Tutorial-07

U19CS009

Page No.:
Date: / /

Q1 $S \rightarrow aAb \mid \varepsilon$
$A \rightarrow aAb \mid \varepsilon$

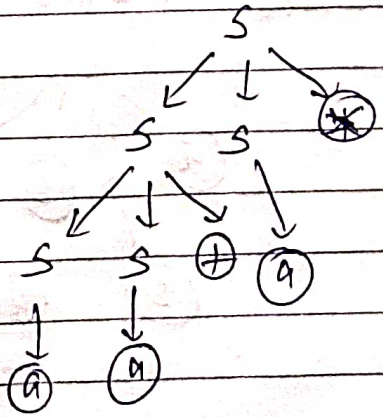
for the given grammar find out terminates and start symbols

→ Terminator → $\{a, b, \varepsilon\}$
non terminal → $\{S, A\}$
start with $\{S\}$

Q-2 Consider the context free grammar
$S \rightarrow SS+ \mid SS* \mid a$
show that string $aa+a*$ can be generator by grammar.

$S \rightarrow SS*$
$S \rightarrow SS+S*$
$S \rightarrow aS+S*$
$S \rightarrow aa+S*$
$S \rightarrow aa+a*$

Q-3 (b) Construct parse tree for the string



Page No.:
Date: / /

(c) What language does this grammar generate? Justify

→ It will generate post fix expression of addition and multiplication.

Q-3 What language is generated by following grammar? In each case justify your ans. Also check which of the grammar is ambiguous

a) $S \rightarrow 0S1 \mid 01$

$L = \{0^n 1^n \mid n >= 1\}$

↳ not ambiguous

b) $S \rightarrow +SS \mid -SS \mid a$

$L = \{$ prefix expression consisting of addition and subtraction operations $\}$

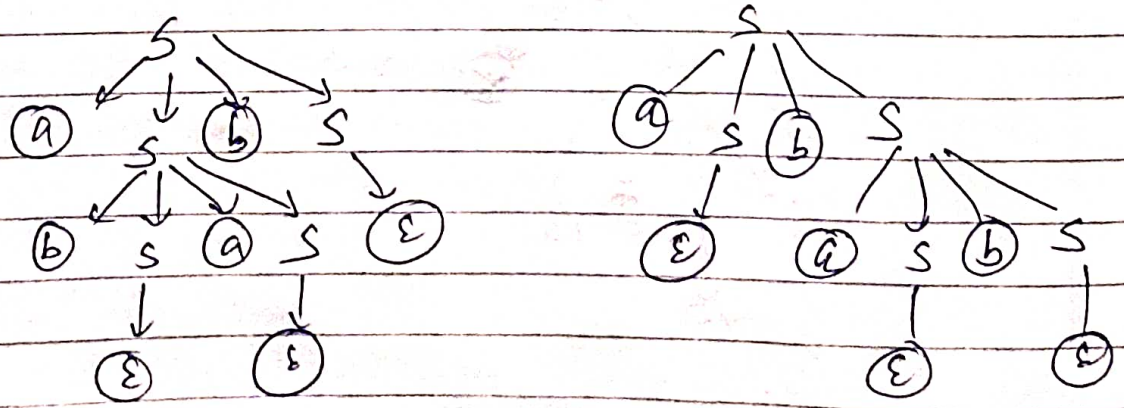
↳ not ambiguous

c) $S \rightarrow aSbS \mid bSaS \mid \varepsilon$

$L = \{$ string with equal no. of a & b $\}$

↳ Ambiguous

say we want to generate abab

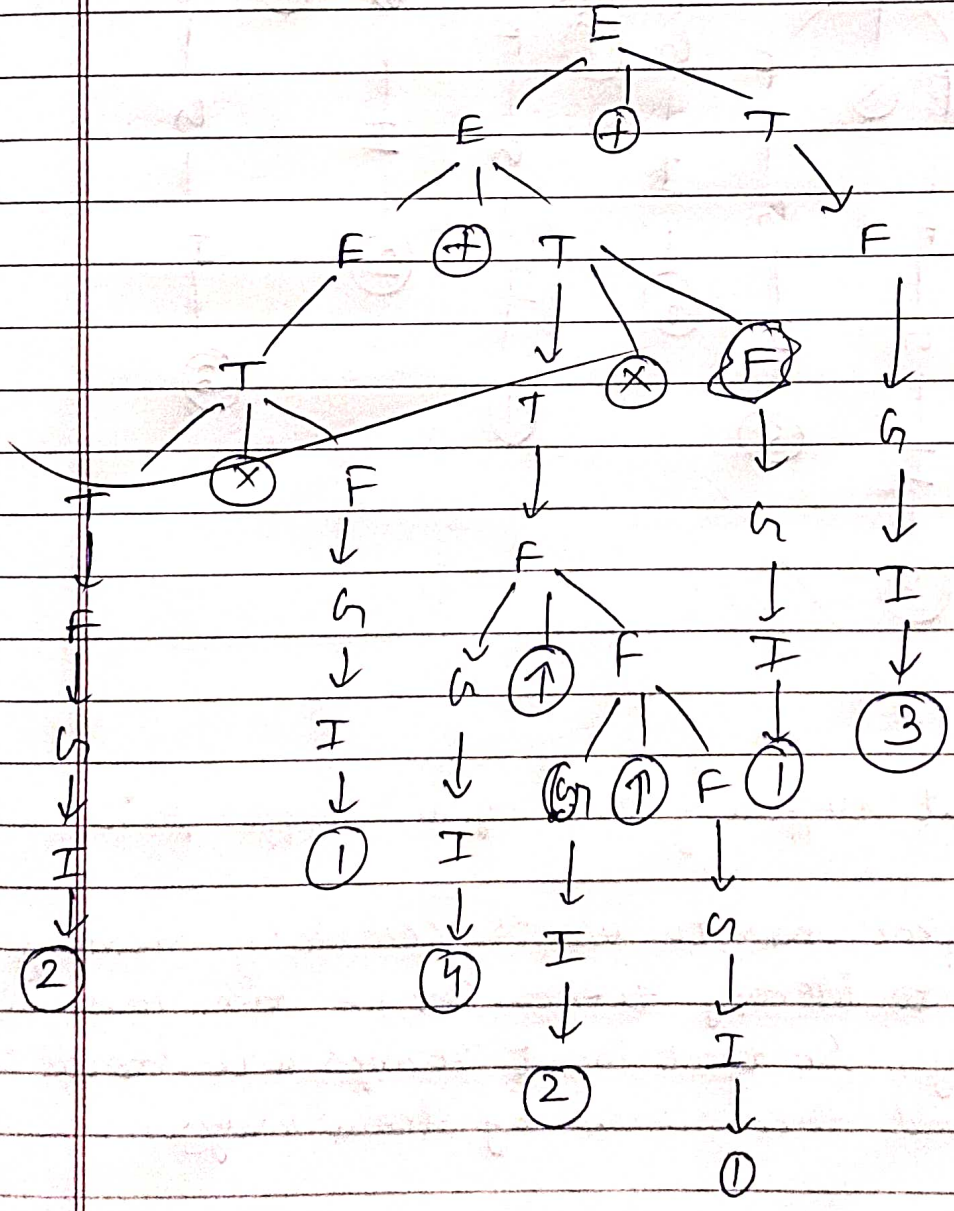


→ We are able to generate two parse trees for same string.

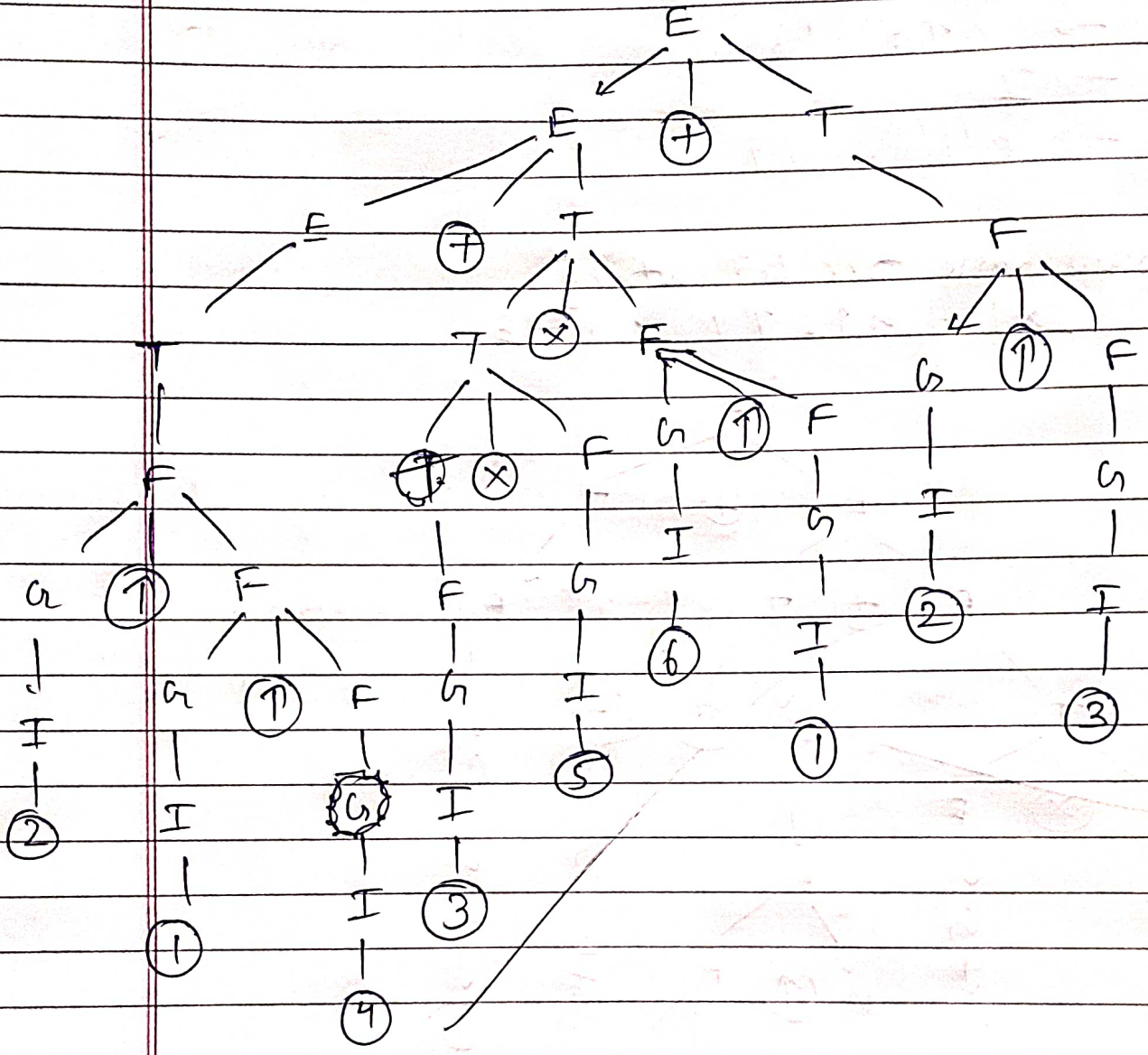
Page No.:
Date : / /

Q4 $E \rightarrow E+T \mid E-T \mid T$
$T \rightarrow T \times F \mid T \% F \mid F$
$F \rightarrow G \uparrow F \mid G$
$G \rightarrow I$
$I \rightarrow 0 \mid 1 \mid \ldots \mid 9$

-> Draw parse tree for string
a) $2 \times 1 + 4 \uparrow 2 \uparrow 1 \times 1 + 3$



b) $2 \uparrow 1 \uparrow 4 + 3 \times 5 \times 6 \uparrow 1 + 2 \uparrow 3$



Q-5 Define input buffering and explain buffer pairs

→ Logical analyzer has to access secondary memory each time to identify tokens. It is time-consuming and costly, so that input strings are stored into a buffer and then scanned by lexical analyzer.

→ Lexical analyzer scans input string from left to right one character at a time to identify tokens. It uses two tokens to scan ① Begin pointer and ② look-ahead pointer

Page No.:
Date : / /

Buffer pair → a specialize buffering technique can decrease the amount of overhead, which is needed to process as Input characters in transferring charaters. It includes two buffers, each includes N-charater size which is reloaded alternatively.