JECRC University

School of Engineering Semester - V^{th}

FLAT Assignment - 3 (Tut - 06)

Section - A (2*5 = 10 Marks)

Q. (1.) If $S \to aSb \mid aAb$, $A \to bAa$, $A \to ba$. Find out the CFL.

Solⁿ:- We have,

$S \to aSb \mid aAb$
$A \to bAa$
$A \to ba$

Now, finding the CFL :-

$S \to aSb$
$\to aaAbb$
$\to aabAabb$
$\to aabbaabb$
$S \to a^4 b^4$

Q. (2.) G is $S \to aS \mid bS \mid a \mid b$ find $L(G)$.

Solⁿ:- We have,
$S \to aS \mid bS \mid a \mid b$

$\because S \to a$ $\quad w_1 = a$
$S \to b$ $\quad w_2 = b$

$S \to aS$
$\to aa$
$S \to bS$
$\to bb$
$S \to aS$
$\to ab$
⋮

$w_3 = aa$
$w_4 = bb$
$w_5 = ab$
⋮
w_n

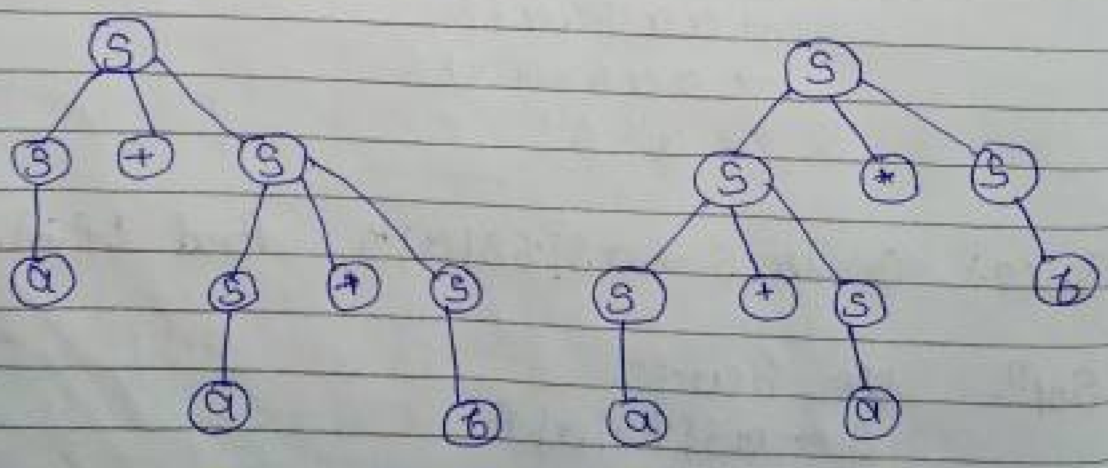
Thus, $L(G) = \{ w \in \{a, b\}^* \ | w| \geq 1 \}$

Q. (3) Define ambiguous grammar with example.

Ans:- Any grammar is said to be ambiguous or having the property of ambiguity if & only if it posses more than one leftmost derivation & rightmost derivation.

e.g. Prove that the grammar G is ambiguous in nature.
$(a + a * b)$
$S \to S+S \mid S*S \mid a \mid b$

Sol^n:-



Here is two derivation trees for $a + a * b$.
So, it is ambiguous in nature.

DATE: / /

PAGE NO.:

Q.(4.) Write the definition of L(G).

Ans:- The set of all strings that can be derived from a grammar is said to be the language generated from that grammar. A language generated by a grammar G is a subset formally defined by

$$L(G) = \{W \mid W \in \Sigma^*, S \Rightarrow_G W\}$$

If $L(G1) = L(G2)$, the grammar G1 is equivalent to the grammar G2.

Q.(5.) L be the set of all palindrome over {0, 1} Construct Grammar

Solⁿ:- We have,
the L be the set of all palindrome over {0, 1}
So,

$$L = \{\varepsilon, 0, 1, 00, 11, 000, 111, 010, 101, 0000, 1111, 0110, 1001, \ldots\}$$

Thus,
The grammar is

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

DATE: / /

PAGE NO.:

Section B (7*3 = 21 Marks)

Q.(1) Find the grammar for the language $L = \{a^i b^j c^j$, where $i >= 0, j >= 1\}$

Sol^n:- We have,

$L = \{a^i b^j c^j$, where $i >= 0$, $j >= 1\}$

$\because a^i b^j c^j$

- We split into two parts i.e.

$\underbrace{a^i}_{A} \underbrace{b^j c^j}_{B}$

- For a^i (cond^n $i >= 0$)

So, $A = a^0$
a^1
a^2
a^3
$\vdots$
a^n

- & For $b^j c^j$ (cond^n $j >= 1$)

So, $B = b^1 c^1$
$b^2 c^2$
$b^3 c^3$
$b^n c^n$

DATE: / /
PAGE NO.:

- Thus, the grammar for the language is

$S \to AB$

$A \to aA \mid \lambda$

$B \to bBc \mid bc$ Ans

Q. (2.) Construct the grammar for the language $L = \{a^n b a^n \mid n >= 1\}$.

Solⁿ:- We have,

The language,

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

For $n = 1$, aba
$n = 2$, aabaa
$n = 3$, aaabaaa
$n = 4$, aaaabaaaa
⋮

Thus,
The ~~languo~~ grammar for the language is

$S \to aSa \mid aba$ or $S \to aSa$
$S \to aba$

DATE : / /

PAGE NO. :

Q.(3.) What is the language generated by the grammar $G = (V, T, P, S)$ where $P = \{S \to ab, S \to aSb\}$.

Sol^n We have,

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

where $P = \{S \to ab, S \to aSb\}$

$\because S \to aSb$ & $\boxed{S \to ab}$

$\to aabb$

$\to a^2 b^2$

$a^3 b^3$

$a^4 b^4$

$\vdots$

$a^n b^n$

Thus, $L = \{ab, aabb, aaabbb, aaaabbbb, aaaaabbbbb, \ldots\}$

Hence,

$L(G) = \{a^n b^n \mid n \geq 1\}$ Ans

DATE: / /

PAGE NO.:

Section C (11 * 3 = 33 Marks)

Q. (1) Explain the Normal Form And convert the following CFG in

$G = (\{S, A, B\}, \{a, b\}, \{S \to AB, A \to aA | bB | b, B \to b\}, S)$, in GNF

$G = (\{S, A, B, D\}, \{a, b, d\}, \{S \to aAD, A \to aB | bAB, D \to d, B \to b\}, S)$, in CNF.

Solⁿ:- A normal form for a context-free grammar (CFG) is a standardized way of writing production rules to facilitate parsing & theoretical analysis. The main normal forms are Chomsky Normal form (CNF) & Greibach Normal Form (GNF).

(i) Chomsky Normal Form (CNF) — A CFG is in CNF if the productions are in the following forms:

$A \to a$

$A \to BC$

where A, B & C are non-terminal & a is a terminal.

(ii) Greibach Normal Form (GNF) — A CFG is in Greibach Normal form if the productions are in the following forms:

DATE: / /

PAGE NO.:

$A \to b$

$A \to bC_1C_2 \ldots C_n$

where $A, C_1 \ldots C_n$ are non-terminals & b is a terminal.

- Conversion CFG to GNF

We have,

$S \to AB$

$A \to aA \mid bB \mid b$

$B \to b$

Step 1:- Remove NULL production
There are no null production

Step 2:- Remove unit production
There are no unit production

Step 3:- Obtain GNF:-

$S \to aAB \mid bBB \mid bB$

$A \to aA \mid bB \mid b$

$B \to b$

- Convert CFG to CNF

We have,

$S \to aAD$

$A \to aB \mid bAB$

$D \to d$

$B \to b$

DATE: / /
PAGE NO.

Step 1:- Remove NULL production
There are no null production

Step 2:- Remove unit production
There are no unit production

Step 3:- Obtain CNF (N.T → T)

(i) S → EAD (Replace a with E)
E → a

A → EB | FAB (Replace ~~F~~b with F)
F → b

D → d
B → b

(ii) S → EG (Replace AD with G)
G ~~A~~ → AD
E → a

A → EB | FH (Replace AB with H)
H → AB
F → b

D → d
B → b

Hence, it is GNF form.

DATE: / /
PAGE NO.:

Q.(2) Explain the pumping lemma for regular language. And using pumping lemma prove that the language $A = \{a^n b^n \mid n >= 0\}$ is not regular.

Ans:- Pumping lemma is used to prove that a language is not regular but it cannot be used to prove that a language is regular.

It states that if A is a regular language, then A has a pumping length 'p' such that any string 's' whose $|S| \geq P$ may be divided into 3 parts $S = xyz$ such that the following conditions must be true:

(i) $xy^i z \in A$ for every $i \geq 0$.
(ii) $|y| > 0$
(iii) $|xy| \leq P$

* To prove: The language $A = \{a^n b^n \mid n >= 0\}$ is not regular.

Step 1: Analysis
$n = 0$; $a^0 b^0$, Length 0
$n = 1$, ab, length 2
$n = 2$, aabb, length 4
$n = 3$, aaabbb, length 6

The language consist strings with length is always even.

DATE: / /
PAGE NO.:

Step 2:- Assumption

It is assume that the language A is regular

Step 3:- For $w = xy^iz$ using pumping lemma & for given language $A = a^n b^n$.

Step 4:- For $i = 0$, $w = xy^0z$
For $i = 1$, $w = xyz$
For $i = 2$, $1 \le |xy^2z| \le n$

Since $a^n b^n \rightarrow n + n = 2n$

Adding $2n$ on both sides

$$1 + 2n \le |xy^2z| \le n + 2n$$

$$\Rightarrow 1 + 2n \le |xy^2z| \le 3n$$

$$\Rightarrow 1 - 1 + 2n \le |xy^2z| \le 3n + 1$$

$$2n \le |xy^2z| \le 3n + 1$$

For $n = 1$, length 2, 3, 4

The language consist of strings with length is not always even.

This makes contradiction hence

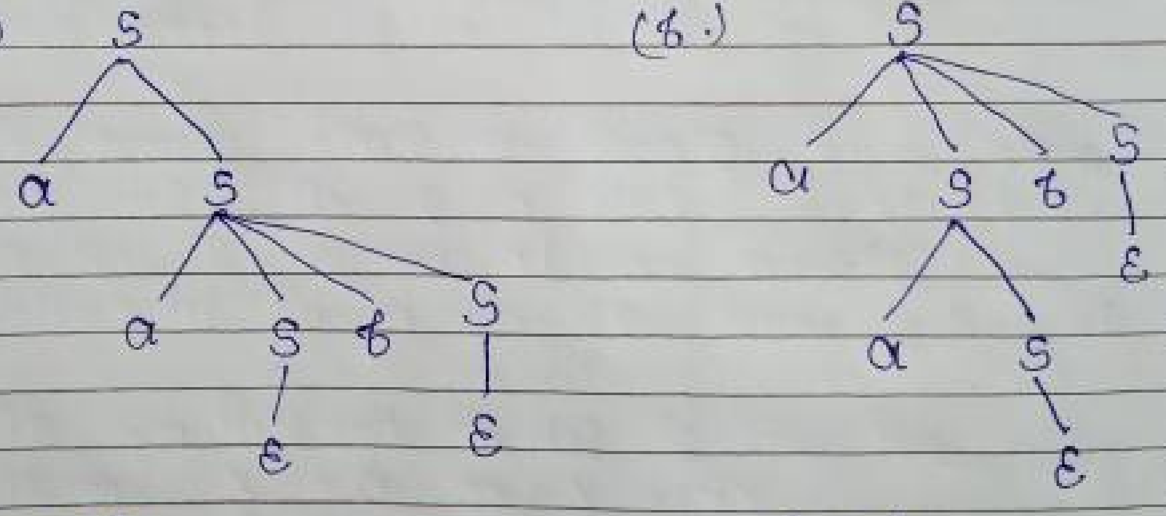
the given language is not regular.
Hence proved.

Q. (3) (A.) consider the grammar $P = \{S \rightarrow aS \mid aSbS \mid \epsilon\}$ is ambiguous by constructing:
(i) two parse tree
(ii) two leftmost derivation

Sol^n:- We have,
the grammar $P = \{S \rightarrow aS \mid aSbS \mid \epsilon\}$

(i) two parse tree

Sol^n:- (a) (b.)



aab aab

Hence, it is two parse trees for aab

So, it is ambiguous in nature.

DATE : / /

PAGE NO. :

(ii) two leftmost derivation

s|ε}

(a) $S \rightarrow aS$

$aaSbS$,

$aa\varepsilon bS$

$aa\varepsilon b\varepsilon$

$\rightarrow aab$

~~aab~~

(aab)

$S \rightarrow aSbS$

$aaSbS$

$aa\varepsilon bS$

$aa\varepsilon b\varepsilon$

$\rightarrow aab$

(aab)

Hence, it is two leftmost derivation for aab

So, it is ambiguous in nature.

(B.) Explain the Chomsky classification of grammars in details with example.

<u>Ans:-</u> Chomsky gave a mathematical model of grammar which is effective for writing computer languages

The four types of grammar according to Noam Chomsky are:-

Grammar Type	Grammar accepted	language accepted	Automation
(i) Type-0	Unrestricted grammar	Recursively enumerable language	Turing Machine

DATE: / /

PAGE NO.:

(ii)	Type - 1	Context sensitive grammar	Context sensitive language	linear bounded automation
(iii)	Type - 2	Context free grammar	Context free Language	Pushdown automata
(iv)	Type - 3	Regular grammar	Regular language	Finite state automation

(a) Type - 3 grammar

- Type - 3 grammars generate regular languages.
- It must have a single non-terminal on the LHS & a RHS consisting of a single terminal or single terminal followed by a single non-terminal

The productions must be in the form $X \rightarrow a$ or $X \rightarrow aY$

where $X, Y \in$ Non-terminal
$a \in$ Terminal

e.g. — $X \rightarrow \epsilon$
$X \rightarrow a | aY$
$Y \rightarrow b$

DATE: / /
PAGE NO.:

(b) Type - 2 grammar

↳ Type - 2 grammars generates context free languages.

↳ The productions must be in the form

$G \rightarrow (V \cup T)^*$

where $G \in V$
$V, T \in$ Variable & Terminal.

e.g. - $S \rightarrow Xa$
$X \rightarrow a$
$X \rightarrow aX$
$X \rightarrow abc$
$X \rightarrow \epsilon$

(c) Type - 1 grammar

↳ Type - 1 grammars generate context sensitive languages.

↳ The productions must be in the form

$\alpha A \beta \rightarrow \alpha \beta$

where $A \in N$ (Non terminals)
$\alpha, \beta \in (T \cup N)^*$
T → Terminal, N → Non-terminal

DATE : / /

PAGE NO. :

e.g. — $AB \rightarrow AbBC$
$A \rightarrow bcA$
$B \rightarrow b$

(d) Type-0 grammar

↳ Type-0 grammars generate recursively enumerable languages. The productions have no restrictions. There are any phase structure grammar including all formal grammars.

↳ They generate the languages that are recognized by a Turing machine.

↳ The productions can be in the form of $\alpha \rightarrow \beta$

where α is a string of terminal & nonterminal with at least one nonterminal & α cannot be null

β is a string of terminals & non-terminals.

example

$S \rightarrow ACaB$
$Bc \rightarrow acB$
$CB \rightarrow DB$
$aD \rightarrow Db$