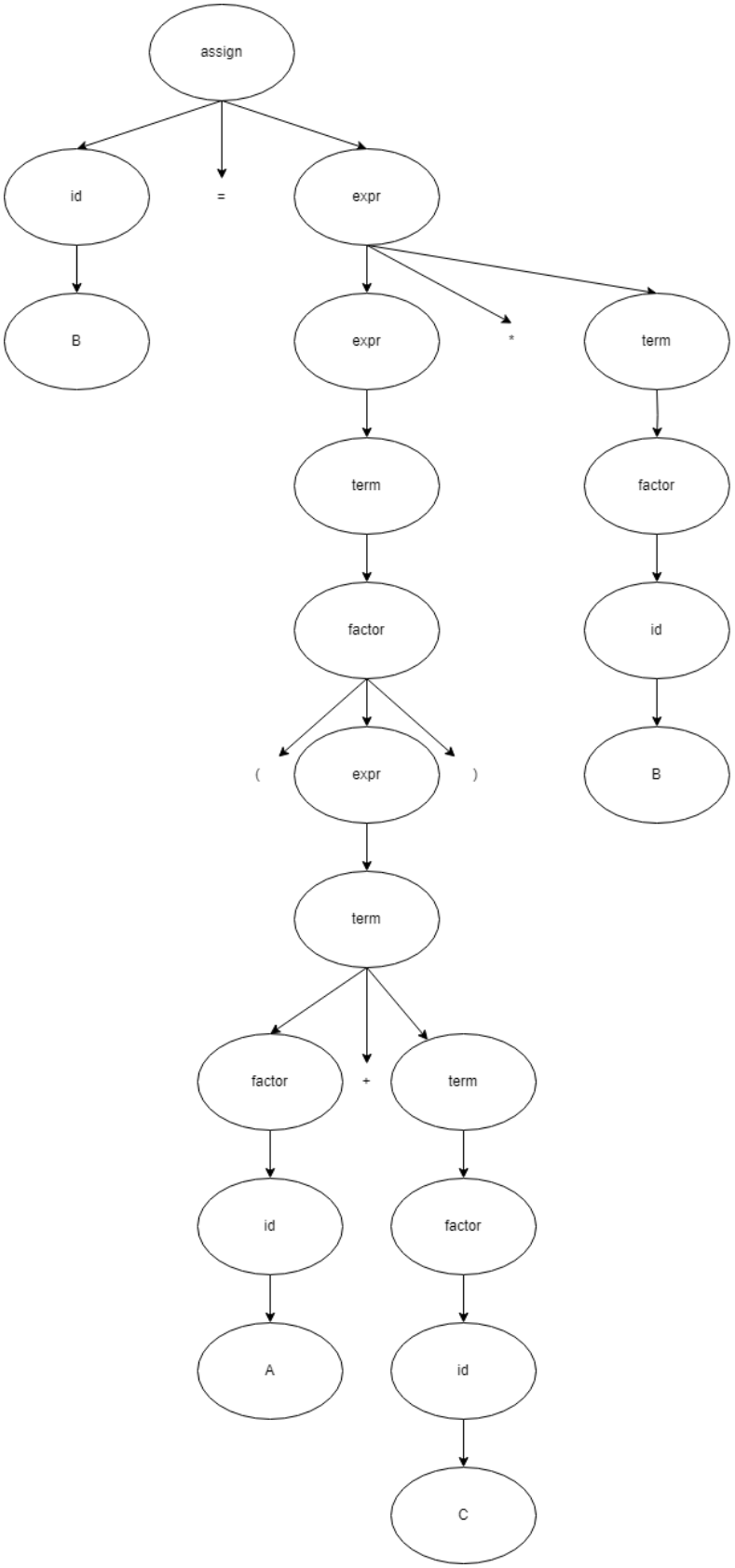
CS 381 Homework 3 – Syntax

1. Using the grammar below, show a parse tree and a leftmost derivation for the sentence



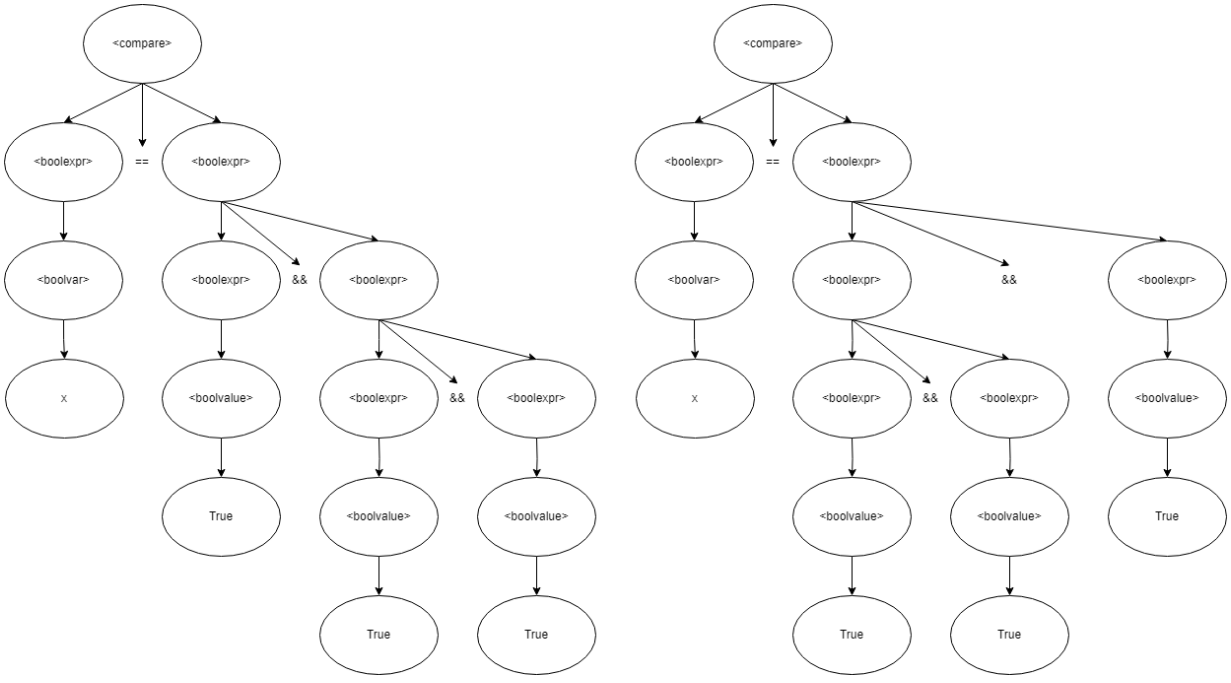
<assign> -> <id> = <expr>
<assign> -> B = <expr>
<assign> -> B = <expr> * <term>
<assign> -> B = <term> * <term>
<assign> -> B = <factor> * <term>
<assign> -> B = ( <expr> ) * <term>
<assign> -> B = ( <term> ) * <term>
<assign> -> B = ( <factor> + <term> ) * <term>
<assign> -> B = ( <id> + <term> ) * <term>
<assign> -> B = ( A + <term> ) * <term>
<assign> -> B = ( A + <factor> ) * <term>
<assign> -> B = ( A + <id> ) * <term>
<assign> -> B = ( A + C ) * <term>
<assign> -> B = ( A + C ) * <factor>
<assign> -> B = ( A + C ) * <id>
<assign> -> B = ( A + C ) * B

2. Rewrite the following BNF to add the postfix ++ and -- unary operators of Java

<assign> -> <id> = <expr>
<expr> -> <expr> * <term> | <term>
<term> -> <factor> + <term> | <factor> - <term> | <factor>
<factor> -> ( <expr> ) | <id> ++ | <id> --
<id> -> A | B | C

3. Show that the following grammar is ambiguous. Note: The terminals symbols are in bold.

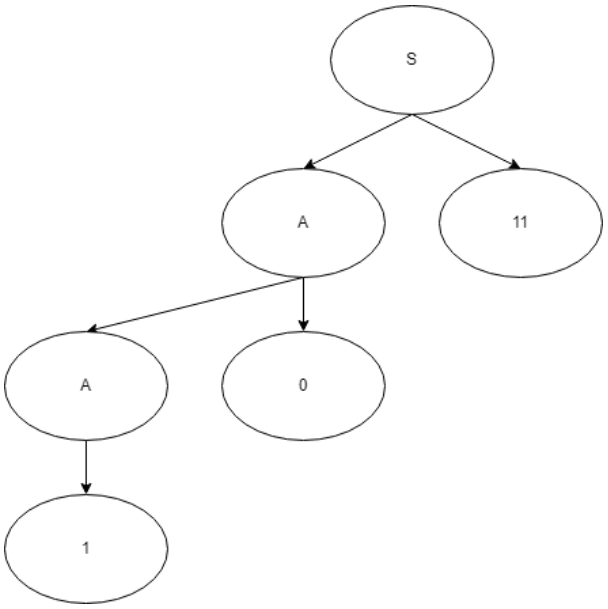
We observe two distinct trees to generate the sentential form **x = True && True && True**, thus showing that the grammar is ambiguous.



4. Write a grammar G for the language L consisting of strings of 0's and 1's that are the binary representation of odd integers greater than 4.

G = { N, Σ, P, S }
N = { S, A }
Σ = { 01, 11, 1, 0}
P = { (S, A01), (S, A11), (A, A1), (A, A0), (A, 1), (A, 0) }
P : S -> A01 | A11
A -> A1 | A0 | 1 | 0

Parse tree for 1011:




Parse tree for 1101: