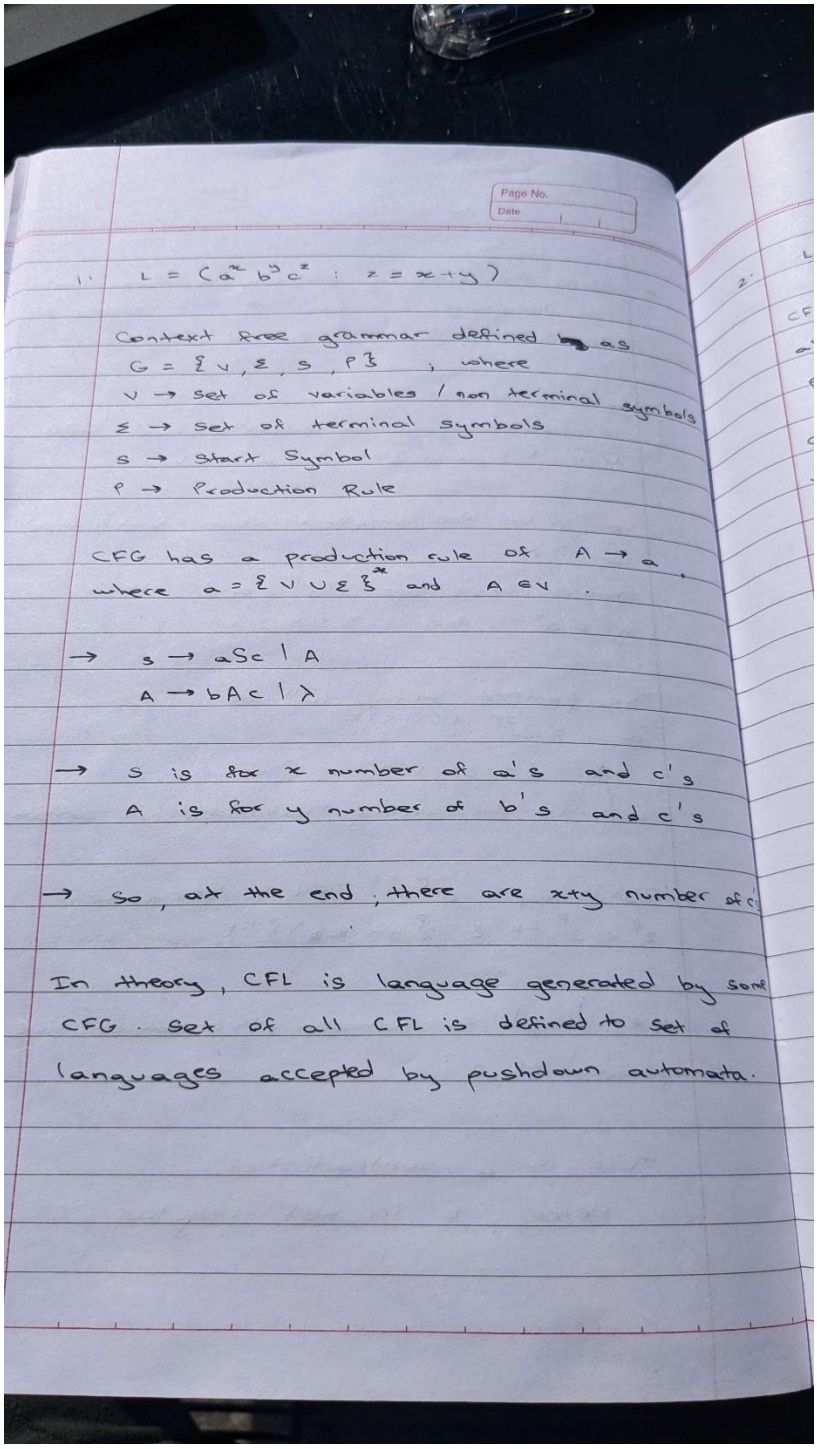
1. Create a context-free grammar for the language L = ($a^x b^y c^z$: z = x + y}.



1. $L = (a^x b^y c^z : z = x+y)$

Context free grammar defined as
$G = \{V, \Sigma, S, P\}$; where
V → set of variables / non terminal symbols
Σ → set of terminal symbols
S → Start Symbol
P → Production Rule

CFG has a production rule of $A \rightarrow a$, where $a = \{V \cup \Sigma\}^*$ and $A \in V$.

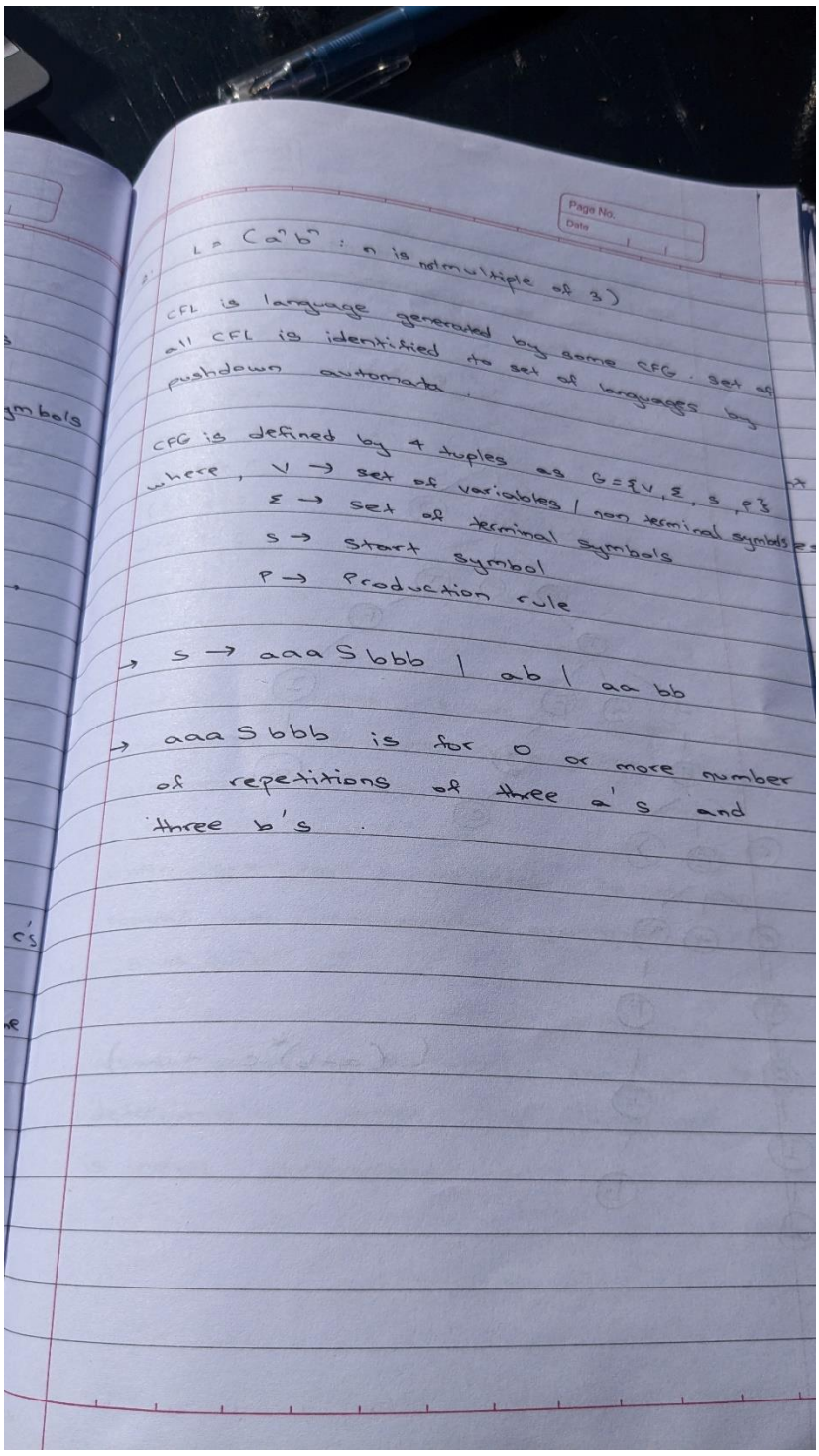
→ $S \rightarrow aSc \mid A$
$A \rightarrow bAc \mid \lambda$

→ S is for x number of a's and c's
A is for y number of b's and c's

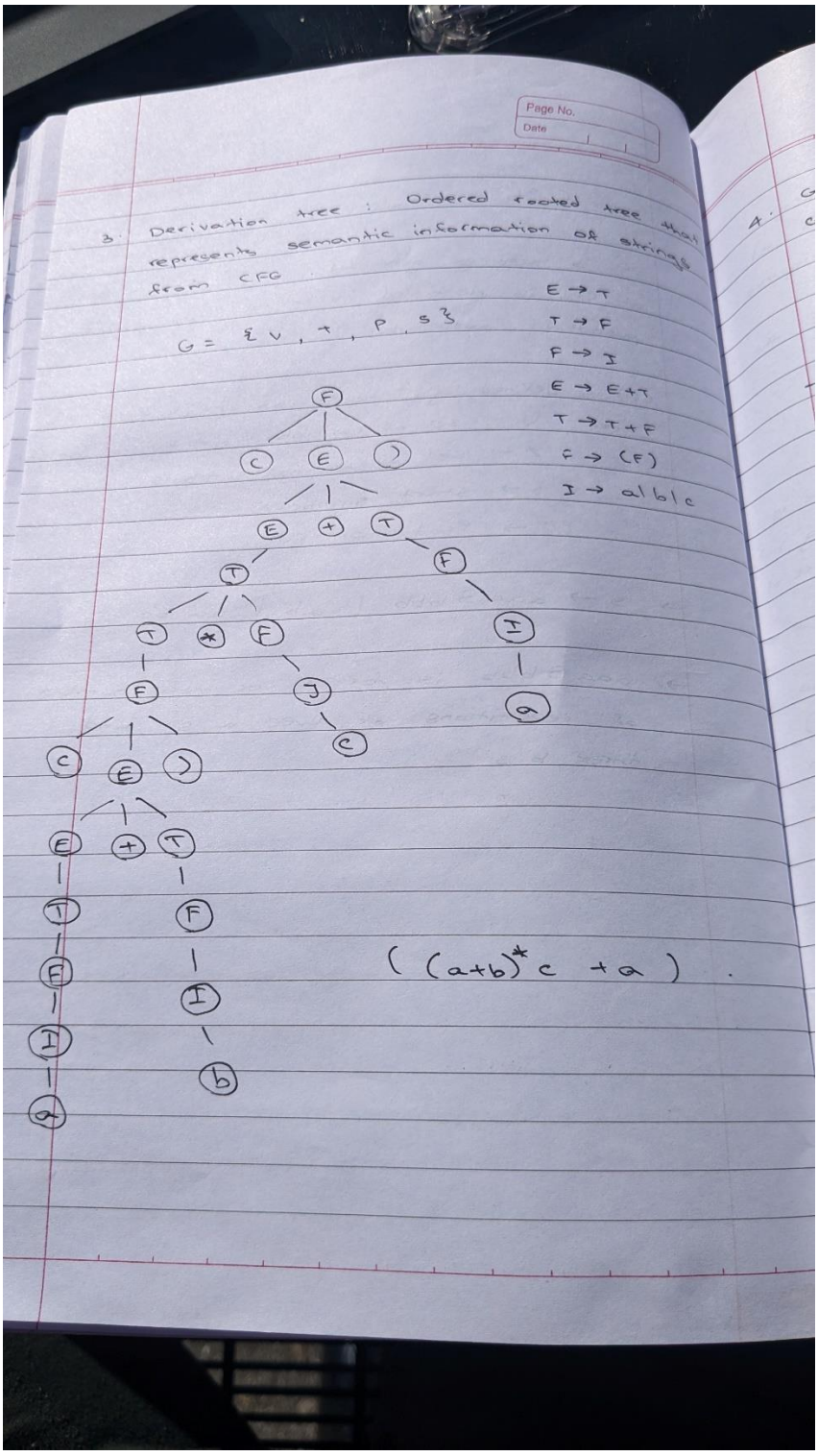
→ So, at the end, there are x+y number of c's

In theory, CFL is language generated by some CFG. Set of all CFL is defined to set of languages accepted by pushdown automata.

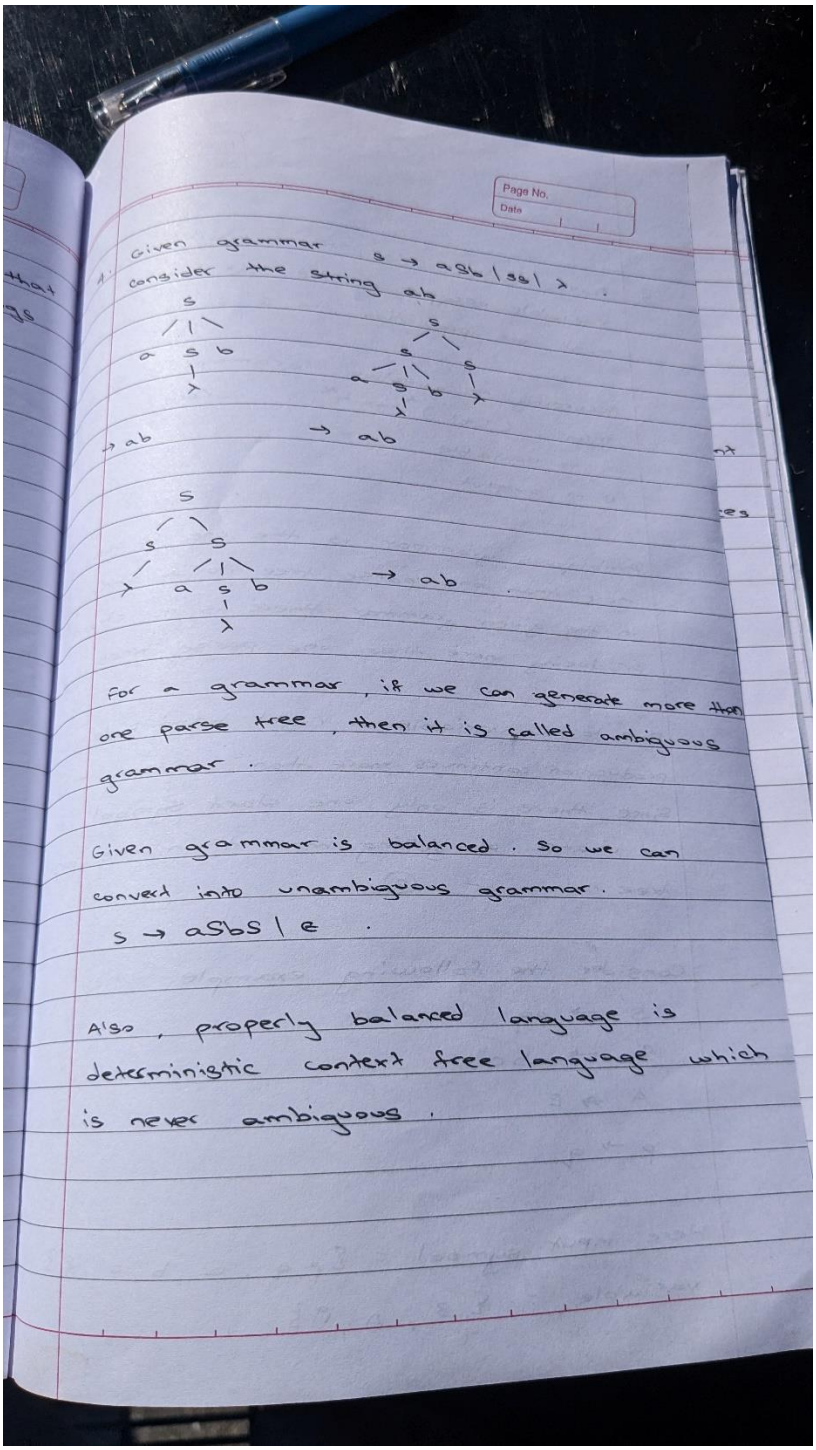
2. Create a context-free grammar for the language L = ($a^n b^n$: n is not a multiple of 3}.



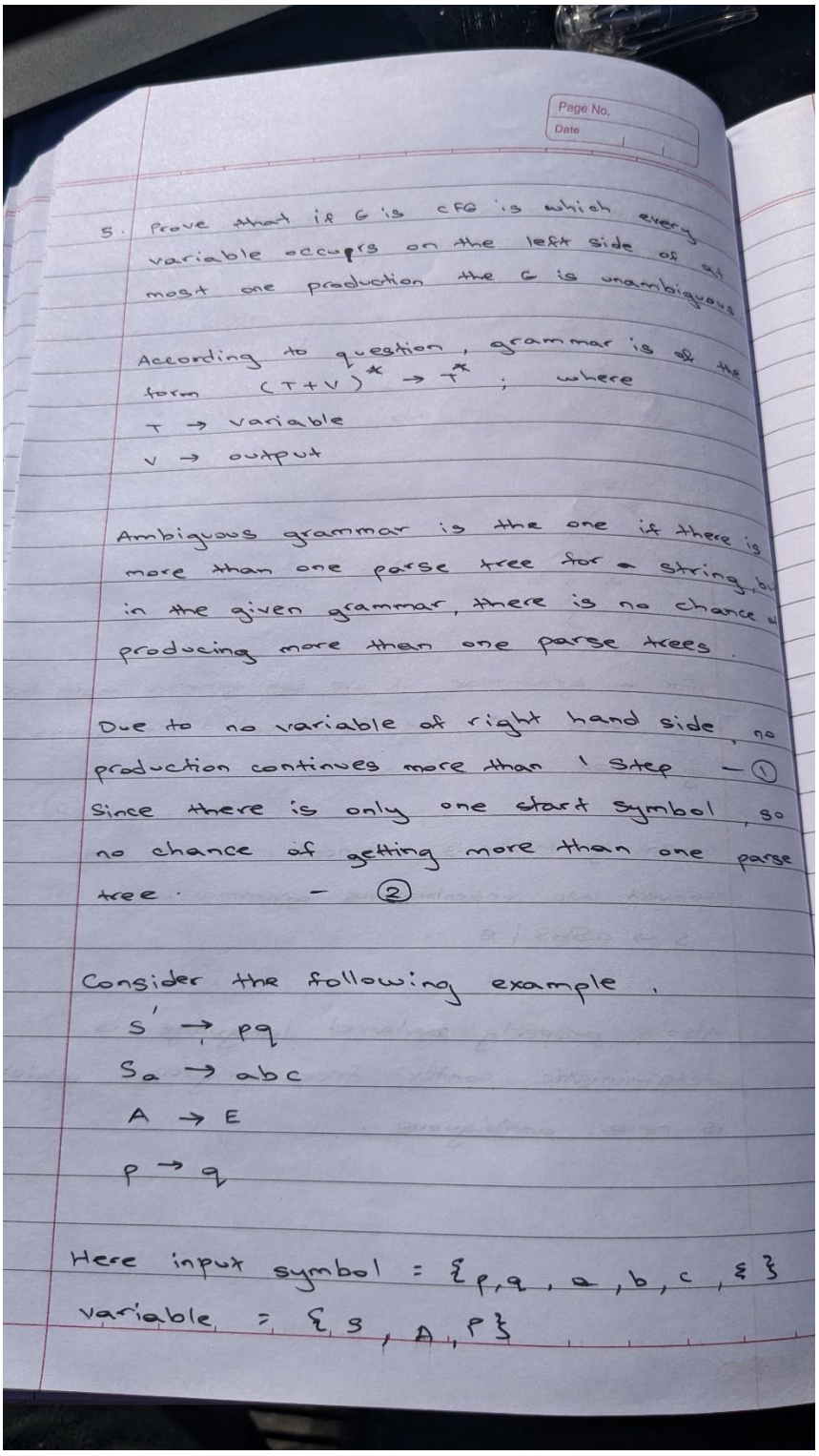
3. Give the derivation tree for (a + b) * c + a, using the grammar in Example 5.12.



4. Show that the following grammar is ambiguous, but that the language it generates is not inherently ambiguous.



5. Prove that if G is a context-free grammar in which every variable occurs on the left side of at most one production, then G is unambiguous.



Page No.
Date

5. Prove that if G is CFG in which every variable occurs on the left side of at most one production the G is unambiguous.

According to question, grammar is of the form $(T+V)^* \rightarrow T^*$; where

T → variable

V → output

Ambiguous grammar is the one if there is more than one parse tree for a string, but in the given grammar, there is no chance of producing more than one parse trees.

Due to no variable at right hand side, no production continues more than 1 step — ①

Since there is only one start symbol, so no chance of getting more than one parse tree. — ②

Consider the following example,

$S' \rightarrow pq$

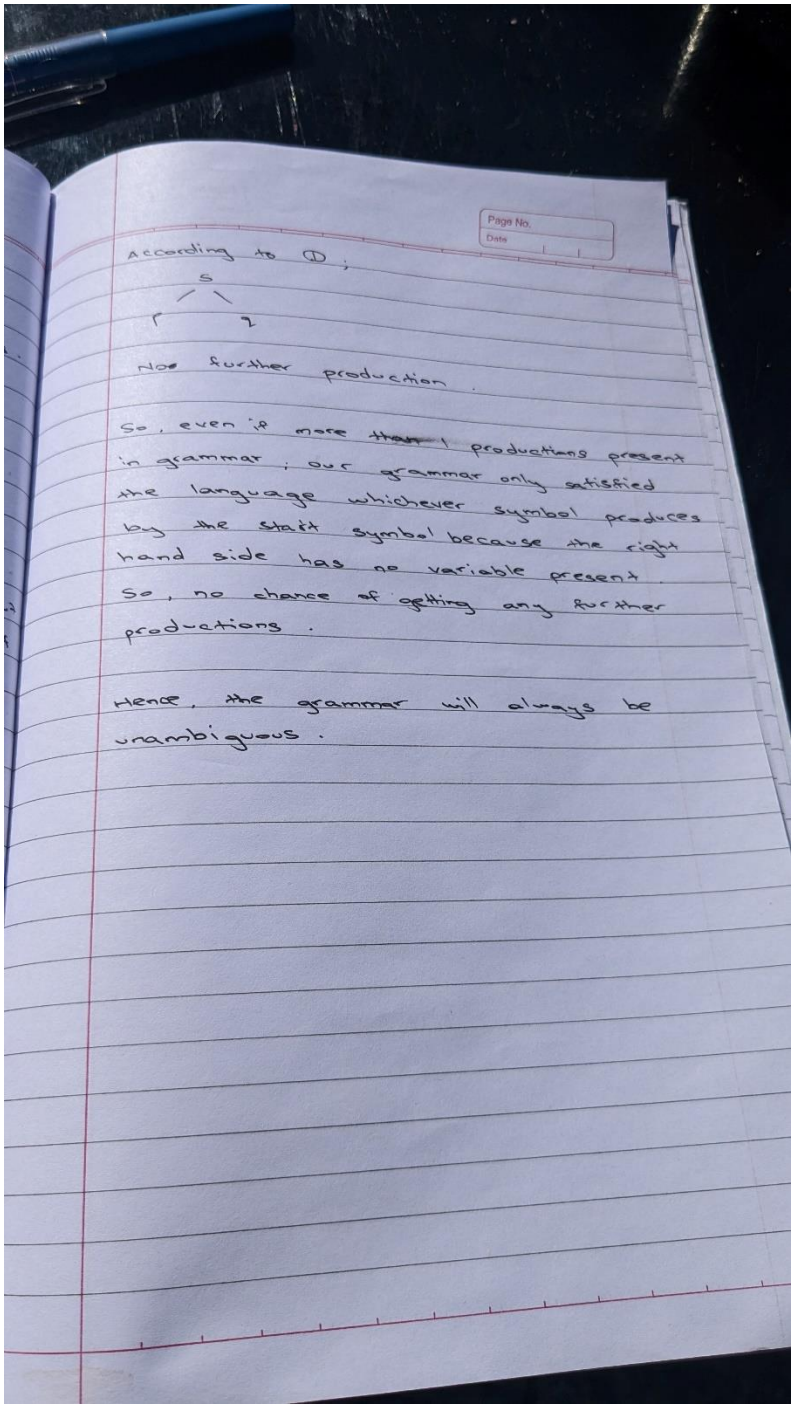
$S_a \rightarrow abc$

$A \rightarrow E$

$p \rightarrow q$

Here input symbol = {p, q, a, b, c, ε}

variable = {S, A, P}


Page No.
Date
According to D;
S
r
2
No further production.
So, even if more than 1 productions present in grammar, our grammar only satisfied the language whichever symbol produces by the start symbol because the right hand side has no variable present. So, no chance of getting any further productions.
Hence, the grammar will always be unambiguous.