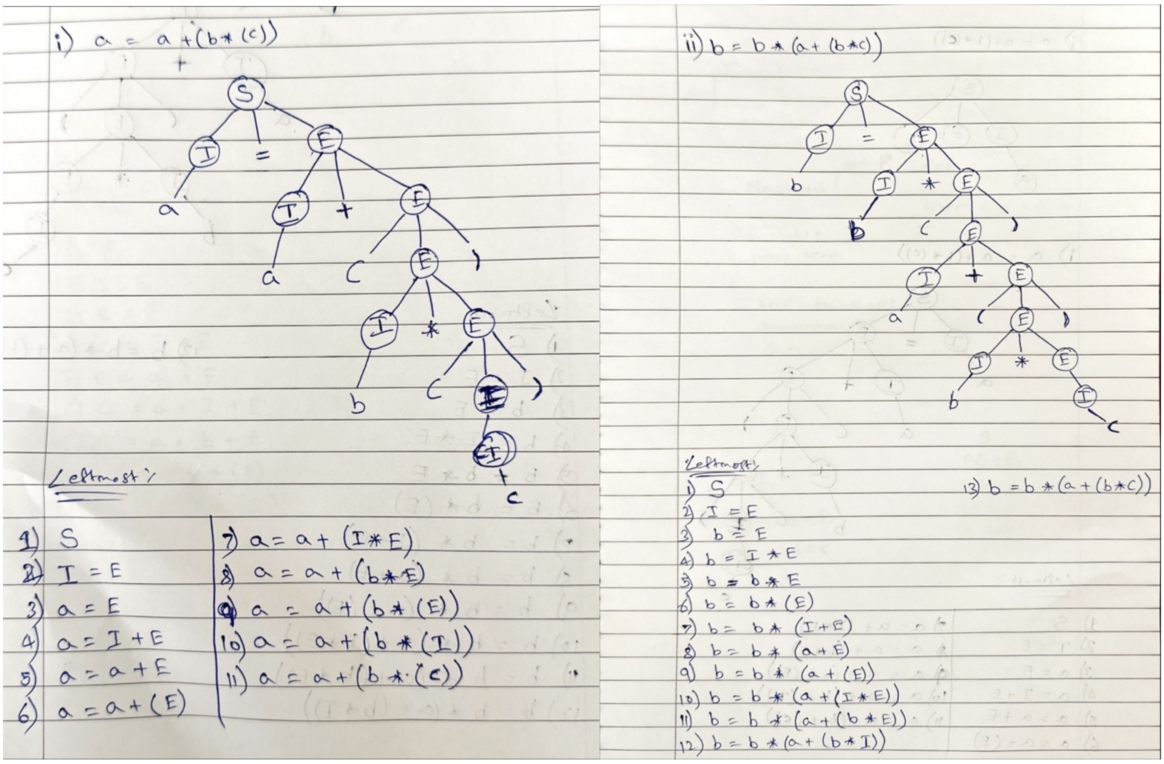
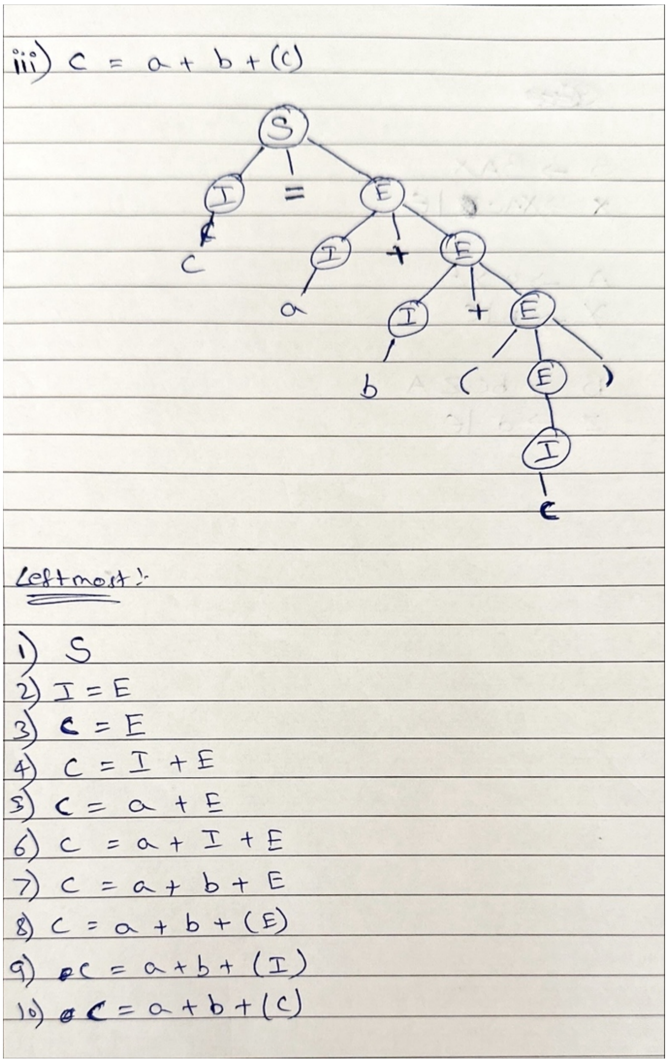
CMSC 331: Principles of Programming Language
Homework 1 Solutions

1. Given the grammar:
   S → I = E
   I → a | b | c
   E → I + E | I * E | (E) | I

   Draw a parse tree and leftmost derivation for the following sentences:
   I. a = a + (b * (c))
   II. b = b * (a + (b * c))
   III. c = a + b +(c)

Sol:





2. Demonstrate that the following grammar is ambiguous:
   S → I
   I → E + E | (E) | E * E | I
   E -> a-z | 0-9

   Here are two different left-most derivation of the same string:

   S → I → E * E → E + E * E → a + b + c
   S → I → E + E → E + E * E → a + b + c

3. Determine which of the sentences (a-e) are in this grammar:
   S → aScB | A | b
   A → cA | c
   B → d | A

a. abcbd not in grammar
S can derive a word that starts with a (aScB) or c (A) or b;
must use first prod; first S must go S → b, so we have abcB
but, “B” only goes to strings that are just b or start with c
b. abababc   “b” can only be produced from an S; S is always followed by “c”;   therefore “ba” can’t be there.
c. cccc   S → A → cA → ccA → cccA → cccc
d. accd S → aScB → aAcB → accB → accd
e. aabcdcc   S → aScB → aaScBcB → aabcBcB → aabcdcB
→ aabcdcA → aabcdcc

4. Convert the following basic EBNF rules to the basic BNF notation:
S → BA{Ac}
A → a[b]A
B → bc[d]A

Sol: S → BAX
X → XAc | Ac
A → aYA
Y → b | bY
B → bcZA
Z → d | dZ