Context Free Grammar

09 January 2022 10:43

→ Create a grammar from language

→ CFL (Context free language) CFG / Relationship btw gramm

c

⇒ GRAmmar & its classifications

Ram reads

<noun><verbs>

Sentences → <noun><verb>

<noun> → Ram / Sita / Sara / Ronit

<verb> → read / talking / sitting / walking

Based on this Grammar

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

$$= (V, T, P, S)$$

V → set of variables $\{S, A, B, C\}$ ⇒ (Capital let

Σ / T → set of terminals $\{a, b, c\}$ → (Small lett

S → starting symbol / $\underline{S \in V}$

$*$ → Zer

$+$ → One

P → set of production rule

$A \to \alpha$ where $A \in (V \cup T)^+$

$\cancel{\alpha} \Rightarrow \alpha \in (V \cup T)^*$

$S \to aA$

Structure of Grammar

L is language over alphabet A, then G for Consist of Rules

$x \to y$

$x/y \to$ denote string of symbols

$S \Rightarrow y$

Ex $A = \{a, b, c\}$ then G for language A^* can be described by following production rul

$S \to \epsilon$ (i)

$S \to aS$ (ii)

$S \to bS$ (iii)

$S \to cS$ (IV)

aacb

$S \Rightarrow aS$ (ii)

$\Rightarrow aaS$ (ii)

$\Rightarrow aacS$ (IV)

$\Rightarrow aacbS$ (iii)

$\Rightarrow aacb$ (i)

1. bbca
2. ababccab
3. cabcabbac

$G = (V, T, P, S)$

$V = \{S\}$ $T = \{0, 1\}$

P is defined as

(i) $S \to \epsilon$

(ii) $S \to 0S1$ Derive 000111

$S \Rightarrow 0S1$ (ii)

$\Rightarrow 00S11$ (ii)

$\Rightarrow 000S111$ (ii)

$\Rightarrow 000111$ (i)

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

ab

aabb

aaabbb . . .

$L = \{(ab)^n \mid n \geq 0\}$

Leftmost & Rightmost Derivation

FasbsAB

$\Rightarrow aabSAB$

$\Rightarrow aabbAAB$

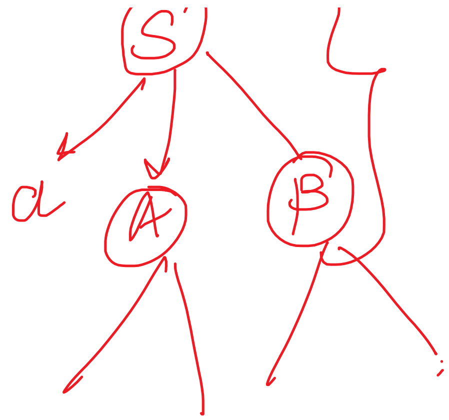
$\Rightarrow aabbaAB$

$\Rightarrow aabbabAB$

ABC

L

$\Rightarrow aabbabab$



Parse Tree

Notations

1. Non-terminal $\Rightarrow$ capital letter
2. Terminals/Auxillary letter $\Rightarrow$ smallcase
3. String of terminals uvwxyz
4. Sentential form (collection)

$S \Rightarrow aSBb$

5. $\rightarrow$ production
6. $\Rightarrow$ Process of derivation

$\stackrel{*}{\Longrightarrow}$ Zero or many

$\stackrel{+}{\Longrightarrow}$ one or more

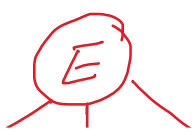
Leftmost / Rightmost Derivation

(a) Leftmost Derivation

$\rightarrow$ If at each step in derivation, a production is app the leftmost variable then it is leftmost derivatio

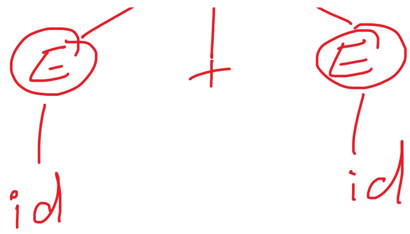
$(\{E\}, \{+, *, id\}, P, E)$

1. $E \rightarrow E + E$
2. $E \rightarrow E * E$
3. $E \rightarrow id$



<u>id+id</u>

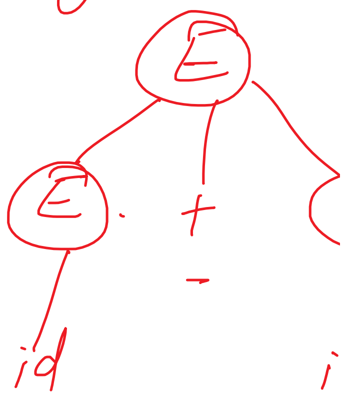
$$
\begin{aligned}
E &\Rightarrow E+E \quad (1)\\
&\Rightarrow id+E \quad (2)\\
&\Rightarrow id+id \quad (3)
\end{aligned}
$$



Rightmost Derivation:-

If at each step in the Derivation, a production rule is applied to rightmost variable, then it is rightmost

$$
\begin{aligned}
E &\Rightarrow E+E \quad (1)\\
&\Rightarrow E+id \quad (3)\\
&\Rightarrow id+id \quad (3)
\end{aligned}
$$



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

Production rule is

$S \rightarrow aAS / a$

$A \rightarrow SbA / SS / ba$

"aabbaa"

Soln ⇒ S is start symbol & Production rule are

1. $S \rightarrow aAS$
2. $S \rightarrow a$
3. $A \rightarrow SbA$
4. $A \rightarrow SS$

③ $A \to ba$

Leftmost

$S \Rightarrow aAS$ (1)
$\Rightarrow aSbAS$ (3)
$\Rightarrow aabAS$ (2)
$\Rightarrow aabbaS$ (5)
$\Rightarrow aabbaa$ (2)

Rightmost

$S \Rightarrow aAS$ (1)
$\Rightarrow aAa$ (2)
$\Rightarrow aSbAa$ (
$\Rightarrow aSbbaa$ (
$\Rightarrow aabbaa$ (

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

② CFL

bbaaba

Leftmost

Rule applied
2
5
4
1
6
3 ✓

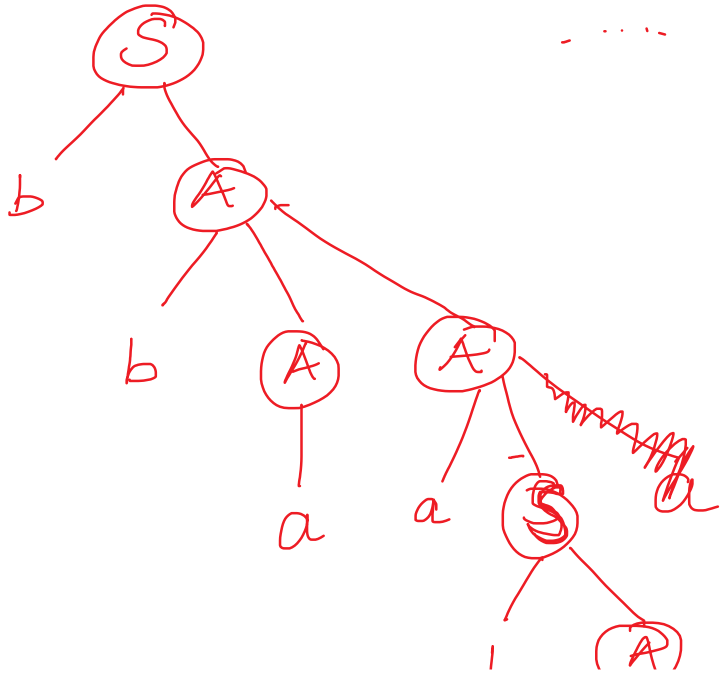
Rightmost

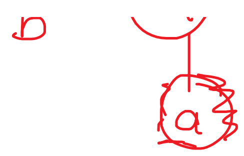
2
5
4

$S \Rightarrow bA$ (2)
$\Rightarrow bbAA$ (5)
$\Rightarrow bbAa$ (3)
$\Rightarrow bbaSa$ (4)
$\Rightarrow bbaaBa$ (1)
$\Rightarrow bbaaba$ (6)

Derivation Rule

Leftmost Parse tree





aaabbb

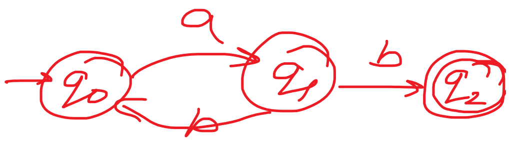
$S \rightarrow aSb$

$S \rightarrow ab$

In this case both Derivation are same

Context Free language

⇒ It is language generated by CFG



$L = \{ w \mid w \subset T^* \;\&\; \text{is derivable from start symbol} \}$

CFG $\quad S \rightarrow aSb \mid ab$

Soln :-

(1) $S \rightarrow aSb$

(2) $S \rightarrow ab$

(a) $S \Rightarrow ab$

(b) $S \Rightarrow aSb \quad (1)$

$\Rightarrow aabb \quad (2)$

(c) $S \Rightarrow aSb \quad (1)$

$\Rightarrow aaSbb \quad (1)$

$\Rightarrow aaabbb \quad (2)$

(d) $S \Rightarrow aSb \quad (1)$

$\Rightarrow aaSbb$ (1)

$\Rightarrow aaaSbbb$ (1)

$\Rightarrow aaaabbbb$

The language generated is

$L = \{ab, aabb, aaabbb \ldots\}$

$a^n b^n \quad n \geq 1$

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

(2) Start deriving string

(a) $S \Rightarrow aB$ (1)

$\Rightarrow ab$ (6)

$L = \{ab, ba, abab, ba$

(b) $S \Rightarrow bA$ (2)

$\Rightarrow ba$ (3)

(c) $S \Rightarrow aB$ (1)

$\Rightarrow abS$ (7)

$\Rightarrow abaB$ (1)

$\Rightarrow abab$

(d) $S \Rightarrow bA$ (2)

$\Rightarrow baS$ (4)

$\Rightarrow$ ~~baba~~ (

$\Rightarrow baaB$ (1)

$\Rightarrow baab$

L = { ab, ba, abab, abba, baab, bbaa, bbaaba, babbaa .

L = Here we are getting string of different length but with equal no of a's & b's

L = { x | x contain equal no of a's & b's }