classmate
Date 04/10/23
Page

Assignment NO.2

Q.1. Explain chomskeys classification of grammar.

→ chomskey has, classified formal grammar in 4 ~~ch~~ categories

i) Type 3 grammar (T_3)
ii) Type 2 grammar (T_2)
iii) Type 1 grammar (T_1)
iv) Type 0 grammar (T_0)

i) Type 3 grammar (T_3) -

It is also called as regular grammar.

If every production of a grammar is in the form $A \longrightarrow \alpha B / B$

where $A, B \in V$

& $\beta, \alpha \in \Sigma^*$

then such grammar known as regular grammar.

The language generated by regular grammar is a regular language. or type 3 language which is further processed by finite automata.

Regular grammars are further classified into two types :

① RLG (Right linear grammar)
② LLG (Left linear grammar)

Every LLG can be converted into RLG. & every RLG can be converted into LLG.

∴ Their expressive power is identi-

cal.

For example, consider following grammar

$$A \longrightarrow Ba$$
$$A \longrightarrow \epsilon$$

② Type 2 grammar (context free grammar)

A grammar 'G' is context free grammar if & only if all the productions of that grammar are in the form

$$A \longrightarrow \alpha$$

where $A \in$ variable
& $\alpha \in (V+T)^*$

Context free grammar generate context free language. of type 2 language.

which is further processed by push down automata i.e. PDA.

push-down automata is a subform of parsor. Following are some of the IMP properties of CFG.

① Context free grammar ambigous or unambigous.

② Context free grammar either deterministic or non-deterministic.

- ~~context free grammar~~ Deterministic grammar generate deterministic CFL (DCFL), which is further processed by deterministic push down automata.
- lly, the language generated by non-deterministic CFG is known as NCFL.
- which is further processed by non-deterministic push down automata.
- Context free grammar are left recursive or right recursive.

③ Type 1 grammar-

This ~~is~~ type of grammar is also known as context sensitive grammar.

If all the productions of the grammar is in the form $x \longrightarrow y$.

where $x \in (V+T)^*$ & $y \in (V+T)^*$ and more over length of x length of $x \leq y$, then this grammar is known as context sensitive grammar (CSG).

one fact can be worth noted regarding CSG that CSG is non contracting grammar.

In the sence, the length of head is atmost equal to length of body.

④ Type 0 grammar-

It is also called as unrestricted grammar.

This type of grammar is supp

osed to generate recursive langua. ge as well as recursive innum. arable.

Both of recursive as well as recursive innumarable is commonly called as type 0 language further processed by tunning machine.

2. Write a grammar for
1. $L = \{ a^m b^n \mid m < n \}$

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

→ 1. $L = \{a^m b^n \mid m < n\}$

$m < n$

$m + p = n$

Rewritte the grammar

$L = \{ a^{m+p} b^{m+p} \mid m \geq 0, p \geq 1 \}$

$= \{ a^m b^m . b^p \mid m \geq 0, p \geq 1 \}$

$S \longrightarrow Sm . Sp$

$Sm \longrightarrow aSm . b \mid \epsilon$

$Sp \longrightarrow bSp \mid b$

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

$m > n$

$m = n + q$

Rewrite the grammar

$L = \{ a^n b^n \mid n \geq 0, q \geq 1 \}$

$L = \{ a^n . a^q . b^n \mid n \geq 0, q \geq 1 \}$

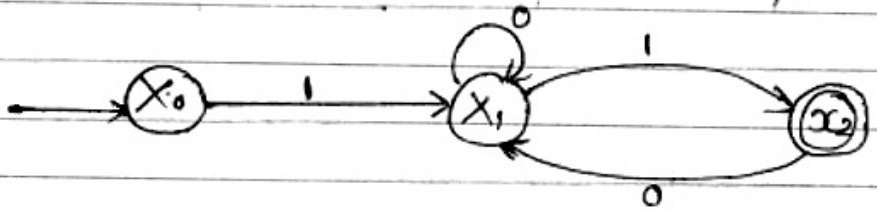
$= \{ a^q a^n b^n \mid n \geq 0, q \geq 1 \}$

$S \longrightarrow SqSn$

$Sq \longrightarrow aSq \mid a$

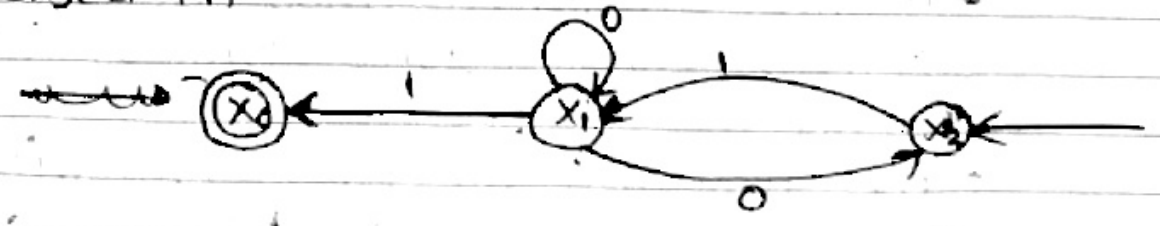
$Sn \longrightarrow aSnb \mid \epsilon$

3. obtain left linear grammar for language accepted by following FA.



→ step 1 - 1st fall, we shall reverse given FA.
Remainber to reverse the FA, swap the initial & final state only & after wards swap the direction of all transition.

Reversed FA -



From this reversed FA, we can write RLG as follows

$X_2 \longrightarrow X_1$

$X_1 \longrightarrow 0X_1 \mid 0X_2 \mid 1X_0$

$X_0 \longrightarrow \epsilon$

upon eliminating the ϵ production grammar becomes

$X_2 \longrightarrow 1X_1$

$X_1 \longrightarrow 0X_1 | 0X_2 | 1X_0 | 1$

Now, to obtain the LLG, we can reverse all the productions of that row. So, the final LLG be,

$X_2 \longrightarrow X_1 1$

$X_1 \longrightarrow X_1 0 | X_2 0 | X_0 1 | 1$

Q.4) Explain the concept of ambigious grammar with example.

→ - Ambiguity is one kind of impurity.
- If a grammar is contain in ambiguity then paesor cannt understand a grammar.
- such kinds of grammars cann't be processed by the system.
- ∴ The grammar must be clean & fine system can easily process of that grammar.
- By removing all the ambiguity of the grammar, grammar become clean & fine, so, that system can easily processit.

Concept of Ambigous grammar -

If a grammar has more than one LMD or more then RMD to derive same I/p string, then such grammar is known as ambigous grammar.

- Ambigous grammar is always unwanted. since, system cannot accept that grammar infact system cann't understand that grammar, hence system cann't process the grammar.

Consider the grammar as -

$E \longrightarrow E+E$

$E \longrightarrow E*E$

$E \longrightarrow id$

with respect to this grammer $V = \{E\}$, $\Sigma = \{id, +, *\}$, $S = \{E\}$, $P = \{E+E, E*E, (id)\}$

- Now, suppose the I/p string id + id * id is derived from this grammer.
- The grammar has more than one LMD as well as more than one RMD to derive the same I/p string id + id * id. This all derivations are expressed below.

LMD 1 : $E \Longrightarrow E+E$

$\Longrightarrow id+E$

$\Longrightarrow id+E*E$

$\Longrightarrow id+id*E$

$\Longrightarrow id+id*Eid$

LMD 2 : $E \Rightarrow E * E$
$\Rightarrow E + E * E$
$\Rightarrow id * E * E$
$\Rightarrow id + id * E$
$\Rightarrow id + id * E$
$\Rightarrow id + id * id$

RMD 1 : $E \Rightarrow E + E$
$\Rightarrow E + E * E$
$\Rightarrow E + E * id$
$\Rightarrow E + id * id$
$\Rightarrow id + id * id$

RMD 2 : $E \Rightarrow E * E$
$\Rightarrow E * E$
$\Rightarrow E + e * id$
$\Rightarrow E + id * id$
$\Rightarrow id + id * id$

As this grammar having multiple LMD or RMD, the grammar is ambigous.

19/10.