSbbaa$ using $A \rightarrow ba$

$\Rightarrow aabbaa$ using $S \rightarrow a$



Ambigous grammar.

Q: $S \to aB \mid bA$

$A \to a \mid aS \mid bAA$

$B \to b \mid bS \mid aBB$

Derive using LMD and RMD.

ⓐ aaabbb ⓑ bbaaba ⓒ aabbba ⓓ aaabbabbba

ⓐ
$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aaaBBB$
$\Rightarrow aaabBB$
$\Rightarrow aaabbB$
$\Rightarrow aaabbb$

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aaBb$
$\Rightarrow aaaBBb$
$\Rightarrow aaaBbb$
$\Rightarrow aaabbb$

ⓑ bbaaba

$S \Rightarrow bA$
$\Rightarrow bbAA$
~~$\Rightarrow bbaA$~~
$\Rightarrow bbaSA$
$\Rightarrow bbaaBA$
$\Rightarrow bbaabA$
$\Rightarrow bbaaba$

ⓒ aabbba

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aabB$
$\Rightarrow aabbS$
$\Rightarrow aabbbA$
$\Rightarrow aabbba$

Q:- Show that the derivation steps and construct derivation tree for string aabbbb by using left most derivation with the grammar defined as:-

$$S \rightarrow AB \mid \Lambda$$
$$A \rightarrow aB$$
$$B \rightarrow Sb$$

Soln: aabbbb

$S \Rightarrow AB$

$\Rightarrow aBB$

$\Rightarrow aSbB$

$\Rightarrow \cancel{aaBbB}\ aABbB$

$\Rightarrow \cancel{aaS}\ aaBBbB$

$\Rightarrow aaSbBbB$

$\Rightarrow aa\Lambda bBbB$

$\Rightarrow aabSbbB$

$\Rightarrow aab\Lambda bbB$

$\Rightarrow aabbbSb$

$\Rightarrow aabbb\Lambda b$

$\Rightarrow aabbbb$

Q:- Let G be the grammar

$S \to 0B \mid 1A$

$A \to 0 \mid 0S \mid 1AA$

$B \to 1 \mid 1S \mid 0B$

For the string 001101010, find

(a) leftmost derivation (b) rightmost derivation (c) derivation tree.

(a) Leftmost Derivation

~~001101010~~

$S \Rightarrow 0B$

$\Rightarrow 00B$

$\Rightarrow 001S$

$\Rightarrow 0011A$

$\Rightarrow 00110S$

$\Rightarrow 001101A$

$\Rightarrow 0011010S$

$\Rightarrow 00110101A$

$\Rightarrow 001101010$

(b) Rightmost Derivation

001101010

$S \Rightarrow 0B$

$\Rightarrow 00B$

$\Rightarrow 001S$

$\Rightarrow 0011A$

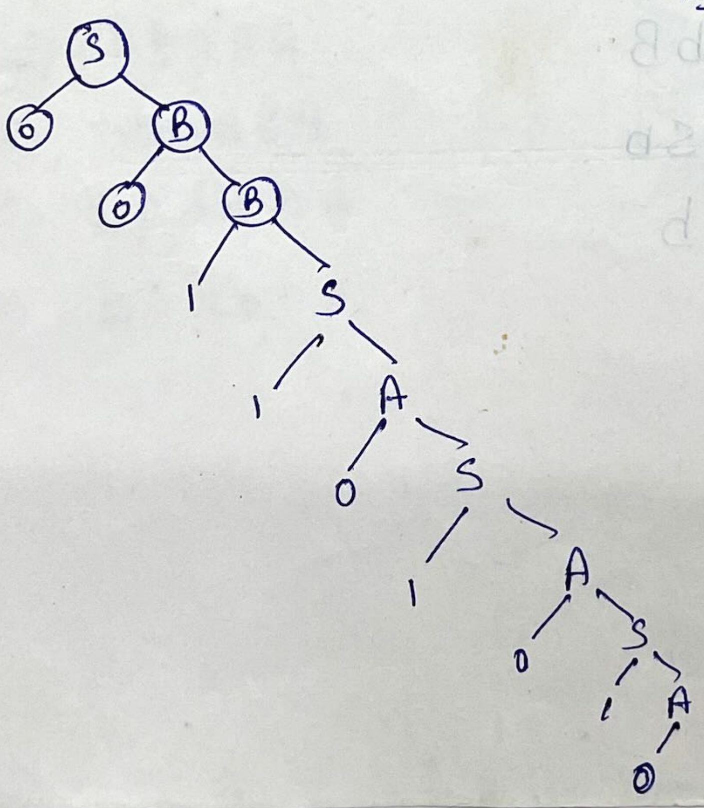
$\Rightarrow 00110S$

$\Rightarrow 001101A$

$\Rightarrow 0011010S$

$\Rightarrow 00110101A$

$\Rightarrow 001101010$



Q:- $S \to aB \mid bA$
$A \to a \mid aS \mid bAA$
$B \to b \mid bS \mid aBB$

Derive the following string using LMD & RMD.
ⓐ aaabbb
ⓑ bbaaba
ⓒ aaba

ⓐ aaabbb

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aaaBBB$
$\Rightarrow aaabBB$
$\Rightarrow aaabbB$
$\Rightarrow aaabbb$

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aaaBBB$
$\Rightarrow aaaBBb$
$\Rightarrow aaaBbb$
$\Rightarrow aaabbb$

ⓑ bbaaba

LMD

$S \Rightarrow bA$
$\Rightarrow bbAA$
$\Rightarrow bbaSA$
$\Rightarrow bbaaBA$
$\Rightarrow bbaabA$
$\Rightarrow bbaaba$

RMD

$S \Rightarrow bA$
$\Rightarrow bbAA$
$\Rightarrow bbAa$
$\Rightarrow bbaSa$
$\Rightarrow bbaaBa$
$\Rightarrow bbaaba$

ⓒ aaba — Not derivable

LMD

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aabB$
$\Rightarrow$

RMD

Simplification of CFG

Q. Remove null prod.

(1) $S \to ABaC$
$A \to BC$
$B \to b \mid \lambda$
$C \to D \mid \lambda$
$D \to d$

$S \to ABaC \mid BaC \mid AaC \mid ABa \mid Aa \mid Ba \mid aC \mid a$
$A \to BC \mid B \mid C$
$B \to b$
$C \to D$
$D \to d$

(2) Remove Unit prod.
$S \to A \mid Bb$
$A \to B \mid Bc$
$B \to a \mid aa$

$B \to a \mid aa$
$A \to a \mid aa \mid Bc$
$S \to a \mid aa \mid Bc \mid Bb$

(3) Eliminate useless variable

$S \to ABaC \mid Db$
$A \to AC \mid ab$
$B \to bc$
$C \to c$
$E \to e$

$S \to ABaC$
$A \to AC \mid ab$
$B \to bc$
$C \to c$

Q: Find the reduced grammar equivalent ... grammar G, whose production are -

$S \rightarrow AB \mid CA$

$B \rightarrow BC \mid AB$

$A \rightarrow a$

$C \rightarrow aB \mid b$

Soln: Step 1: To find the equivalent grammar G_1 such that each variable in G_1 derives some terminal string.

(a) Construction V_{N_1}:

$W_1 = \{A, C\}$ as $A \rightarrow a$ & $C \rightarrow b$

$W_2 = \{A, C\} \cup \{A_1 : A_1 \rightarrow \alpha \text{ with } \alpha \in (\Sigma \cup \{A, C\})^*\}$

$= \{A, C\} \cup \{S\}$

$= \{S, A, C\}$

$W_3 = \{S, A, C\} \cup \{A_1 : A_1 \rightarrow \alpha \text{ with } \alpha \in (\Sigma \cup \{S, A, C\})^*\}$

$= \{S, A, C\} \cup \phi$

$= \{S, A, C\}$

$W_{N_1} = W_3 = W_2 = \{S, A, C\}$

$P_1 : \boxed{S \rightarrow CA \ ; \ A \rightarrow a \ ; \ C \rightarrow b}$

Thus, $G_1 = (\{S, A, C\}, \{a, b\}, \{S \rightarrow CA, A \rightarrow a, C \rightarrow b\}, \{S\})$

Step 2: We construct a grammar $G' = (V_N', \Sigma, P', S)$ equivalent to G_1 so that every symbol in G' appears in some sentential form of G'.

$W_1 = \{S\}$

$W_2 = \{S\} \cup \{X \in V_{N1} \cup \Sigma$: there exists a production $A \to \alpha$ with $A \in W_1$ and α containing $X\}$

$W_2 = \{S\} \cup \{C, A\}$
$= \{S, C, A\}$

$W_3 = \{S, C, A\} \cup \{X \in V_{N1} \cup \Sigma$: there exists a production $A \to \alpha$ with $A \in W_2$ and α containing x

$= \{S, C, A\} \cup \{a, b\}$
$= \{S, C, A, a, b\}$

$W_4 = \{S, C, A, a, b\} \cup \{X \in V_{N1} \cup \Sigma$: there exists a production $A \to \alpha$ with $A \in W_3$ and α containing

$W_4 = \{S, C, A, a, b\}$

$W_4 = W_3$

$P' = \{A \to \alpha : A \in W_4\}$
$= S \to CA$
$A \to a$
$C \to b$

$\therefore G' = (\{S, A, C\}, \{a, b\}, \{S \to CA, A \to a, C \to b\}, \{S\})$

is the reduced grammar.

Q:- Find the reduced grammar that is equi. to the CFG given below:-

$S \to aC \mid SB$

$A \to bSCa$

$B \to aSB \mid bBC$

$C \to aBC \mid ad$

Solⁿ: Step 1: Find the equivalent grammar G_1 such that each variable in G_1 derives some terminal string.

(a) Construction of V_{N1}:

$W_1 = \{C\}$ $\quad C \to ad$

$W_2 = \{C\} \cup \{A_1 : A_1 \to \alpha \text{ with } \alpha \in (\Sigma \cup \{W_1\}^*)\}$

$= \{C\} \cup \{S\}$

$= \{C, S\}$

$W_3 = \{C, S\} \cup \{A_1 : A_1 \to \alpha \text{ with } \alpha \in (\Sigma \cup \{W_2\}^*)\}$

$= \{C, S\} \cup \{A\}$

$= \{C, S, A\}$

$W_4 = \{C, S, A\} \cup \{A_1 : A_1 \to \alpha \text{ with } \alpha \in (\Sigma \cup \{W_3\}^*)\}$

$= \{C, S, A\} \cup \{\emptyset\}$

$= \{C, S, A\}$

$W_{N1} = W_3 = W_4 = \{S, C, A\}$

P_1:- $S \to aC$

$A \to bSCa$

$C \to ad$

#2: every

Step 2: Now, we construct a grammar $G' = (V_N', \Sigma, P', S)$ equivalent to G_1, so that every symbol in G' appears in some sentential form of G'.

$W_1 = \{S\}$

$W_2 = \{S\} \cup \{X \in V_N \cup \Sigma$: where exists a prod. $A \to \alpha$ with $A \in W_1$ & α containing

$= \{S\} \cup \{a, C\}$

$= \{S, a, C\}$

$W_3 = \{S, a, C\} \cup \{\cancel{a}, \cancel{S}, \cancel{C}, a\}$

$= \{S, a, C, d\}$

$W_4 = \{S, a, C, d\} \cup \{a, d\}$

$= \{S, a, C, d\}$

$W_3 = W_4 = \{S, C, a, d\}$

$\therefore P' : \{S \to aC$

$C \to ad\}$

$G' = (\{S, C\}, \{a, d\}, \{S \to aC, C \to ad\}, S\}$

Q: Consider CFG,

$S \to AB \mid a$

$A \to b$ Identify & eliminate use[less]

Soln: Step 1: Find the equivalent grammar G_1 such that each variable in G_1 derives terminal symbol

$W_1 = \{S, A\}$

$W_2 = \{S, A\} \cup \{A_1 : A_1 \to \alpha$ with $\alpha \in (\Sigma \cup \{W_1\}^*)\}$

$= \{S, A\} \cup \{\phi\} = \{S, A\}$

$W_3 = W_2 = V_{N1} = \{S, A\}$

$P_1 : \{S \to a, A \to b\}$

Step 2: Find the grammar G' equivalent to G_1 such that every symbol in G' ~~should~~ appears in some sential form of G'.

$W_1 = \{S\}$

$W_2 = \{S\} \cup \{X : X \in V_{N1} \cup \Sigma$: where exists production $A \to \alpha$ with $A \in W_1$ & α containing $X\}$

$W_2 = \{S\} \cup \{\phi\}$

$= \{S\}$

$W_1 = W_2$ $V_N' = \{S\}$

$\therefore$ $P' = \{S \to a\}$

$\therefore$ $G' = (\{S\}, \{a\}, \{S \to a\}, S)$

...limination of Null Production

A context free grammar may have productions of the form $A \to \Lambda$. A production of the form $A \to \Lambda$, where A is variable, is called a null production.

Q: Consider the grammar G whose productions are $S \to aS \mid AB$, $A \to \Lambda$, $B \to \Lambda$, $D \to b$. Construct a grammar G_1 without null productions.

Soln: **Step 1:-** Construction of set W of all nullable variable

Step 2: Construction of P'.

* **Step 1:** $W_1 = \{ A_1 \in V_N : A_1 \to \Lambda \text{ is a production in } G \}$

$= \{A, B\}$

$W_2 = W_1 \cup \{ A_2 \in V_N : \text{there exists a production } A_2 \to \alpha \text{ with } \alpha \in W_1 \}$

$= \{A, B\} \cup \{S\} = \{S, A, B\}$

$W_3 = W_2 \cup \{A_3\}$

$= W_2 \cup \{\phi\} = \{S, A, B\}$

$W_2 = W_3 = \{S, A, B\}$

* **Step 2:-** Construction of P'

(a) Any production whose RHS is not having nullable variable on RHS is included in P'.

(b) The production are obtained either by not erasing any nullable variable on the RHS of $A \to X_1 X_2 \dots X_K$ or by erasing some or all nullable variables provided some symbol appears on the RHS after erasing.

(c) $S \rightarrow$

(a) $D \rightarrow b$

(b) $S \rightarrow aS$ gives $S \rightarrow aS$ & $S \rightarrow a$

$S \rightarrow AB$ gives $S \rightarrow AB$, $S \rightarrow A$ & $S \rightarrow B$.

$\therefore$ The required grammar without null production is

$$G_1 = (\{S, A, B, D\}, \{a, b\}, P', S)$$

P':- $S \rightarrow aS$

$S \rightarrow a$

$D \rightarrow b$

$S \rightarrow AB$

$S \rightarrow A$

$S \rightarrow B$

Q:- Remove the null production from the following CFG by preserving meaning of it -

$S \rightarrow XYX$

$X \rightarrow 0X \mid \wedge$

$Y \rightarrow 1Y \mid \wedge$.

Sol^n: Step 1: $W_1 = \{X, Y\}$

$W_2 = W_1 \cup \{A_1\}$

$= W_1 \cup \{S\}$

$= \{S, X, Y\}$

$W_3 = W_2 \cup \{A_2\} = W_2 \cup \{\phi\}$

$= \{X, Y, S\}$

Step 2: Construction of P'

$S \to XYX$ gives $S \to XY$, $S \to YX$, $S \to XX$, $S \to XYX$
$S \to X$, $S \to Y$.

$X \to 0X$ gives $X \to 0$, $X \to 0X$,

$Y \to 1Y$ gives $Y \to 1Y$, $Y \to 1$,

Now, collectively we write CFG without null production for given CFG is -

P' :- $S \to XYX \mid XY \mid YX \mid XX \mid X \mid Y$
$X \to 0 \mid 0X$
$Y \to 1 \mid 1Y$

Q: Construct CFG without null production from the one which is given below:-

$S \to a \mid Ab \mid aBa$
$A \to b \mid \Lambda$
$B \to b \mid A$.

Sol^n: Step 1: Construction of set W of nullable variables.

$W_1 = \{A\}$

$W_2 = \{A\} \cup \{A_1\}$
$= \{A\} \cup \{S, B\}$
$= \{S, A, B\}$

$W_3 = W_2 \cup \{A_2\}$

$W_3 = W_2 = \{S, A, B\}$

Step 2: Construction of P'

(a) $A \to b$

(a) $B \to b$ & $S \to a$

(b) $S \to Ab$ gives $S \to A$, $S \to b$, $S \to Ab$

$S \to aBa$ gives $S \to a$, $S \to aa$, $S \to B$, $S \to aB$, $S \to Ba$

$B \to A$

Now collectively, we can write CFG without the null production is –

$$a \mid Ab \mid b \mid aBa \mid aa$$

$$S \to \cancel{a \mid A \mid b \mid Ab \mid aa \mid B \mid aB \mid Ba}$$

$A \to b$

$B \to b \mid A$

Q: Design a CFG for the regular expression $r = (a+b)^* bb(a+b)^*$, which is free from null productions.

Soln:

$S \to BAB$

$A \to bb$

$B \to aB \mid bB \mid \wedge$

$S \to BAB \mid A$

$A \to bb$

$B \to a \mid aB \mid b \mid bB$

Q: (i) $S \to AB \mid \Lambda$

$A \to aASb \mid a$ Remove null

$B \to bS$

(ii) $S \to ABA$

$A \to a \mid \Lambda$

$B \to b \mid \Lambda$

Soln: (i) Nullable Variable - $\{S\}$

$S \to AB$

$A \to a$

$A \to aASb$ $\quad A \to aAb \mid aASb$

$B \to bS$ $\quad B \to b \mid bS$

New Production:-

$S \to AB$

$A \to a \mid aAb \mid aASb$

$B \to b \mid bS$

(ii) Nullable Variable - $\{A, B\}$

$S \to ABA$ $\quad S \to AB \mid BA \mid AA \mid B \mid A \mid ABA$

$A \to a$

$B \to b$

Q: $S \to aSb \mid aAbB \mid a \mid \epsilon$

$A \to aA \mid aBbA \mid \epsilon$

$B \to bB \mid \epsilon$

Soln: Nullable Variable = $\{S, A, B\}$

$S \to aSb$	$S \to ab \mid aSb$
$S \to aAbB$	$S \to aAb \mid abB \mid ab \mid aAbB$
$S \to a$	$S \to a$
$A \to aA$	$A \to a \mid aA$
$A \to aBbA$	$A \to aBb \mid abA \mid ab \mid aBbA$
$B \to bB$	$B \to bB \mid b$

$\therefore$ $S \to ab \mid aAb \mid abB \mid aAbB \mid a \mid aSb$

$A \to a \mid aA \mid aBb \mid abA \mid ab \mid aBbA$

$B \to bB \mid b$

Q:

$S \to XY$	$S \to XY$
$X \to aW$	$X \to aW \mid a$
$Y \to bZ$	$Y \to bZ \mid b$
$W \to Z$	$W \to Z$
$Z \to AB$	$Z \to AB \mid A \mid B$
$A \to aA \mid \epsilon$	$A \to aA \mid a$
$B \to bB \mid \epsilon$	$B \to bB \mid b$

Elimination of Unit Production

A unit production in a Context free grammar G is a production of the form $A \to B$, where A & B are variables in G.

Q: Let G be $S \to AB$, $A \to a$, $B \to C|b$, $C \to D$, $D \to E$ & $E \to a$. Eliminate unit production and get an equivalent grammar.

Soln:-

Step 1: Construction of ~~suitable~~ the set of variables derivable from Variable.

Step 2:- Constructions of A- productions in G_1

Step 1: Construction of the set of variables derivable from Variable

$W_0(S) = \{S\}$

$W_1(S) = W_0(S) \cup \{B \in V_N : C \to B \text{ is in } P \text{ with } C \in W_0(S)\}$

$= \{S\} \cup \phi = \{S\}$

$\therefore W(S) = \{S\}$

$W_0(A) = \{A\}$

$W_1(A) = W_0(A) \cup \{\phi\} = \{A\}$

$W(A) = \{A\}$

$W_0(B) = \{B\}$

$W_1(B) = W_0(B) \cup \{C\} = \{B, C\}$

$W_2(B) = W_1(B) \cup \{D\} = \{B, C, D\}$

$W_3(B) = W_2(B) \cup \{E\} = \{B, C, D, E\}$

$W_4(B) = W_3(B)$

$\underline{W(B)} = \{B, C, D, E\}$

$\underline{W_0(C)} = \{C, D, E\}$

$\underline{W(D)} = \{D, E\}$

$\underline{W(E)} = \{E\}$

Step 2: Construction of A-production in G_1:

(a) non-unit production in G

(b) $A \to \alpha$ whenever $B \to \alpha$ is in G with $B \in W(A)$

(a) $A \to a$, $B \to b$, $E \to a$, $S \to AB$

(b)

$S \to AB$

$A \to a$

$B \to a|b$

$C \to a$

$D \to a$

$E \to a$

Q: $S \to AB \mid A$
$A \to C \mid d$
$C \to d$

Solⁿ: $S \to AB \mid d$
$A \to d$

Q: $S \to Aa \mid B$
$B \to A \mid bb$
$A \to a \mid bc \mid B$

$S \to B$
$B \to A$
$A \to B$

$S \to Aa \mid a \mid bc \mid bb$
$B \to a \mid bc \mid bb$
$A \to a \mid bc \mid bb$

Q: $S \to aSb \mid a \mid A$
$A \to aA \mid Ab \mid a$

$S \to A$

$S \to aSb \mid a$
$A \to aA \mid Ab \mid a$

Solⁿ: Given $G = (\{S, A\}, \{a, b\}, P, S)$

P: $S \to aSb \mid a \mid A$
$A \to aA \mid Ab \mid a$

Let G' be the grammar with no useless variable.

$G' = (V', T', P', S)$

Production

$A \to a$
$A \to Ab$
$A \to aA$
$S \to aSb$
$S \to a$
$S \to A$

$A \to aA \mid Ab \mid a$
$S \to aSb \mid a \mid aA \mid Ab \mid a$

$S \to a$; $S \to Ab$; $S \to aA$

New Production Rules

$S \to a \mid Ab \mid aA \mid aSb$
$A \to aA \mid Ab \mid a$

Q: $S \to A | bb$ Reduce the following grammar. ✓
$A \to B | a$
$B \to S | b$

Sol^n: Step 1:- Eliminate Useless Variable.
No useless variable.

Step 2:-

Production	New Production
$S \to A$	$S \to B$, * $S \to b$; $S \to a$
* $S \to bb$	
$A \to B$	* $A \to b$, $A \to S$, * $A \to bb$
* $A \to a$	
$B \to S$	* $B \to bb$ $B \to A$ * $B \to a$
* $B \to b$	

New Production

$S \to a | b | bb$
$A \to a | b | bb$
$B \to a | b | bb$

$B \to bb | b | a$
$A \to b | a | bb$
$S \to b | a | bb$

After eliminating useless variable,

$S \to a | b | bb$

Q: $S \to AB \mid CA$

$A \to a$

$B \to BC \mid AB$

$C \to aB \mid b$

Step 1: Elimination of useless variable-

$S \to CA$

$A \to a$

$C \to b$

Q:- $S \to AB$

$A \to a$

$B \to C \mid b$

$C \to D$

$D \to E$

$E \to a$

$S \to AB$

$A \to a$

$B \to b$

$B \to C \to D \to E \to a \Rightarrow B \to a$

$C \to D \to E \to a \Rightarrow C \to a$

$D \to E \to a \Rightarrow D \to a$

$E \to a$

Eliminating useless variable-

$S \to AB$

$A \to a$

$B \to b \mid a$

Q:-

Q: Let G be $S \to AB$
$A \to a$
$B \to C|b$
$C \to D$
$D \to E$, $E \to a$

Last Page

Q: $S \to Aa|B$
$B \to A|bb$
$A \to a|bc|B$

$S \to Aa$ ✓
$S \to B$; $S \to bb$ ✓
$S \to B \to A \to a$
bc
$S \to a$ ✓
$S \to bc$ ✓
$B \to A \to bb \Rightarrow B \to bc$ ✓
$B \to A \to a \Rightarrow B \to a$ ✓
$A \to a$ ✓
$A \to bc$ ✓
$A \to B \to bb \Rightarrow A \to bb$ ✓

New Production

$S \to Aa|bb|a|bc$
$A \to a|bc|bb$
$B \to a|bc|bb$

After removing useless

$S \to Aa|bb|a|bc$
$A \to a|bc|bb$

Q: $S \to AB|A$ $A \to C|d$ $C \to d$

$S \to AB$
$S \to A \to C \to d$ $S \to d$
$A \to C \to d$

New Productions

$S \to AB|d$
$A \to d$
$C \to d$

Removing Useless

$S \to AB|d$
$A \to d$

(a) $S \rightarrow AB1 \mid 0$
$A \rightarrow 00A \mid B$
$B \rightarrow 1A1$

(b) $A \rightarrow BC$
$B \rightarrow CA \mid b$
$C \rightarrow AB \mid a$

Soln: (b) $A \rightarrow BC$
$B \rightarrow CA \mid b$
$C \rightarrow AB \mid a$

Step 1: No Unit, No Null, No Useless.

Step 2: Given grammar is in CNF.

Step 3: ~~Reduction~~ Equivalent GNF.

Substitute $A = A_1$ $B = A_2$ $C = A_3$

$\therefore A_1 \rightarrow A_2 A_3$ — ok

$A_2 \rightarrow A_3 A_1 \mid b$ — ok

$A_3 \rightarrow A_1 A_2 \mid a$ $V_i \rightarrow V_j$ $i<j$ No $i<j$ not Satisfie

$\therefore A_3 \rightarrow A_1 A_2 \mid a$

$A_3 \rightarrow A_2 A_3 A_2 \mid a$

$A_3 \rightarrow A_3 \underbrace{A_1 A_3 A_2}_{\alpha} \mid \underbrace{b A_3 A_2}_{\beta_1} \mid \underbrace{a}_{\beta_2}$

$A_3 \rightarrow b A_3 A_2 \mid a \mid b A_3 A_2 Z \mid aZ$ — ok

$Z \rightarrow A_1 A_3 A_2 \mid A_1 A_3 A_2 Z$ — Needs to be Modified

$Z \rightarrow A_2 A_3 A_3 A_2 \mid A_2 A_3 A_3 A_2 Z$

$Z \rightarrow A_3 A_1 A_3 A_3 A_2 \mid A_3 A_1 A_3 A_3 A_2 Z \mid b A_3 A_3 A_2 \mid b A_3 A_3 A_2 Z$

$Z \rightarrow b A_3 A_2 A_1 A_3 A_3 A_2 \mid b A_3 A_2 A_1 A_3 A_3 A_2 Z \mid b A_3 A_3 A_2 \mid b A_3 A_3 A_2 Z$

$A_3 \rightarrow b A_3 A_2 \mid a \mid b A_3 A_2 Z \mid aZ$

$A_2 \rightarrow b A_3 A_2 A_1 \mid a A_1 \mid b A_3 A_2 Z A_1 \mid a Z A_1 \mid b$

$A_1 \rightarrow b A_3 A_2 A_1 A_3 \mid a A_1 A_3 \mid b A_3 A_2 Z A_1 A_3 \mid a Z A_1 A_3 \mid b A_3$

Q: Let G be the grammar

$S \to aB \mid bA$

$A \to a \mid aS \mid bAA$

$B \to b \mid bS \mid aBB.$

for the string aaabbabbba find the leftmost derivation.

Solⁿ:

$S \Rightarrow aB$

$\Rightarrow aaBB$ $(B \to aBB)$

$\Rightarrow aaaBBB$ $(B \to aBB)$

$\Rightarrow aaabBB$ $(B \to b)$

$\Rightarrow aaabbB$ $(B \to b)$

$\Rightarrow aaabbaBB$ $(B \to aBB)$

$\Rightarrow aaabbabB$ $(B \to b)$

~~$\Rightarrow aaabbabbB$~~ $\Rightarrow aaabbabbS$ $(B \to bS)$

~~$\Rightarrow aaabbabb$~~ $\Rightarrow aaabbabbbA$ $(S \to bA)$

$\Rightarrow aaabbabbba$ $(A \to a)$.

Q2: Let G be the grammar

~~$S \to aB|bA$~~ $S \to aCa$

~~$A \to a|aS|bAA$~~ $C \to aCa \mid b$

find L(G).

Solⁿ:-

$S \Rightarrow aCa \Rightarrow aba$

$S \Rightarrow aCa \Rightarrow aaCaa \Rightarrow aaaCaaa \Rightarrow aaabaaa$

It consists of the languages of the strings palindrome over equal no. of a and middle symbol is b.

Q:- Show that id + id * id can be generated by two distinct leftmost derivation in the grammar.

$$E \to E+E \mid E*E \mid (E) \mid id$$

Solⁿ:-

$$E \Rightarrow E+E$$
$$\Rightarrow id+E$$
$$\Rightarrow id+E*E$$
$$\Rightarrow id+id*E$$
$$\Rightarrow id+id*id$$

$$E \Rightarrow E*E$$
$$\Rightarrow E+E*E$$
$$\Rightarrow id+E*E$$
$$\Rightarrow id+id*E$$
$$\Rightarrow id+id*id$$

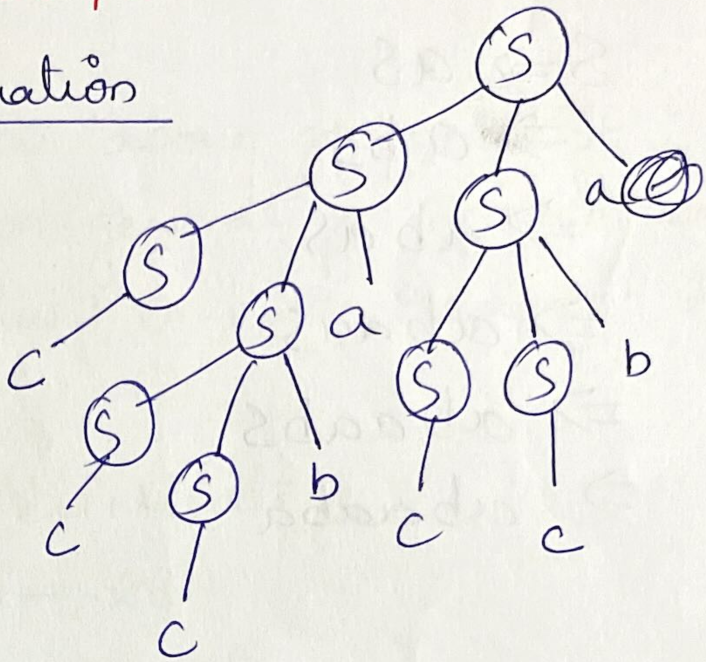
Ambiguous grammar.

Q:- For the grammar $S \to A1B$

Q:- Obtain leftmost derivation, rightmost derivation and derivation tree for the string "cccbaccba". The grammar is:-

$S \to SSa \mid SSb \mid c$

Sol^n:- Leftmost derivation



$S \Rightarrow SSa$
$\Rightarrow SSaSa$
$\Rightarrow cSaSa$
$\Rightarrow cSSbaSa$
$\Rightarrow ccSbaSa$
$\Rightarrow cccbaSa$
$\Rightarrow cccbaSSba$
$\Rightarrow cccbacSba$
$\Rightarrow cccbaccba$

Righmost derivation

$S \Rightarrow SSa$
$\Rightarrow SSSba$
$\Rightarrow SScba$
$\Rightarrow Sccba$
$\Rightarrow SSaccba$
$\Rightarrow SSSbaccba$
$\Rightarrow SScbaccba$
$\Rightarrow Sccbaccba$
$\Rightarrow cccba\ ccba$

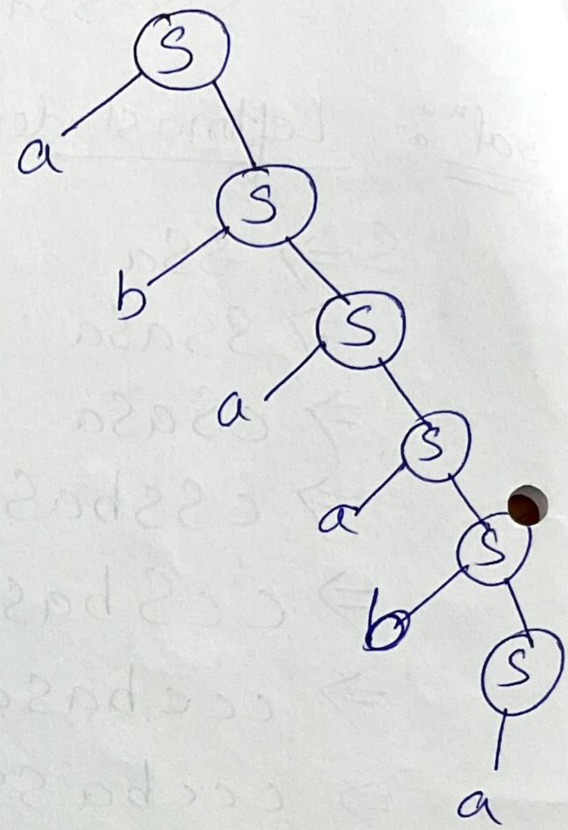
Q: A CFG is given by the production

$S \to aS \mid bS \mid a$

Obtain the derivation tree of the word "abaaba".

Sol^n:

$S \Rightarrow aS$
$\Rightarrow abS$
$\Rightarrow abaS$
$\Rightarrow abaaS$
$\Rightarrow abaabS$
$\Rightarrow abaaba$



Q:- Consider a CFG

$S \to AB|0$, $A \to 1$

Identify and eliminate useless symbols.

Sol^n:-

$S \to AB|0$

$A \to 1$

By observing the above grammar it is clear that ~~the~~ B is non-generating symbol. Since A derives 1, S derives 0 but B doesn't derive anystring.

So, we can eliminate

$S \to AB$

Now, CFG becomes,

$S \to 0$

$A \to 1$

Here, A is non reachable symbol, since it cannot be reached by ~~non-generating~~ start symbol S.

$\therefore$ New Reduced CFG is

$S \to 0|1$

Q:- Remove the useless symbol from the given CFG,

$S \rightarrow 0B \mid 1X$

$A \rightarrow BA0 \mid 1SX \mid 0$

$B \rightarrow 0SB \mid 1BX$

$X \rightarrow SBD \mid 0B0 \mid 0$

Soln: From the above grammar we can see that A is non reachable symbol from start symbol and B is non-generating symbol.

∴ Reduced grammar will be,

$S \rightarrow 1X$

$X \rightarrow 0$

Q:- $S \to ASB \mid \epsilon$

$A \to aAS \mid a$

$B \to SbS \mid A \mid bb$

Soln:- Removal of ϵ productions :-

$S \to ASB \mid AB$

$A \to aAS \mid a \mid aA$

$B \to SbS \mid A \mid bb \mid Sb \mid bS \mid b$

Removal of Unit Production :-

$S \to ASB \mid AB$

$A \to aAS \mid a \mid aA$

$B \to SbS \mid bb \mid Sb \mid bS \mid b \mid aAS \mid a \mid aA$

Removal of Useless Symbol :-

$S \to ASB \mid AB$

$A \to aAS \mid a \mid aA$

$B \to SbS \mid bb \mid Sb \mid bS \mid aAS \mid aA \mid a \mid b$

Q: $S \to 0A0 \mid 1B1 \mid BB$
$A \to C$
$B \to S \mid A$
$C \to S \mid \epsilon$

Sol^n: Removal of ϵ production:-

$$\left\{\begin{array}{l|l} S \to 0A0 \mid 1B1 \mid BB & S \to 0A0 \mid 1B1 \mid BB \mid 00 \\ A \to C \mid \epsilon & A \to C \\ B \to S \mid A & B \to S \mid A \mid \epsilon \\ C \to S & C \to S \end{array}\right\}$$

$$\left\{\begin{array}{l} S \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB \mid B \mid \epsilon \\ A \to C \\ B \to S \mid A\cancel{\epsilon} \\ C \to S \end{array}\right.$$

$$\left\{\begin{array}{l} S \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB \mid B \\ A \to C \\ B \to S \mid A \\ C \to S \end{array}\right.$$

Removal of Unit Production

$S \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB\cancel{B}$
$A \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB$
$B \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB$
$C \to 0A0 \mid 00 \mid 1B1 \mid 11 \mid BB$

Removal of Useless Variable

$S \to 0A0 \mid 1B1 \mid 00 \mid 11 \mid BB$
$A \to 0A0 \mid 1B1 \mid 00 \mid 11 \mid BB$
$B \to 0A0 \mid 1B1 \mid 00 \mid 11 \mid BB$

Q: $S \to Ab$

$A \to a$

$B \to C \mid b$

$C \to D$

$D \to E$

$E \to a$

Solⁿ: Elimination of Unit Production:-

$S \to Ab$

$A \to a$

$B \to a \mid b$

$C \to a$

$D \to a$

$E \to a$

Elimination of Useless Variable:-

$S \to Ab$

$A \to a$

Not able to reach B, C, D, E

Q: $S \to aC \mid SB$

$A \to bSCa$

$B \to aSB \mid bBC$ Simplify the grammar.

$C \to aBC \mid ad$

Solⁿ: No Null production.

No Unit Production.

A is Non reachable State
B is non generating state } Useless Variable

Final Simplified Grammar.

$S \to aC$

$C \to ad$

Q1. CFG to CNF

$S \to ASB$

$A \to aAS \mid a \mid \epsilon$

$B \to SbS \mid A \mid bb$

Soln: Step 1: i) Removal of null production,

$A \to \epsilon$

$S \to ASB \mid SB$

$A \to aAS \mid a \mid aS$

$B \to SbS \mid bb \mid \epsilon \mid A$

$B \to \epsilon$

~~$S \to ASB \mid SB \mid AS \mid A$~~

~~$A \to aAS \mid a \mid aS$~~

~~$B \to SbS \mid bb$~~

$S \to ASB \mid SB \mid AS \mid S$

$A \to aAS \mid a \mid aS$

$B \to SbS \mid bb \mid A$

ii) Removal of Unit Production,

~~$S \to ASB \mid AB \mid AS \mid aAS \mid a \mid aS$~~

~~$A \to aAS \mid a \mid aS$~~

~~$B \to SbS \mid bb$~~

$S \to ASB \mid SB \mid AS$

$A \to aAS \mid a \mid aS$

$B \to SbS \mid bb \mid aAS \mid a \mid aS$

Step 2:- Conversion to CNF

$S \to AS \mid ASB \mid AB$

$A \to aAS \mid aS \mid a$

$B \to SbS \mid bb \mid aAS \mid aS \mid a$

$X \to a$, $Y \to b$

$S \to AS \mid ASB \mid AB$

$A \to XAS \mid XS \mid a$

$B \to SYS \mid YY \mid XAS \mid XS \mid a$

~~P → AS, Q → XA, R → SY~~

$P \to AS$ $Q \to XA$ $R \to SY$ ~~$S_2 \to XA$~~

$S \to AS \mid PB \mid AB$

$A \to QS \mid XS \mid a$

$B \to RS \mid YY \mid QS \mid XS \mid a$

$X \to a$

$Y \to b$

$P \to AS$

$Q \to XA$

$R \to SY$

Q:
$S \rightarrow ABA$
$A \rightarrow aA \mid bA \mid \epsilon$
$B \rightarrow bB \mid aA \mid \epsilon$

Soln:
$S \rightarrow ABA \mid AB \mid BA \mid AA \mid A \mid B$
$A \rightarrow aA \mid bA \mid a \mid b$
$B \rightarrow bB \mid aA \mid a \mid b$

$S \rightarrow$