Ambiguity in Grammar

A grammar is said to be ambiguous if there exists,

* more than one left most derivation (or)
* more than one right most derivation (or)
* more than one parse tree for given i/p string.

→ If the grammar is not ambiguous, then we call unambigious.

→ If the grammar has ambiguous, then it is not good for compiler construction.

→ no method can automatically detect and remove the ambiguity, but we can remove ambiguity by re-writing the whole grammar without ambiguity.

Example:

Let us consider the grammar with production rule

$E \rightarrow I$

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$I \rightarrow \epsilon \mid 0 \mid 1 \mid 2 \mid \ldots 9.$

Dr. I. Jasmine

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

$V = \{I, E\}$

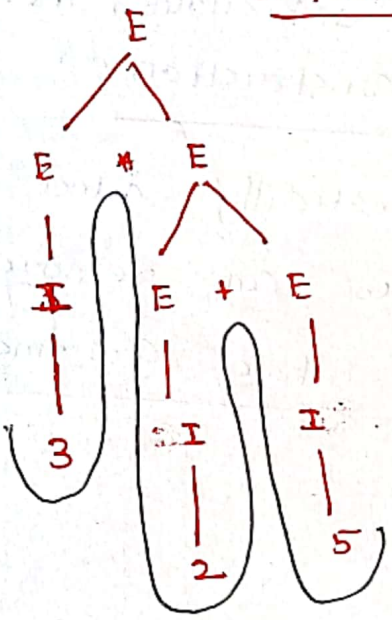
$T = +, *, (,), e, 0, 1, \ldots 9\}$

$S = E$

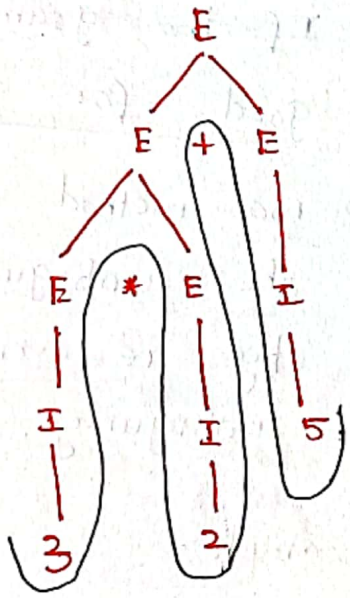
Solution:

For the String "3 * 2 + 5", the above grammar can generate two parse tree by leftmost derivation.

LM Tree 1:



LM Tree 2:



Since there are two parse tree for single String the grammar is ambiguous.

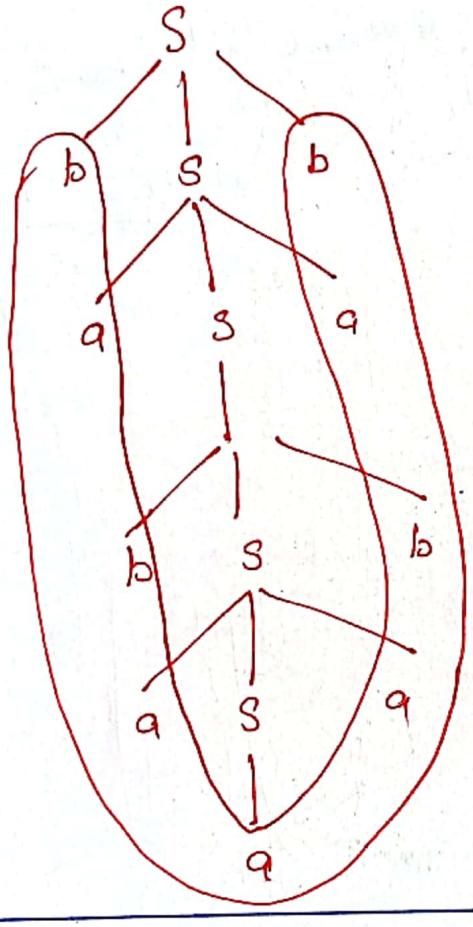
Dr. I. Jasmine

1. Draw a derivation tree for the String babaaabab for the CFG given by G

where.

$$P = \{ S \to aSa$$
$$S \to bSb$$
$$S \to a \mid b \mid \epsilon \}$$

2.



2. For the grammar G defined by the production

$$S \to A \mid B \mid A1B$$
$$A \to 0A \mid \epsilon$$
$$B \to 0B \mid 1B \mid \epsilon$$

Find the parse tree for the yields

(i) 1001 (ii) 0001 (iii) 00011

Dr. I. Jasmine