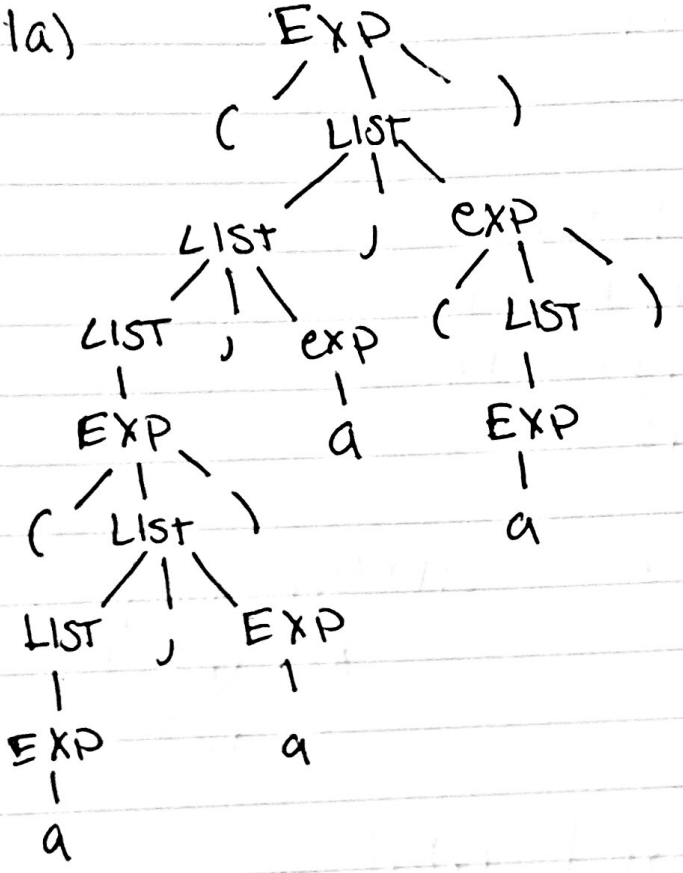
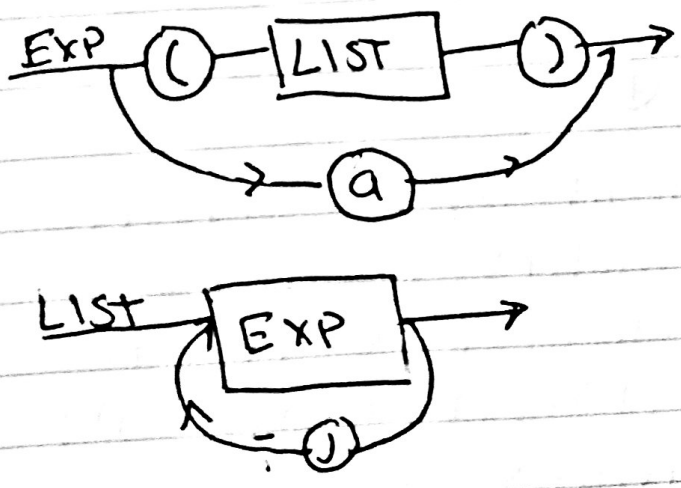
1) EXP ::= (LIST) | a
LIST ::= LIST, EXP | EXP

1a)



1b) EBNF
EXP ::= (LIST) {a}
LIST ::= <exp> {, exp}

1c)



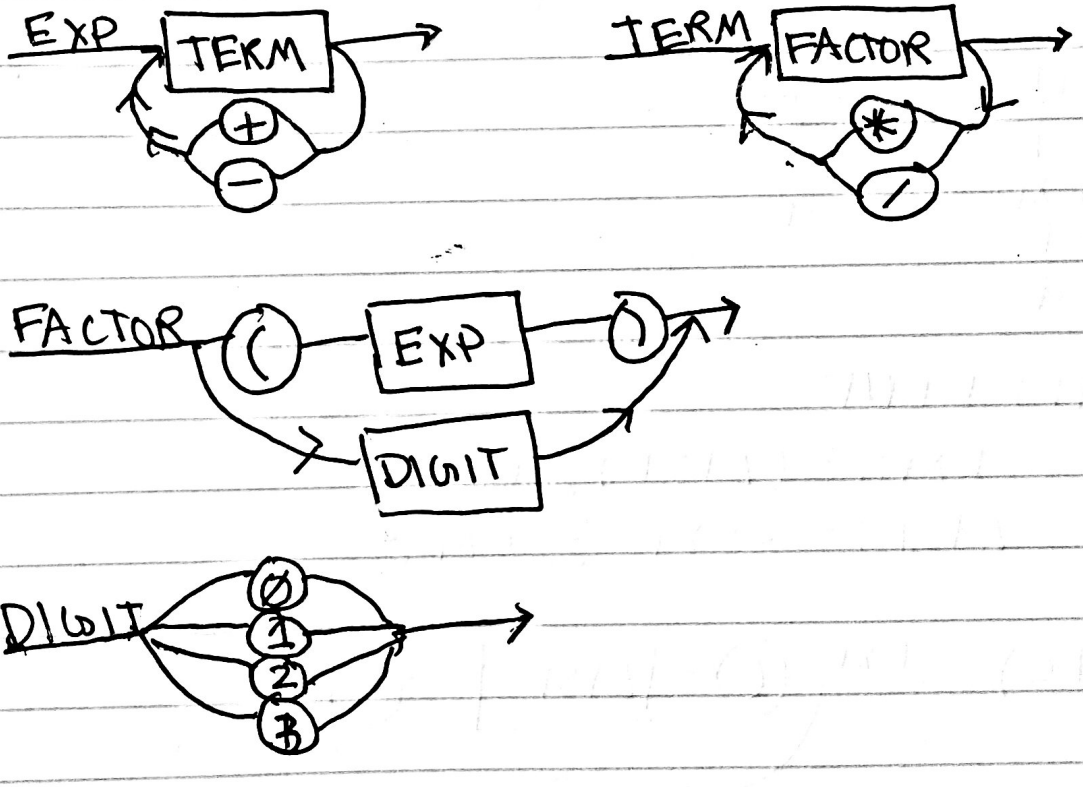
1d) FIRST(EXP) = { a, (}
~~FIRST(LIST) = {) }~~
FIRST(LIST) = FIRST(EXP)
= { a, (}
FOLLOW(LIST) = {) }

FOLLOW(EXP) = FOLLOW(LIST)
∪ { , }
= {), , }

2) EXP ::= EXP + TERM | EXP - TERM | TERM
TERM ::= TERM * FACTOR | TERM / FACTOR | FACTOR
FACTOR ::= (EXP) | DIGIT
DIGIT ::= 0 | 1 | 2 | 3

a) EBNF
EXP ::= [(+ | -)] TERM
TERM ::= [(* | /)] FACTOR
FACTOR ::= (EXP) { DIGIT }
DIGIT ::= 0 | 1 | 2 | 3

b)



c): • First set of any two choices must no
have any tokens in common
ex) FACTOR ::= (EXP) | DIGIT
FIRST(EXP) ∩ FIRST(DIGIT) = ∅
• When structures are optional
ex) S → B [A] D
If A is optional then;
FIRST(A) ∩ FOLLOW(A) = ∅

2d) FIRST(DIGIT) = { 0 1 2 3 }

FIRST (FACTOR) = FIRST(EXP) ∪ FIRST(DIGIT)

= { (} ∪ { 0 1 2 3 } = { (0 1 2 3 }

First (TERM) = FIRST(FACTOR) = { (0 1 2 3 }

FIRST(EXP) = FIRST(TERM) = { (0 1 2 3 }

FOLLOW(EXP) = {) }

FOLLOW(TERM) = { + - } ∪ FOLLOW(EXP)

{ + - } ∪ {) } = { + -) }

FOLLOW(FACTOR) = { * / } ∪ FOLLOW(TERM)

= { * / + -) }

FOLLOW(DIGIT) = FOLLOW(FACTOR) = { * / + -) }

2e) FIRST(DIGIT) ∩ FOLLOW(DIGIT) =

{ 0 1 2 3 } ∩ { * / + -) } = ∅