Assignment - 02

Define context free grammer. obtain the context free grammer for the following.

(i) $L = \{ w w^R : w \in (a,b)^* \}$

(ii) write a CFG to generate balanced paranthesis where Bal = $\{ w \in \{ \}, (\}^*$; paranthesis are balanced$\}$. Justify the answer.

The context free grammer can be formally defined as a set denoted by $G = (T, V, P, S)$ where

T = terminal V = non-terminal P = production

S = start symbol

(i) $L = \{ w w^R : w \in (a,b)^* \}$ R = Reverse

$L = \{aa, bb, abba, baab \ldots\}$

$S \rightarrow \varepsilon \mid asa \mid bsb$

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

$P = \{ S \rightarrow \varepsilon \mid asa \mid bsb \}$

S is a start symbol

(ii) $L = \{ w \in \{()\}^* :$ parantheses are Balanced$\}$

$S \rightarrow \varepsilon$ $S \rightarrow (S)$ $S \rightarrow (SS)$

$G = (V, T, P, S)$ $V = \{S, (,)\}$

$T = \{ (,) \}$

$P : \{ S \rightarrow S$, $S \rightarrow (S)$ $S \rightarrow SS \}$

S is a start Symbol.

Define leftmost and right most derivations with Examples.

leftmost derivation: In the derivation process if a left most variable is replaced at Every step, then the derivation is said to be leftmost derivation.

Rightmost derivation: In the derivation process, if a right most variable is replaced at every step, then the derivation is said to be right most derivation.

$E \to E+E$

$E \to E-E$

$E \to E*E$

$E \to E/E$

$E \to E^E$

$E \to id$

Derivation for the string id+id*id

$E \overset{+}{\underset{lm}{\Rightarrow}} E+E$

$\Rightarrow id+E$

$\Rightarrow id+E*E$

$\Rightarrow id+id*E$

$E \overset{+}{\underset{lm}{\Rightarrow}} id+id*id$

$E \underset{rm}{\Rightarrow} E+E$

$\Rightarrow E+E*E$

$\Rightarrow E+E*id$

$\Rightarrow E+id*id$

$E_{rm} \Rightarrow id+id*id$

Define PDA and instantaneous of PDA obtain a PDA to accept the language $L=\{wcw^R : w \in (a,b)^*\}$ Draw the transition diagram of PDA. Show the moves by this PDA for the string abbcbba.

PDA is a seven-tuple $M=(Q,\Sigma,\Gamma,\delta,q_0,z_0,F)$

Q = A finite set of states

Σ = A finite input alphabet

Γ - A finite stack alphabet

q_0 - The initial / starting state

z_0 - A starting stack symbol

F - A set of final / accepting states

δ: A transition function

where,

$\delta: Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to$ finite subsets of $Q \times \Gamma^*$

$L=\{wcw^R : w \in (a,b)^*\}$ considered to be ϵ

Here the separator b/w w and w^R

for instance

aa = aa

abba = abϵba

baab = baϵab

the ID for this PDA will be

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

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

$\delta(q_0, a, b) = (q_0, ab)$

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

$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, \varepsilon, a) = (q_1, a)$

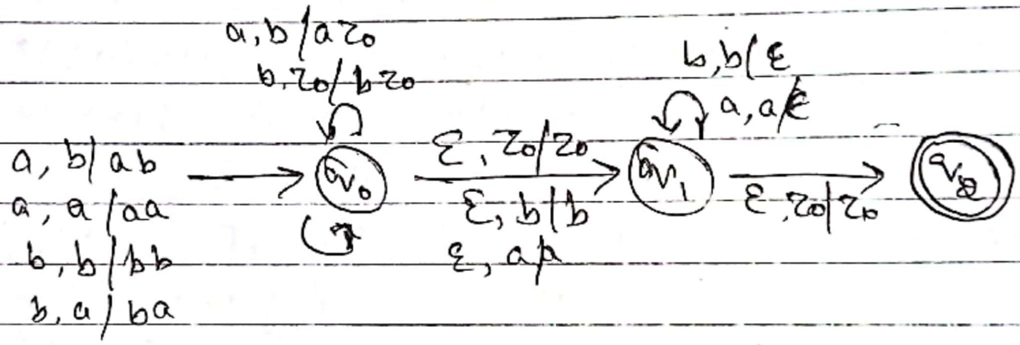
$\delta(q_0, \varepsilon, b) = (q_1, b)$

$\delta(q_0, \varepsilon, Z_0) = (q_1, Z_0)$

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

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

$\delta(q_1, \varepsilon, Z_0) = (q_2, Z_0)$ Accept



Simulation for abbcbba

$\delta(q_0, abbcbba, Z_0) \vdash \delta(q_0, bbcbba, aZ_0)$

$\delta(q_0, bcbba, baZ_0)$

$\delta(q_0, cbba, bbaZ_0)$

$\delta(q_0, \varepsilon bba, bbaZ_0)$

$\delta(q_1, bba, bbaZ_0)$

$\delta(q_1, ba, baZ_0)$

$\delta(q_1, a, aZ_0)$

$\delta(q_1, a, aZ_0)$

$\delta(q_1, \varepsilon, Z_0)$

$\delta(q_2, Z_0)$

Accept.

(4) what is CNF and GNF?. convert the grammer in CNF

$S \rightarrow ABa$

$A \rightarrow aab$

$B \rightarrow ac$

> CNF is a type of Normal form in which CFG of the form, it is said to be CNF. NT→NT : NT and NT→Terminal.

GNF : is a type of normal form we, will discuss one more normal form called GNF.

NT → ε. NT

$S \rightarrow ABa$ $\quad A \rightarrow aab$ $\quad B \rightarrow Ac$

$S \rightarrow AP$ $\quad A \rightarrow QR$ $\quad B \rightarrow AW$

$B \rightarrow BQ$ $\quad R \rightarrow QT$ $\quad W \rightarrow c$

$Q \rightarrow a$ $\quad T \rightarrow b$

Hence, ~~A~~ $S \rightarrow AP$, $P \rightarrow BQ$, $Q \rightarrow a$, $A \rightarrow QR$, $R \rightarrow QT$

$T \rightarrow b$ $\quad B \rightarrow AW$ $\quad W \rightarrow c$.

(5) for the following CFG $S \rightarrow$ ~~...~~ $aSbb \mid aab$, obtain the corresponding PDA.

> $S \rightarrow aSbb$

$S \rightarrow aSBB$ $\quad B \rightarrow b$

$S \rightarrow aab$ $\quad S \rightarrow aAB$

Now, Consider the grammer,

$S \rightarrow aSBB$, $\quad S \rightarrow aAB$ $\quad A \rightarrow a$, $B \rightarrow b$

The PDA Can be $\quad \delta(q_0 . a , S) = \delta(q_0\ SBB)$

$\delta(q_0 , a , S) = \delta(q_0\ AB)$

$\delta(q_0\ a\ A) = \delta(q_0\ \varepsilon)$

$\delta(q_0\ b\ B) = \delta(q_0\ \varepsilon)$

$\delta(q_0\ \varepsilon\ z_0) = \delta(q_0\ \varepsilon)$ Accept.

(6) Define ambiguous grammer. & ST following expression grammer is ambiguous over the string id * id + id write Equivalent unambiguous grammer for the Same grammer.

$E \rightarrow E+E \quad E \rightarrow E/E$

$E \rightarrow E-E \quad E \rightarrow id$

$E \rightarrow E*E$

> If there are 2 parse tree with same side left or right for same grammar then the grammar is called ambiguous grammer.

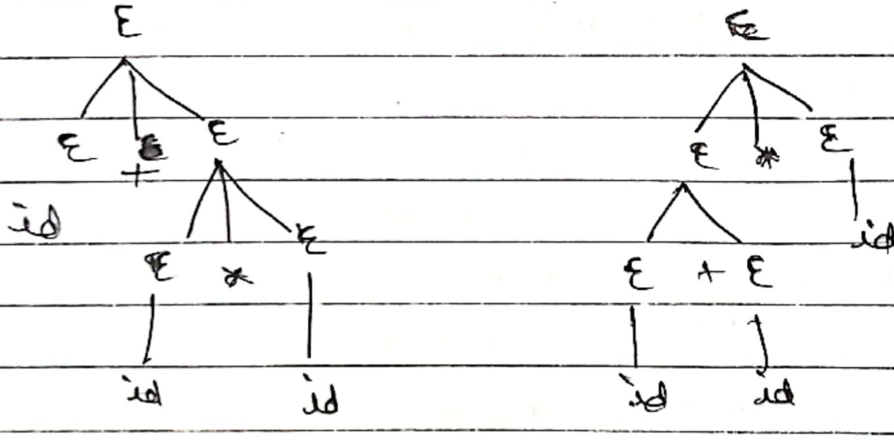
$E \rightarrow E+E \qquad E \rightarrow E*E$

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

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

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

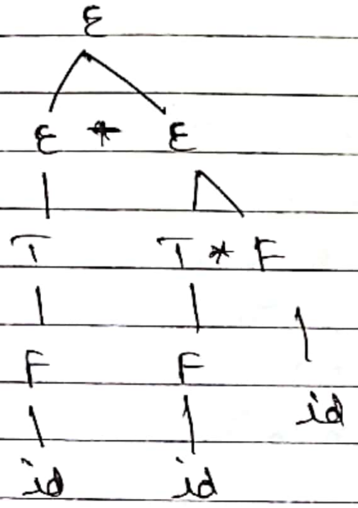
$\Rightarrow id+id*id \qquad E \Rightarrow id+id*id$



∴ two parse tree are obtained the Same Sentences id+id*id by applying left most derivation, the grammar is ambiguous. We should take care that highest precedence operation should be at the least level

So, we will introduce several difference symbols
like $E \rightarrow E+T/T$ } once we have reached T, we
$T \rightarrow T*F/F$ } can't generate any +
$F \rightarrow id$

Now the unambiguous grammer would be id + id * id



So we are taking care of precedence by defining difference level.

(7) Define PDA, obtain a PDA to accept the foll. language
$L = \{ n_a(w) = n_b(w) \text{ where } n \geq 1 \}$
Draw the transistion diagram for PDA. Also show the moves made by the PDA for the string "aabbab"

>

This is a language of equal number of a's and equal number of b's. When stack is empty then whatever we read either 'a' or 'b' we will simply push it onto the stack. If we read 'a' and at the top of the stack is b is present then it will be erased by ε, vice versa.

$L = \{ n_a(w) = n_b(w) \text{ where } n \geq 1 \}$

The instantaneous description for this language is given below.

$\delta(q_0\ a\ z_0) = (q_0\ a\ z_0)$

$\delta(q_0\ b\ z_0) = (q_0, bz_0)$

$\delta(q_0\ a\ a) = (q_0\ aa)$

$\delta(q_0\ b\ b) = (q_0\ bb)$

$\delta(q_0\ a.\ b) = (q_0\ \varepsilon)$

$\delta(q_0\ b.\ a) = (q_0\ \varepsilon)$

$\delta(q_0\ \varepsilon\ z_0) = (q_1\ z_0)$

$\delta(q_0, aabbab, z_0) \vdash (q_0, abbab, az_0)$
$\vdash (q_0, bbab, aaz_0)$
$\vdash (q_0, bab, z_0)$
$\vdash (q_0, ab, bz_0)$
$\vdash (q_0, b, z_0)$
$\vdash (q_0, \varepsilon, bz_0)$
$\vdash (q_1, b)$

Accept state.

(8) Define inherently ambiguous language with example.

> Inherently ambiguous language is a context free language that has no non ambiguous grammar.

Eg:-

$\{a^i b^j c^k \mid i = j \text{ or } j = k\}$

Let L be a CFL. if every context free grammar G with $L = L(G)$ is ambiguous, then L is said to be inherently ambiguous language.

(9) Let G be the grammar.

$S \rightarrow aB \mid ba$

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

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

for the string aaabbabbba find

(a) left most derivation (b) Right most derivation (c) parse tree

Consider left most derivation,

$S \Rightarrow aB \quad (S \rightarrow aB)$

$\Rightarrow aaBB \quad (B \rightarrow aBB)$

$\Rightarrow aa\cancel{a}BBB \quad (B \rightarrow aBB)$

~~$\Rightarrow$ aaabbABB ($S \rightarrow bA$)~~

$\Rightarrow aaabSBB \quad (B \rightarrow bS)$

$\Rightarrow aaabbABB \quad (S \rightarrow bA)$

$\Rightarrow aaabbaBB \quad (A \rightarrow a)$

$\Rightarrow aaabbabB \quad (B \rightarrow b)$

$\Rightarrow aaabbabbS \quad (B \rightarrow bS)$

$\Rightarrow aaabbabbbA \quad (S \rightarrow bA)$

$\Rightarrow aaabbbabbba \quad (A \rightarrow a)$