# CS & IT ENGINEERING

## Theory of Computation

### Miscellaneous Topics (Part 1)

**Lecture No. 1**

By- DEVA Sir


01 Simplification of CFG

02 CNF CFG

03 GNF CFG

04 CYK Algorithm

05 Doubts

# ① Simplification of CFG :

$X \rightarrow \varepsilon$

NULL Rule

$X \rightarrow Y$

UNIT Rule

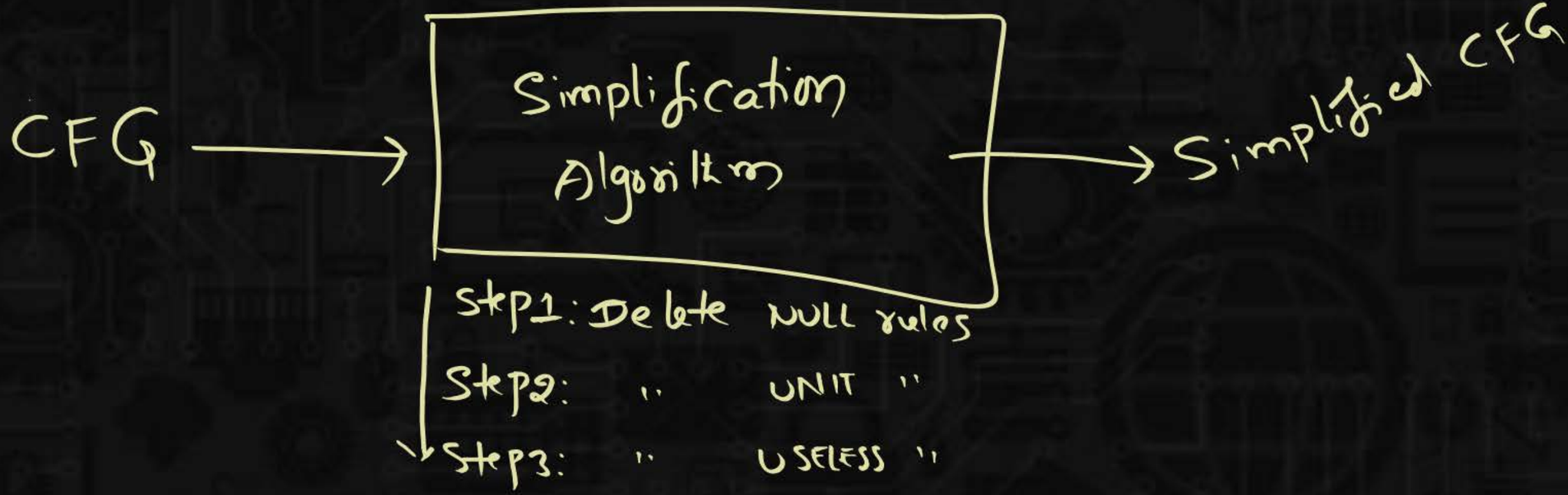
USELESS

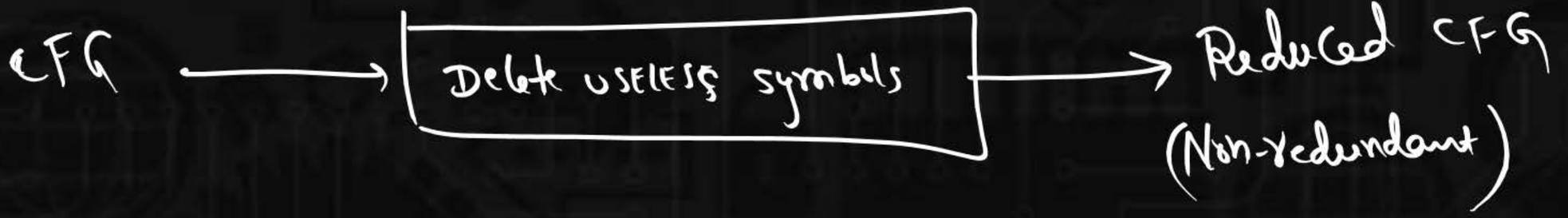
1) $X \rightarrow \alpha$

But X is not reachable from start

2) $X \rightarrow aX$

X can't derive any one string


CFG
Simplification Algorithm
Simplified CFG
Step1: Delete NULL rules
Step2: " UNIT "
Step3: " USELESS "


CFG
Delete USELESS symbols
Reduced CFG
(Non-redundant)

# I) Elimination of NULL Rules

$S \rightarrow aAb \mid bBa$

$A \rightarrow bS \mid aAa \mid bB \mid \varepsilon$

$B \rightarrow cA \mid bB \mid \varepsilon$

Step1: Delete $A \rightarrow \varepsilon$

$S \rightarrow aAb \mid bBa \mid ab$

$A \rightarrow bS \mid aAa \mid bB \mid aa$

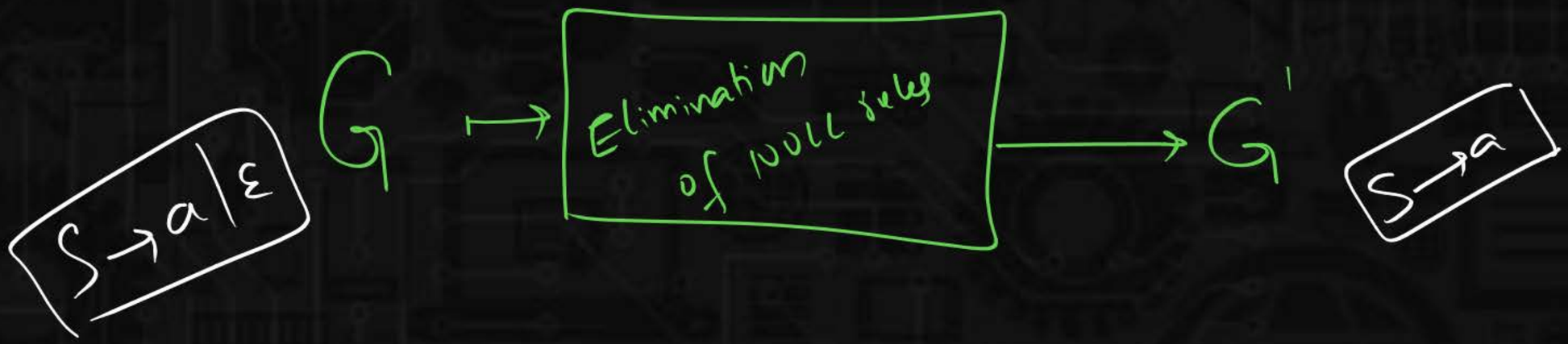
$B \rightarrow cA \mid bB \mid \varepsilon \mid c$

Step2:

$S \rightarrow aAb \mid bBa \mid ab \mid ba$

$A \rightarrow bS \mid aAa \mid bB \mid aa \mid b$

$B \rightarrow cA \mid bB \mid c \mid b$




$$L(G') = L(G) - \{\varepsilon\}$$

# II) Elimination of UNIT Rules

$S \rightarrow A \mid ab \mid B$

$A \rightarrow Sa \mid S \mid b$

$B \rightarrow a$

**Step1:** Delete $S \rightarrow A$

$S \rightarrow ab \mid B \mid Sa \mid \cancel{S} \mid b$

$A \rightarrow Sa \mid S \mid b$

$B \rightarrow a$

**Step2:** Delete $S \rightarrow B$

$S \rightarrow ab \mid Sa \mid b \mid a$

$A \rightarrow Sa \mid S \mid b$

$B \rightarrow a$

**Step3:** Delete $A \rightarrow S$

$S \rightarrow ab \mid Sa \mid b \mid a$

$A \rightarrow Sa \mid b \mid ab \mid a$

$B \rightarrow a$

# III) Elimination of USELESS Rules (USELESS symbols)

$S \to Aa \mid Bb \mid eD$

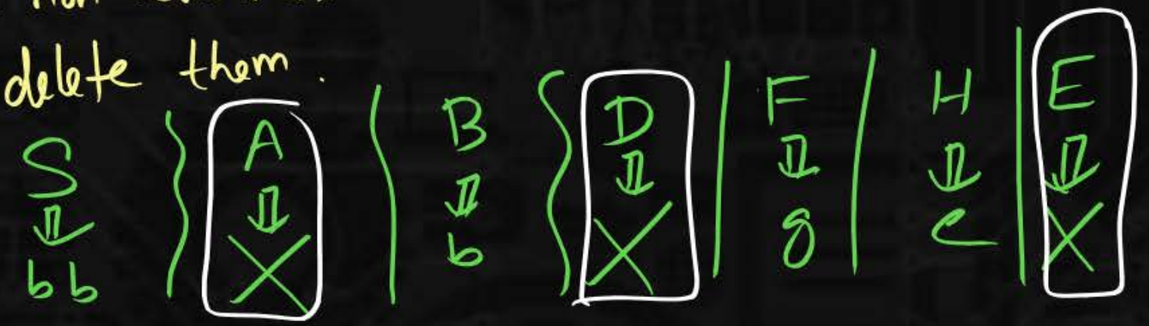
$A \to bA \mid dD$

$B \to b \mid Ea$

$F \to g$

$H \to e$

Step1: Find non-terminals which can't derive any and delete them.




E, A and D are USELESS

$S \to Bb$

$B \to b$

$F \to g$

$H \to e$

Step2: Find unreachable non-terminals and delete them

Reachable {S, B, g}

Delete F & H

$S \to Bb$

$B \to b$

# Normal Forms of CFG:

$V \longrightarrow$ Any

$S \rightarrow \underbrace{aSbABcdef}_{\text{anything}}$

Normal forms:

1) In DBMS: 1NF, 2NF, ...
2) Digital Logic: SOP, POS
3) Propositional Logic: CNF, DNF
4) In TOC: CNF CFG, GNF CFG

## CNF CFG

$V \rightarrow VV \mid T$

Example:

$S \rightarrow SS \mid SA \mid a$

$A \rightarrow AB \mid b$

$B \rightarrow d$

## GNF CFG

$V \rightarrow TV^*$

Example:

$S \rightarrow aSAB \mid b$

$A \rightarrow b$

$B \rightarrow dA$

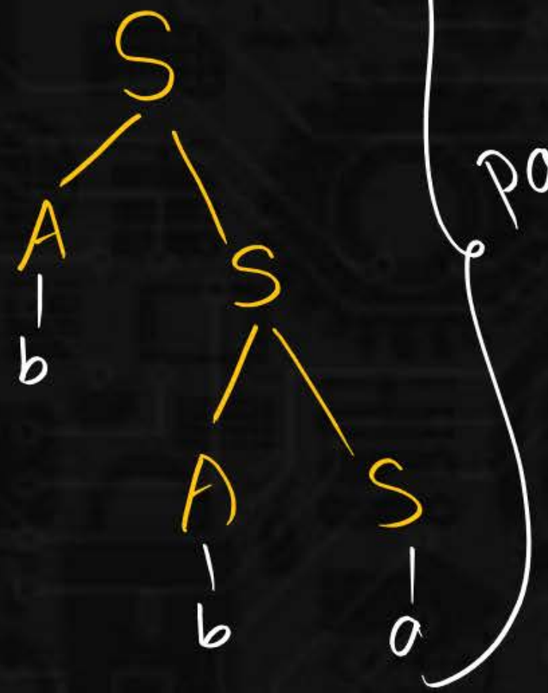
# CNF CFG

Chomsky Normal Form

$S \rightarrow AS \mid a$

$A \rightarrow b$

$w = bba$




Parse tree is binary tree

If Parse tree constructed using CNF CFG then it is always binary tree.

# Length of derivation using CNF CFG for n length string is $2n-1$

I) $|w| = 0$

⇓

no derivation using CNF

II) $|w| = 1$ [Assume $w = a$]

S

⇓

a

Length of derivation = 1

III) $|w| = 2$ [Assume $w = ab$]

S

⇓

AB

⇓

aB

⇓

ab

Length of derivation = 3

IV) $|w| = 3$

$w = aaa$

S

⇓

AB

⇓

aB

⇓

aCD

⇓

aaD

⇓

aaa

Length of derivation = 5

If $|w| = n$ ⇒ Length of derivation = $(2n-1)$ using CNF

| | |
|---|---|
| 1 | 1 |
| 2 | 3 |
| 3 | 5 |
| 4 | 7 |
| 5 | 9 |

# Length of derivation using GNF CFG for n length string is $n$

I) $n=0$ ✗

II) $n=1$ $(|w|=1)$

$\boxed{w=a}$

Length of derivation = 1


III) $n=2$ $(|w|=2)$

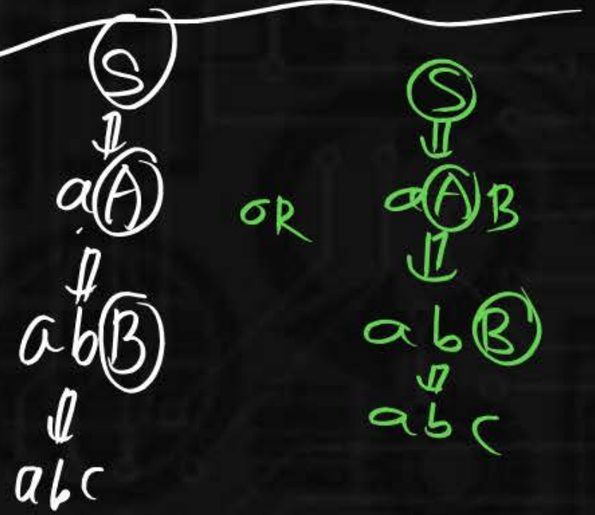
$w=ab$

Length of derivation = 2


IV) $n=3$ $(|w|=3)$

$w=abc$

Length of derivation = 3




IV*

# CYK Algorithm / Membership Algorithm

- Bottom up parsing
- Dynamic programming
- $O(n^3)$ for n length input
- Using CNF CFG, it verifies membership

# CYK Algorithm

$S \rightarrow AB \mid AS \mid a$

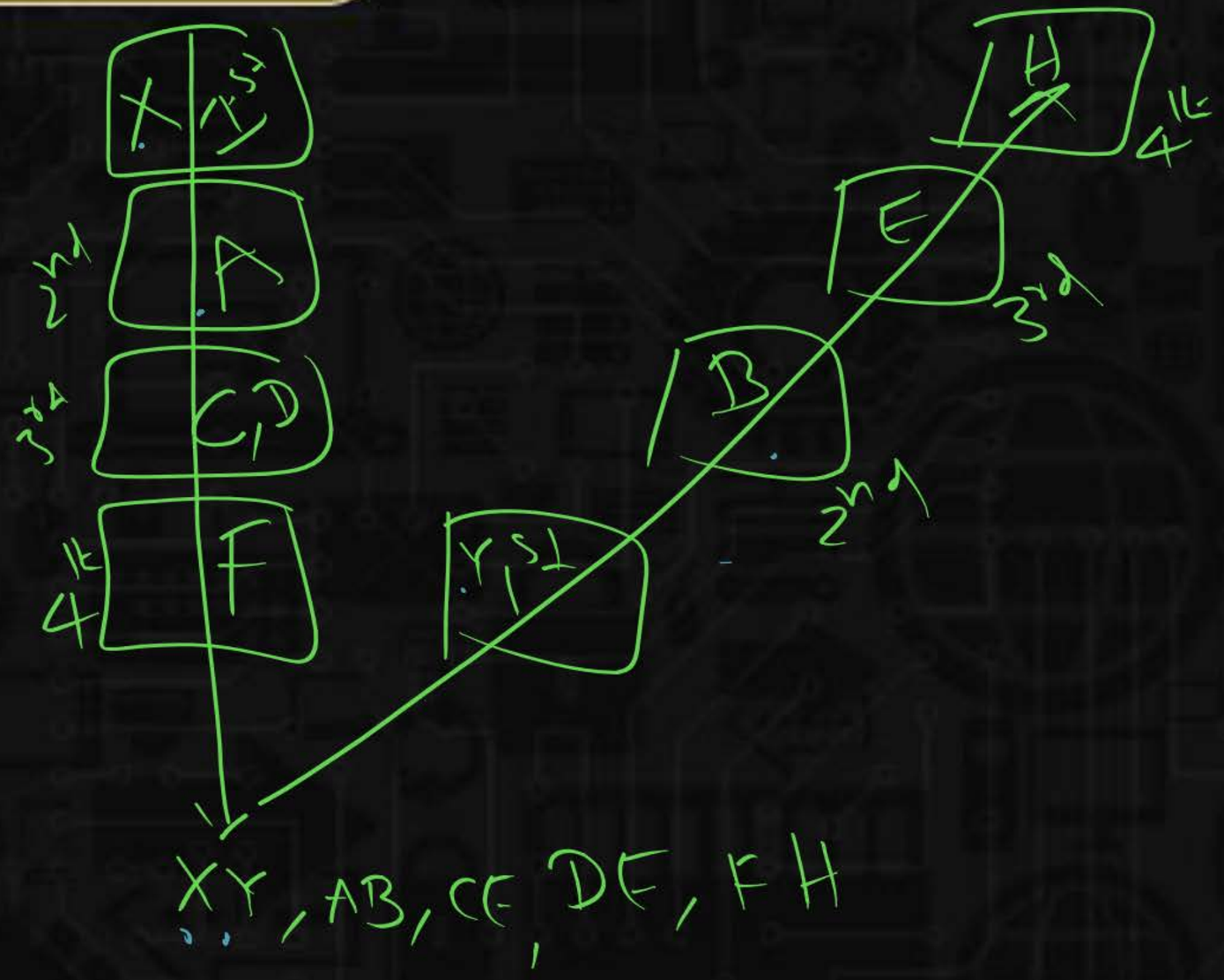
$A \rightarrow BA \mid b$

$B \rightarrow AS \mid BS \mid a$

$w = aba$

| a | b | a |
|---|---|---|
| {S, B} (a) | {A} (b) | {S, B} (a) |
| SA, BA {A} (ab) | AS, AB {S, B} (ba) | |
| SS, SB, BS, BB, AS, AB {B, S} (aba) | | |

S can derive aba


X
Y
A
C,D
F
Y
B
E
H
XY, AB, CE, DE, FH

# Summary

Simplification ✓

CNF & GNF ✓

CYK Algo ✓

CO-REL is complement of REL


THANK YOU GW
SOLDIERS !