Yuhan Xiao Homework 1

1. (a) True. Because a grammar is ambiguous if there exist at least two distinct parse trees for the derivation of the same string. These two parse trees may not change the meaning of this string.

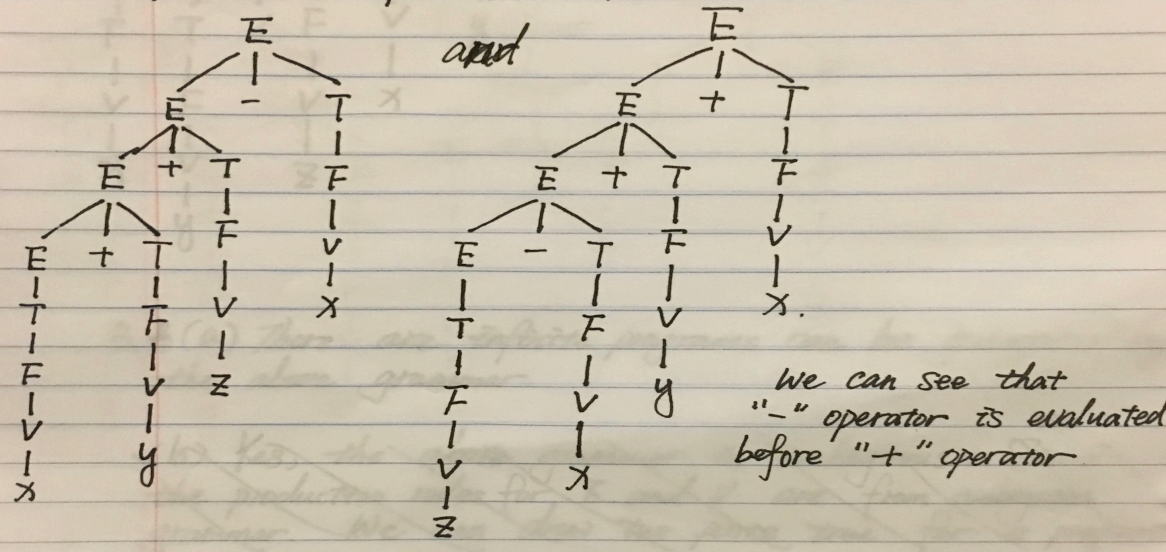
(b) True. Because if two parse trees for a correct string ~~empties~~ generated by this grammar have different meaning, they must be different trees. So this grammar is ambiguous.

(c) True. Because an unambiguous grammar can only have one pharse tree for derivation.

(d) True. Because I runned it and it printed 8 In Java, ++y evaluated before * 2.

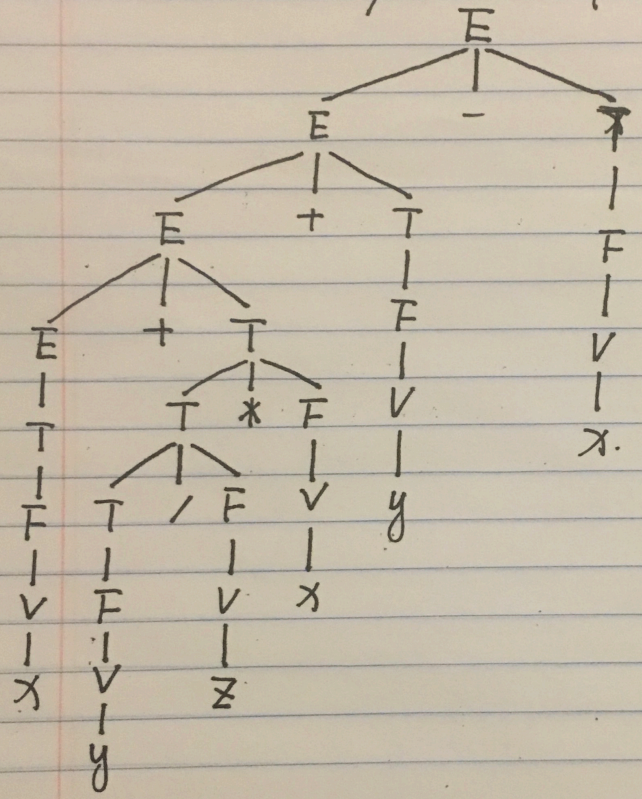
2. (a) "+" and "−" operators are both left associative. Because the head "E" appears at the left side of the operators

(b) No, we cannot ~~because it's an ambiguous grammar.~~ Proof by counter-example: x+y+z−x. and z−x+y+x. We can have parse Trees like:



we can see that "−" operator is evaluated before "+" operator

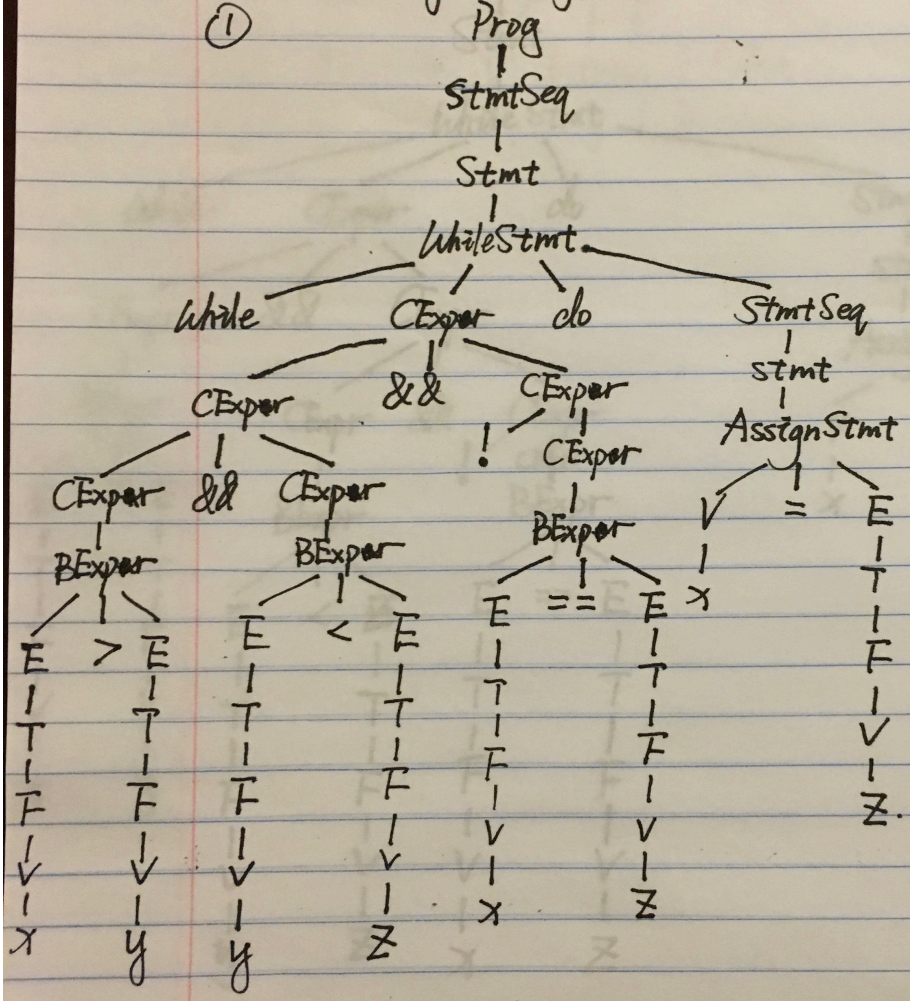
(c) Yes, x+y/z*x+y-x is correct for this grammar.
We can draw a parse tree for it:



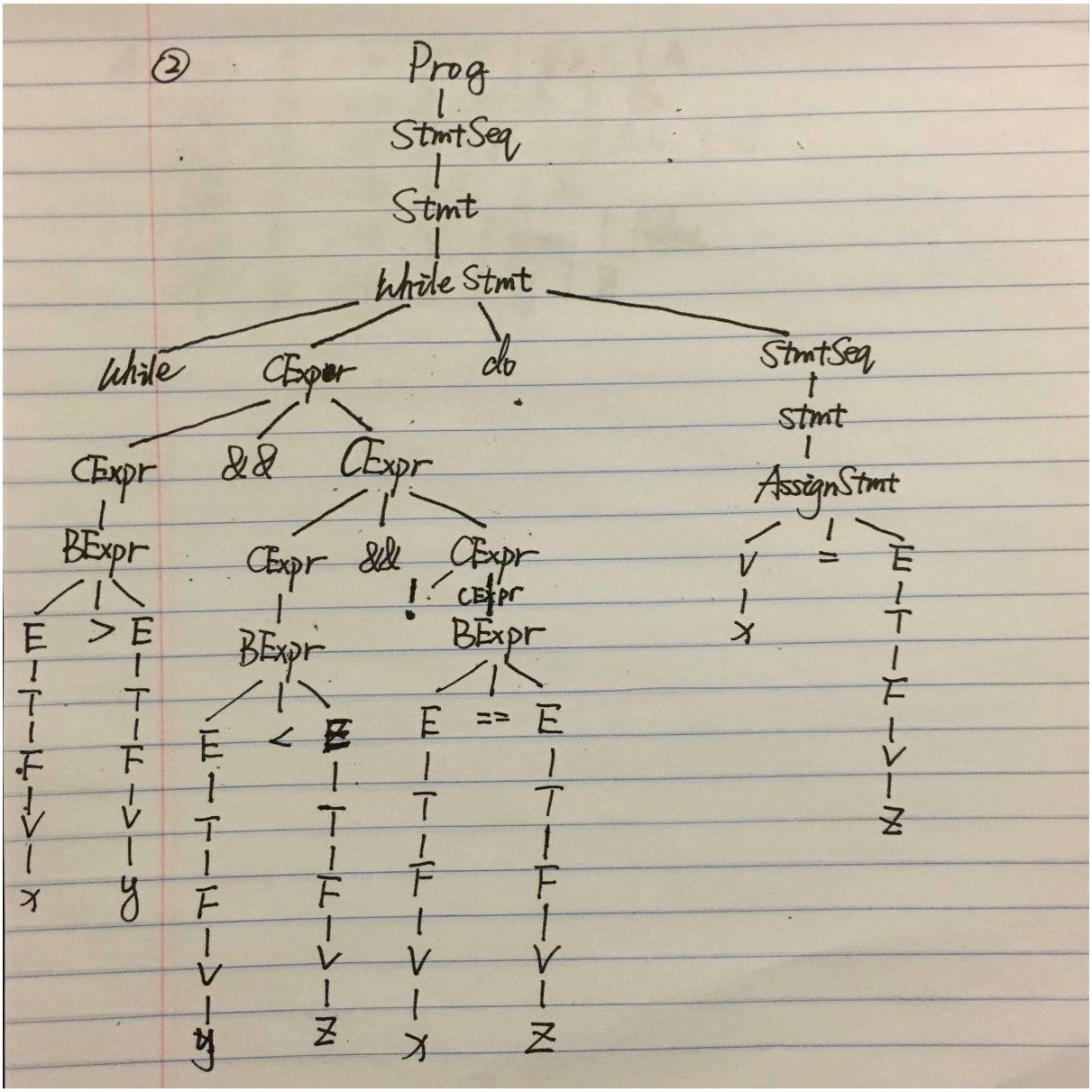
3. (a) There are infinite programs can be generated by the above grammar.

(b) Yes, the above grammar is ambiguous.
We can draw two pharse trees for this program:
while (x > y && y < z && ! x == z) do x = z

①



②



4. (a) $S \rightarrow \exists V.S \mid \forall V.S \mid A$
   (b) $A \rightarrow B \Rightarrow A \mid B$
   (c) $B \rightarrow B \vee C \mid B \wedge C \mid C$
   (d) $C \rightarrow \neg C \mid F$
   (e) $F \rightarrow V \mid \text{true} \mid \text{false}$
   (f) $V \rightarrow x \mid y \mid z$