Question 1

program: list EOF;

list: LP RP | LP list+ RP;

LP: ‘(’;
RP: ‘)’;

Question 2
grammar MC;

program: var_decl+ EOF;

var_decl: type identifier_lst SEMI;

type: ‘int’ | ‘float’ | ‘boolean’;
identifier_lst: Identifier (‘,’ Identifier)*;

Identifier: [a-zA-Z]+;
SEMI: ‘;’;

Question 3
grammar MC;

program: expr+ EOF;

expr: expr1 (‘^’) expr | expr1;
expr1: expr1 (‘+’|’-’) expr2 | expr2;
expr2: expr2 (‘*’|’/’) expr3 | expr3;
expr3: ’!’ expr3 | expr4;
expr4: ‘(‘ expr4 ’)’| Integer | Identifier | func_call | expr;

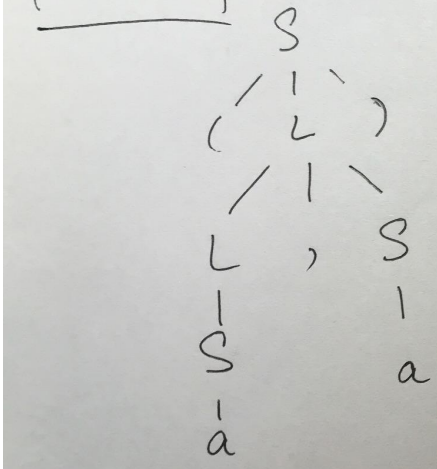
func_call: Identifier ‘(‘ expr_lst? ‘)’;
expr_lst: expr (‘,’ expr)*;

Integer: [1-9] [0-9]* | [0];
Identifier: [a-zA-Z]+;

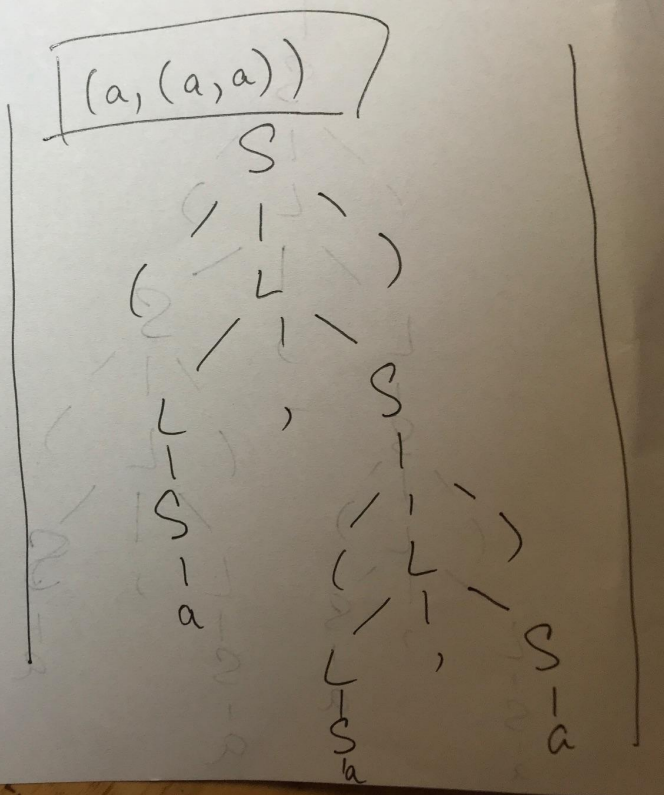
Question 4

a. Non-terminal set {S, L}
   Terminal set {a}
   Start symbol: S
b.
c. Answer

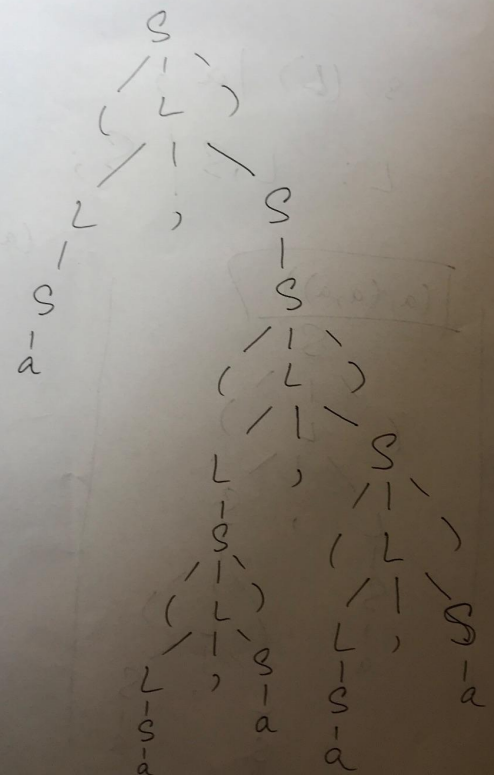
(a, a)



(a, (a, a))

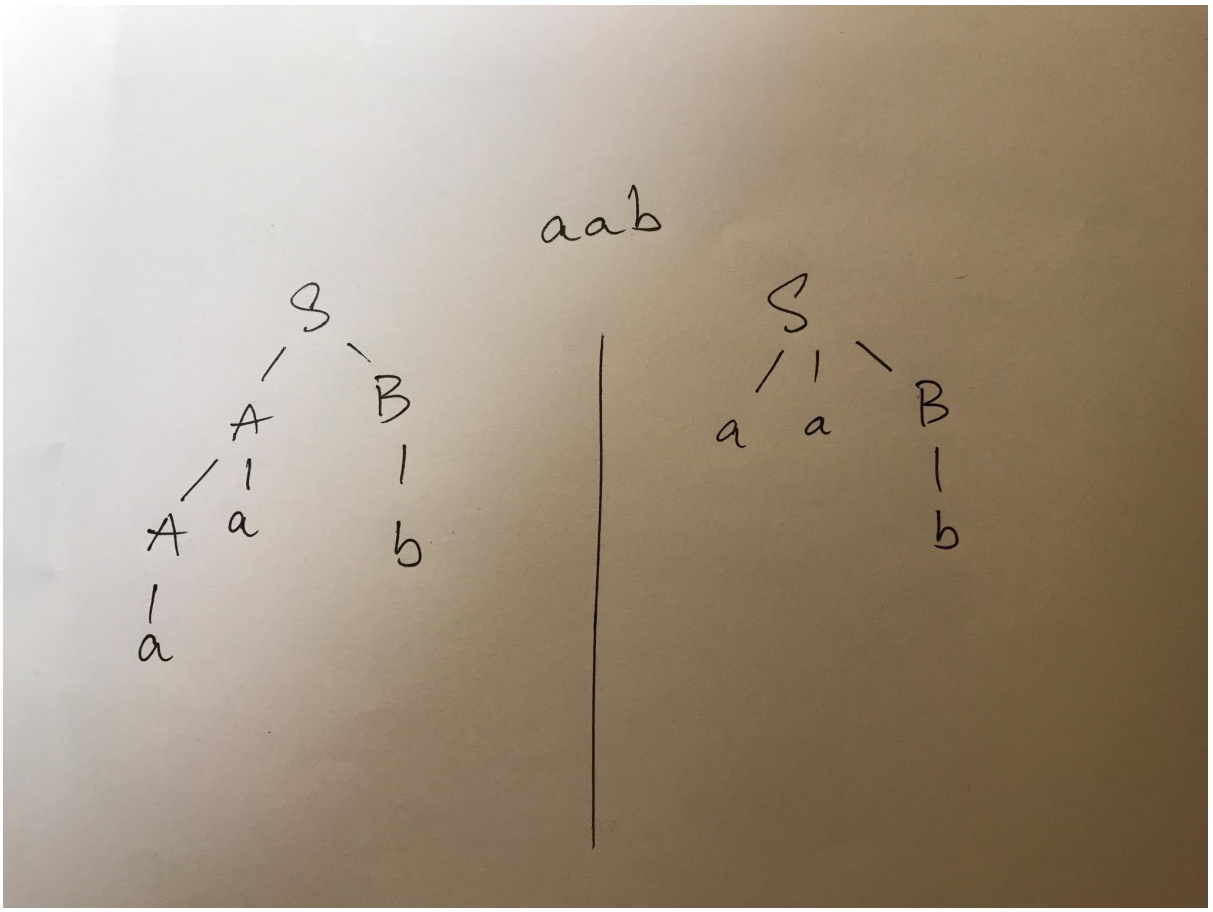


(a, ((a, a), (a, a)))



Question 5

a. For the string ‘aab’, we could generate 2 parse trees. Therefore, it is ambiguous.



b.