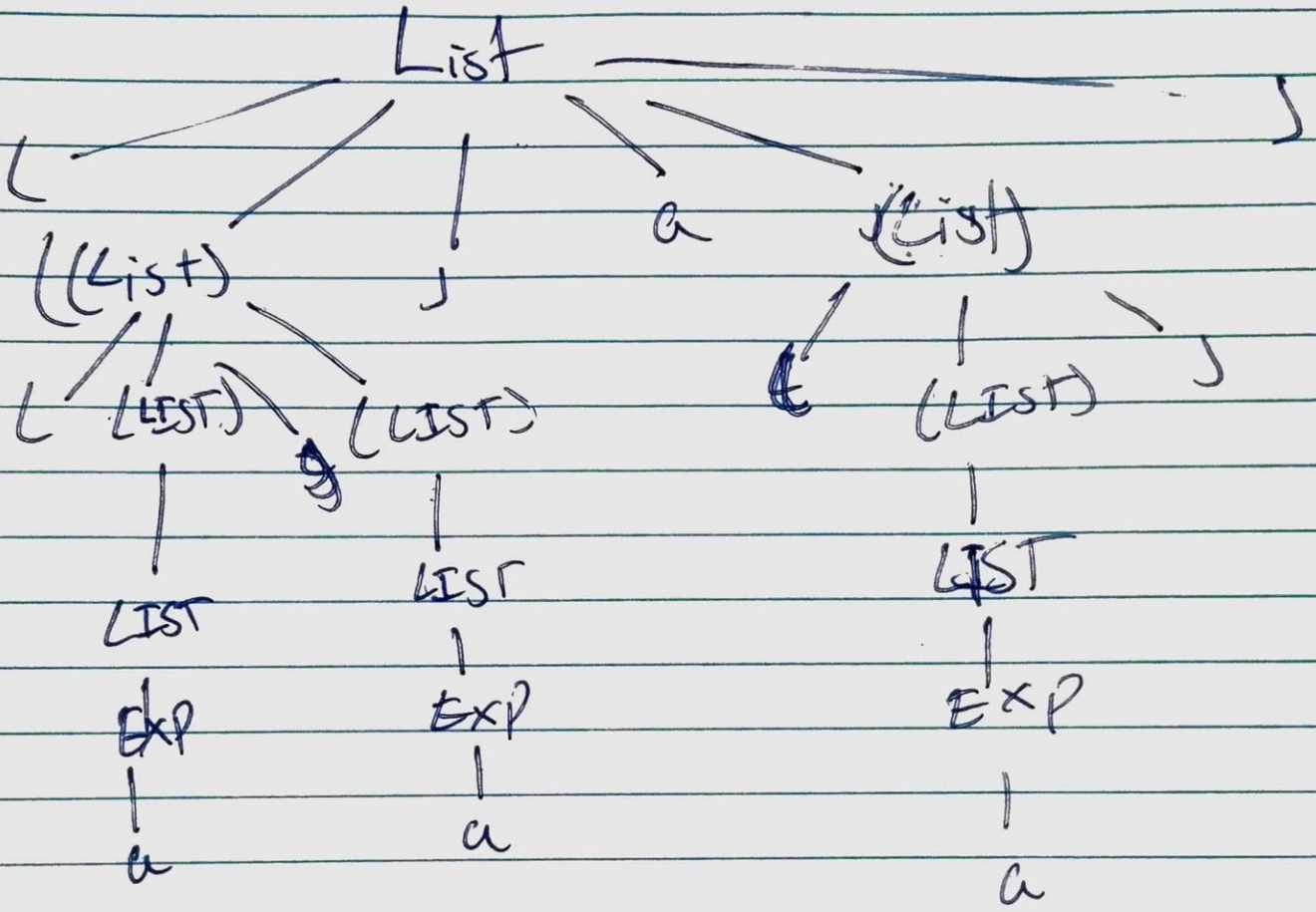
Consider following BNF grammar

EXP::= (LIST) | a

LIST::= LIST, EXP | EXP

Parse tree for

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

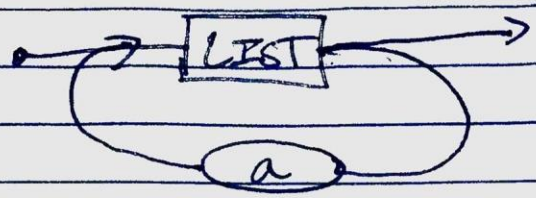


EBNF:

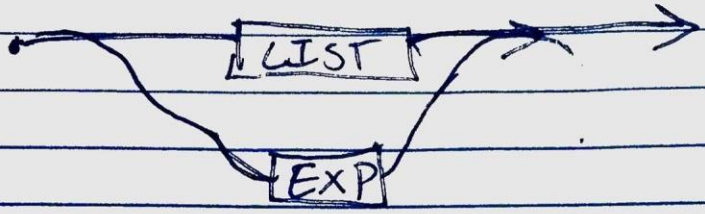
EXP = LIST | a

LIST = LIST, {EXP};

EXP:



LIST



2) Consider grammar:

EXP ::= EXP + TERM | EXP - TERM | TERM

TER ::= TER * FACTOR | TERM / FACTOR | FACTOR

FACTOR ::= (EXP) | DIGIT

DIGIT ::= 0 | 1 | 2 | 3

Convert to EBNF

EXP = { EXP ["+" | "-"] } TERM

TERM = { TERM ["*" | "/"] } FACTOR

FACTOR = ["(" EXP ")" | DIGIT]

DIGIT = "0" | "1" | "2" | "3"

FIRST(EXP) = (|0|1|2|3 = (, 0, 1, 2, 3,].

FIRST(TERM) = (|0|1|2|3 = 0, 1, 2, 3, (

FIRST(FACTOR) = (, 0, 1, 2, 3,

FIRST(DIGIT) = 0|1|2|3 = 0, 1, 2, 3

FOLLOW(EXP) = $, +, −,)

FOLLOW(TERM) = *, 1, +, −, $,)

FOLLOW(FACTOR) = *, 1, +, −, $,)

FOLLOW(DIGIT) = *, 1, +, −, $,)