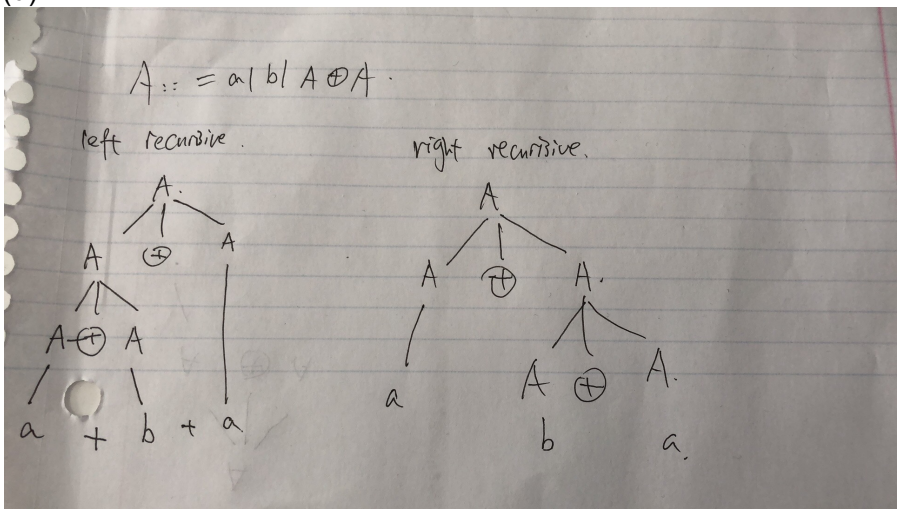
CSCI3315-Lab2 Writeup
Name: Lei Teng
Partner: Zijun Xu

2.
(a)
A, B are non-terminal
a, b, c are terminal
String created by the set of grammars can only be a, b, c, and empty.

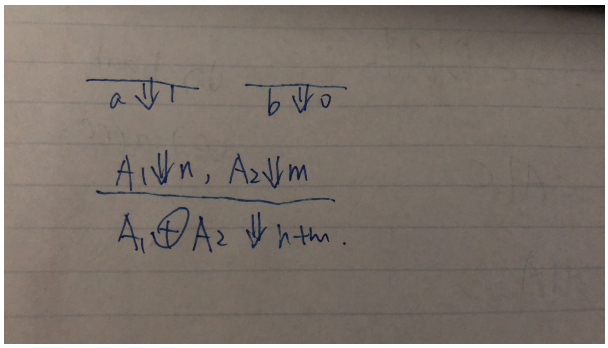
(b)
S => AaBb
=> baBb
=> baab
S => AaBb
=> AbaBb
=> bbaBb
=> bbaab

(c)
S => aScB|A|b
=> aScB
=> abcB
=> abcd
S => aScB|A|b
=> aScB
=> aAcB
=> accB
=> accA
=> accc
(d)



since it can be derived as two different trees, it is ambiguous

(e)



3.
(a)
The first grammar: begins with e. It can be derived as operand or e operator operand. If go operand, it would go to termination and if go e operator, operand is non-terminal.
The second grammar: esuffix can either be empty (ε) or operator operand esuffix.
Since both grammar have the choice to call themselves, there is only one same math operation in the expression, so whatever how many times it repeats, they are always the same.

(b)
In nodeJs:
3 << 0 = 3
3 << 1 - 1 = 3

val x1 = (3<<1)-1
val x2 = 3<<(1-1)
if (x1 == 3) printf("- has higher precedence") else ("<<has higher precedence)
So '-' has higher precedence than '<<'.

(c)
S ::= -A|A
A ::= B.C|B.CEB|B.CE-B
B ::= DC|D
C ::= Z|ZC
D ::= 1|2|3|4|5|6|7|8|9
Z ::= 0|D