Context Free Grammar (CFG).

$G(V, \Sigma, P, S)$

V: non-terminal (finite set of)

Σ: Terminal

S: Start

P: production rule.

$\subseteq V \times (V \cup \Sigma)^*$

$A \rightarrow \alpha$

$A \in V$ $\quad \alpha \in (V \cup \Sigma)^*$

$V = \{S\}$ $\quad S = \{S\}$

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

$S \rightarrow$ Apply per rule recursively

$S \Rightarrow \alpha$

$S \overset{*}{\Rightarrow} \alpha$

$\alpha \in \Sigma^*$

$L(G) = \{ w \in \Sigma^* \mid S \overset{*}{\underset{G}{\Rightarrow}} w \}$

$S \Rightarrow aSb \mid bSa \mid \varepsilon$

$S \Rightarrow aSb$

$\Rightarrow aaSbb$

Constructing CFG:

Given a language L, Describe G?

$L = L(G)$

$$\boxed{\begin{array}{c} L \subseteq L(G) \\ L(G) \subseteq L \end{array}}$$

Ex $L_1 = \{ a^n b^n \mid n \geq 0 \}$

$S \rightarrow \varepsilon$

$S \rightarrow aSb$

if L is divided in two regions, and each region is related to each to other.

then you need to generate both the regions in random. (saath saath).

$L_2 \{a^n b^m \mid n = 2m\}$.

$S \rightarrow aaSbb \mid \varepsilon$

$L_3 \{a^n b^m \mid \cancel{n=2m}) \; n \le m \le 2n\}$

$m \ge n$

$m \le 2n$

1 2

~~$S \rightarrow aSb \mid aaSb \mid \varepsilon$~~

$S \rightarrow aSb \mid aaSb \mid$

$S \rightarrow aSb \mid aSbb \mid \varepsilon$

$S \rightarrow AB \mid \varepsilon$

$A \rightarrow a$

$B \rightarrow b \mid bb$

$L_4 \{a^n b^n c^m \mid n, m \ge 0\}$

$S \rightarrow$

$L_4 = \{ w \in \{0,1\}^* \mid \#0's = \#1's \}$

$S \rightarrow aSb \mid bSa \mid \varepsilon \mid SS$

abbbaa- $S = AB \mid BA \mid \varepsilon \mid$

$A = a$

$B = b$

All regular languages are context free.

$D = (Q, \Sigma, \delta, q_0; F)$

$L = L(D)$

Constructing CFG

$V = \{ R_i \mid q_i \in Q \}$

$\Sigma = \Sigma$

$P =$

→ if $\delta(q_i, a) = q_j$ then a production

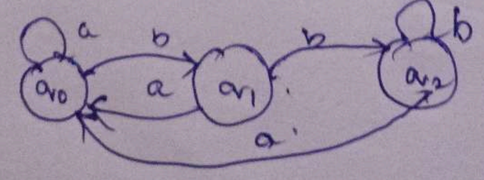
$R_i \rightarrow aR_j$

if q_i is final state

$R_i \rightarrow \varepsilon$ in P

$S = \{R_0\}$

$L = \{ w \in \{a,b\}^* \mid w$ ends with $bb \}$



$R_0 \to aR_0$

$R_0 \to bR_1$

$R_1 \to aR_0 \mid bR_2$

$R_2 \to aR_0 \mid bR_2 \mid \varepsilon$.

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

$L(G) = \{ w \in \Sigma^* \mid S \overset{*}{\underset{G}{\Rightarrow}} w \}$

$\alpha \overset{1}{\underset{G}{\Rightarrow}} \beta$ or $\alpha \Rightarrow \beta$

$\boxed{\overset{*}{\Rightarrow}}$ Transitive closure of $\Rightarrow$

$\alpha \overset{0}{\underset{G}{\Rightarrow}} \alpha$

$\alpha \overset{n+1}{\underset{G}{\Rightarrow}} \beta \qquad \alpha \overset{n}{\underset{G}{\Rightarrow}} \gamma \qquad \gamma \overset{1}{\underset{G}{\Rightarrow}} \beta$.

$L = \{ ww^R \mid w \in \{0,1\}^* \}$

↳ palindrome of even length.

$\boxed{S \to 0S0 \text{ or } 1S1 \text{ or } \varepsilon} \Rightarrow$ even length palindrome

$\boxed{S \to 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon}$

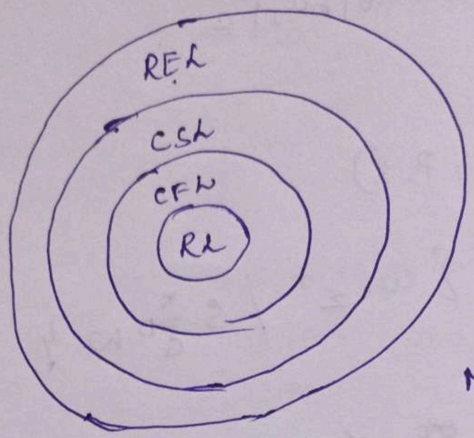
↳ palindrome of all lengths.

0S0

0110

$L = \{ w \in \{ (,) \}^* \mid w$ is balanced paranthesis $\}$

$S = (S) \mid \varepsilon \mid SS$



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

Each RL is a context free langauge.

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

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

$V = \{ R_i \mid q_i \in Q \}$

$\Sigma = \Sigma$

For $S = \{R_0\}$ Starting state of DFA

$P =$ if $\delta(q_i, a) = q_j$

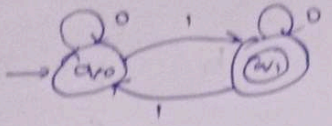
$\Downarrow$

$R_i \longrightarrow aR_j$

if R_i is final state.

$R_i \longrightarrow \varepsilon$.

Ex→



$L(M) = \{ w \in \Sigma^* \mid \text{has odd \# 1's} \}$

$R_0 \to 0R_0$
$R_0 \to 1R_1$
$R_1 \to 1R_0$
$R_1 \to 0R_1 / \varepsilon$.

left hand side only one non-terminal

$L(G) = L(M)$.

~~$S \to S1S0S$~~

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

$S \to aSbS \mid bSaS \mid \varepsilon$

ababba

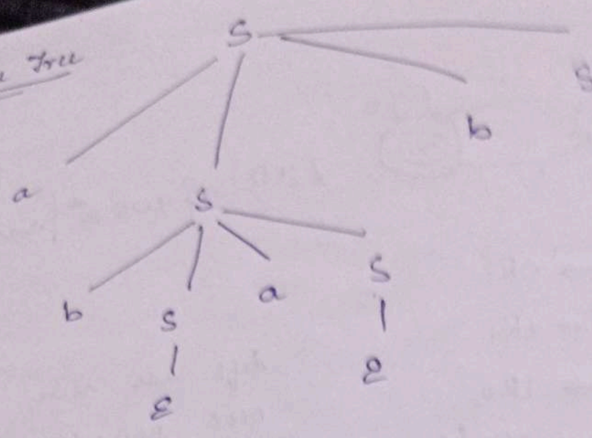
$S \to aSbS \quad (S \to aSbS)$

$\to abSaSbS \quad (S \to bSaS)$

$\to ababB \quad (S \to \varepsilon$
$S \to \varepsilon$
$S \to \varepsilon)$

Parse Tree

or Syntax tree



Properties:

① Syntactic structure of a string $w \in L(G)$.

② Root node of a parse tree is start symbol.

③ Each internal node will be a non-terminal.

④ Each leaf node is a terminal.

⑤ If an internal node labelled as R and the children of R from left to right are $\alpha_1, \alpha_2 \cdots \alpha_n$, then $R \rightarrow \alpha_1, \alpha_2 \cdots \alpha_n$ will be a production rule in G.

⑥. If you traverse all the leaf nodes of this parse tree left to right you will get the string w for which tree is find.

Sentential Form :

$S \xRightarrow{*} w$

$S \xRightarrow{*} \alpha$

$\uparrow (V \cup \Sigma)^*$

Sentential form

if α is made of only Σ, then α is known as sentence / string.

$S \Rightarrow aSbS \mid bSaS \mid \varepsilon$

$w = ababba$

left most non-terminal

$S \rightarrow aSbS$	$S \rightarrow aSbS$
$abSaSbS$	$S \rightarrow bSaS$
$abSaSbbSaS$	$S \rightarrow bSaS$
$ababba$	$S \rightarrow \varepsilon$
	$S \rightarrow \varepsilon$
	$S \rightarrow \varepsilon$
	$S \rightarrow \varepsilon$

replacing the leftmost non-terminal

$\Rightarrow$ leftmost derivation.

Sentential Form :

$S \xRightarrow{*} \omega$

$S \xRightarrow{*} \alpha$

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

Sentential form

if α is made of only Σ, then α is known as sentence / string.

$S \Rightarrow aSbS \mid bSaS \mid \varepsilon$

ω = ababba

left most non-terminal

$S \rightarrow aSbS$	$S \rightarrow aSbS$
$abSaSbS$	$S \rightarrow bSaS$
$abSaSbbSaS$	$S \rightarrow bSaS$
$ababba$	$S \rightarrow \varepsilon$
	$S \rightarrow \varepsilon$
	$S \rightarrow \varepsilon$
	$S \rightarrow \varepsilon$

replacing the leftmost non-terminal

$\Rightarrow$ leftmost derivation.

replacing rightmost non-terminal

$\Rightarrow$ right most derivation.

For a string $w \in L(G)$, if we can have more than one leftmost derivation, then grammar G is known as ambigous ~~language~~ grammar.

$\Rightarrow$ There will be 2 different parse trees.

Normal Form: to solve ambiguity.

CNF
GNF