Spring - 2024

Group-A

$\boxed{1}$

(a) Construct a CFG that generates the following language over the alphabet $\Sigma = \{0, 1\}$.

$L = \{W \mid$ the length of w is odd and its middle symbol is '0' $\}$.

Ans.

$L = \{0, 000, 101[illegible], 001, 10001, \ldots\}$

CFG,

$S \rightarrow 0 \mid 0S1 \mid 1S1 \mid 0S1 \mid 1S0$.

Generate - 10001,.

$S \rightarrow 1S1$

$\rightarrow 10S01 \quad [S \rightarrow 0S0]$

$\rightarrow 10001 \quad [S \rightarrow 0]$

Or,

$S \rightarrow A01B$

$\rightarrow 1A \mid 0A \mid \epsilon$

$\rightarrow 1B \mid \epsilon$

Again Generate, 1011

$S \rightarrow A01B$

$\rightarrow 1A01B \quad [A \rightarrow 1A]$

$\rightarrow 101B \quad [A \rightarrow \epsilon]$

$\rightarrow 1011B \quad [B \rightarrow 1B]$

$\rightarrow 1011 \quad [B \rightarrow \epsilon]$

OR Construct a CFG corresponding to the regular expression $(0+1)^*011^*$ over the alphabet $\Sigma = \{0, 1\}$. This is, any string described as "any combination of '0' and '1' followed by '01' ending with any number of 1's" belongs to the associated language.

Ans.

Productions:

$S \rightarrow A011B$

$A \rightarrow 0A \mid 1A \mid \epsilon$

$B \rightarrow 1B \mid \epsilon$

(b) Give the formal definition of Context-free Grammar (CFG) and Chomsky Normal Form (CNF).

Ans.

CFG :

A CFG is defined as a 4-tuple :
$G = (V, \Sigma, P, S)$.

i. V : A finite set of variable.

ii. Σ : A finite set of terminal symbols.

iii. P or R : A finite set of production rules of the form $A \to \alpha$, where $A \in V$ and $\alpha \in (V \cup \Sigma)^*$.

iv. S : A start symbol ($S \in V$).

CNF :

A CFG is in CNF if every rule is of the form,

$A \to BC$
$A \to a$

Where a is any terminal and A, B, C are any variable, except that B and C may not be the start variable. In addition, the rule $S \to \epsilon$ is permitted where S is the start

(C) When is a grammar called ambiguous?

Consider the following grammar -

$S \to SAB \mid \epsilon$, $A \to AaB \mid a$, $B \to AS \mid b$

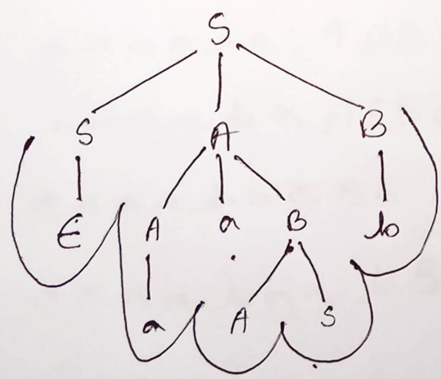
This grammar is ambiguous, Show in particular that the string aaaabaabbabab has two;

i. Parse trees ii. Leftmost derivations iii. Rightmost derivations.

Ans.

In CFG, ambiguity refers to a situation where a grammar generates the same string in several different ways.

(i)



Generated String -

aaaabaabbabab.

(ii)

$S \rightarrow SAB$

$\rightarrow AB \quad [S \rightarrow \epsilon]$

$\rightarrow AaBB \quad [A \rightarrow AaB]$

$\rightarrow aaBB \quad [A \rightarrow a]$

$\rightarrow aaASB \quad [B \rightarrow AS]$

$\rightarrow aaaBSB \quad [A \rightarrow a]$

$\rightarrow aaabSB \quad [B \rightarrow b]$

$\rightarrow aaabSABB \quad [S \rightarrow SAB]$

$\rightarrow aaabABB \quad [S \rightarrow \epsilon]$

$\rightarrow aaabaBB \quad [A \rightarrow a]$

$\rightarrow aaaabaBB \quad [A \rightarrow a]$

$\rightarrow aaaabaASB \quad [B \rightarrow AS]$

$\rightarrow aaaabaSB \quad [A \rightarrow a]$

$\rightarrow aaaabaSABB \quad [S \rightarrow SAB]$

$\rightarrow aaaabaASBB \quad [S \rightarrow \epsilon]$

$\rightarrow aaaabaaBB \quad [A \rightarrow a]$

$\rightarrow aaaabaabB \quad [B \rightarrow B]$

$\rightarrow aaaabaabb\ abab.$

(iii)

$S \rightarrow SAB$

$\rightarrow SAb$ [$B \rightarrow b$]

$\rightarrow SAaBb$ [$A \rightarrow AaB$]

$\rightarrow SAab$ [$A \rightarrow a$]

$\rightarrow Sab$ [$S \rightarrow \epsilon$]

$\rightarrow SABab$ [$S \rightarrow SAB$]

$\rightarrow SAASab$ [$B \rightarrow AS$]

$\rightarrow SAAaBSAb$ [$S \rightarrow SAB$]

$\rightarrow SAAabSab$ [$A \rightarrow a$]

$\rightarrow SAaabSab$ [$A \rightarrow a$]

$\rightarrow SaaabSab$ [$b \rightarrow b$]

$\rightarrow SaaabABab$ [$S \rightarrow SAB$]

$\rightarrow SaaaabAbab$ [$S \rightarrow \epsilon$]

$\rightarrow aaaabaabbabab$

$\boxed{2}$

(a) Using the pumping lemma show that the following languages are not context-free:

$L = \{a^n b^n c^i \mid i \leq n, i \geq 0, n \geq 0\}$,

$L = \{x^n y^n z^n \mid n \geq 1\}$.

Ans. For first language,

Assume L is context-free.

Choose $w = a^p b^p c^p$, where p is the pumping length.

Split $w = uv~~x~~xy$; with $|vxy| \leq p$ and $v, y \neq \epsilon$

Pumping v and y increases a's or b's but keeps c's constant, violating $i \leq n$.

Hence, L is not context-free.

For second language,

Assume L is context-free.

Choose $w = x^p y^p z^p$, where p = Pumping length.

Split $w = uvxyz$, with $|vxy| \leq p$ and $v, y \neq \epsilon$

Pumping v or y creates strings where the counts of x, y, z are unequal, violating $n = n = n$.

Hence, L is not context-free.

OR Can you give a context-free grammar (CFG) for the following language over the alphabet $\Sigma = \{a, b\}$ –

All strings in the language $\{a^n b^{2n} c^{4n} \mid n \geq 0\}$

If you cannot, justify the reason.

Ans.

The language is not context-free.

Because it requires enforcing multiple constraints:

$|b| = 2|a|$ and $|c| = 4|a|$, which CFGs cannot handle simultaneously.

This dependency among a, b, and c goes beyond the power of CFG. Hence, no CFG can generate L.

(b) How can CFG be simplified? Write down the procedure for eliminating unit productions from a CFG. Remove the unit productions from the following grammar –

$S \to AC$, $A \to a$, $C \to D|d$ $D \to E$, $E \to b$

Ans.

CFG can be made simpler by removing all the extragenous symbols while yet persuing a converted grammar that is equivalent to the original grammar.

Step-1: Remove unit production,

$A \to AB$

$A \to a$

$B \to a|b$

$C \to a$

$D \to a$

$E \to a$

Remove unreachable state,

$S \to AB$

$A \to a$

$B \to ab$.

(C) Convert the following CFG into an equivalent CFG in CNF :

$S \to TX$

$T \to 0T0 \mid 1T1 \mid \# X$

$X \to 0X \mid 1X \mid \epsilon$

Ans.

Let's convert ~~the~~ CFG to CNF –

Step-1: Removing null productions :-

$S_0 \to S$

$S \to TX \mid T$

$T \to 0T0 \mid 1T1 \mid \# X \mid \#$

$X \to 0X \mid 1X \mid 0 \mid 1$

Step-2: Removing unit productions:-

$S_0 \to S$

$S \to TX \mid 0T0 \mid 1T1 \mid \# X \mid \#$

$T \to 0T0 \mid 1T1 \mid \# X \mid \#$

$X \to 0X \mid 1X \mid 0 \mid 1$

Step-3 : Remove long productions:-

$S_0 \rightarrow S$

$S \rightarrow TX \mid 0A \mid 1B \mid \#X \mid \#$

$T \rightarrow 0C \mid 1D \mid \#X \mid \#$

$X \rightarrow 0X \mid 1X \mid 0 \mid 1$

$A \rightarrow T0$

$B \rightarrow T1$

$D \rightarrow T1$

$C \rightarrow T_0$

Step-4: Replace with CNF terminals:-

$S_0 \rightarrow S$

$S \rightarrow TX \mid EA \mid FB \mid \#X \mid \#$

$T \rightarrow EC \mid FD \mid \#X \mid \#$

$X \rightarrow EX \mid FX \mid E \mid F$

$A \rightarrow T_0$

$B \rightarrow T_1$

$C \rightarrow T_0$

$D \rightarrow T_1$

$E \rightarrow 0$

$F \rightarrow 1$

Step-6: Final grammar in CNF :-

$S_0 \rightarrow S$

$S \rightarrow TX \mid EA \mid FB \mid X$

$T \rightarrow EC \mid FD \mid X$

$X \rightarrow EX \mid FX \mid E \mid F$

$A \rightarrow T_0$

$B \rightarrow T_1$

$C \rightarrow T_0$

$D \rightarrow T_1$

$E \rightarrow 0$

$F \rightarrow 1$