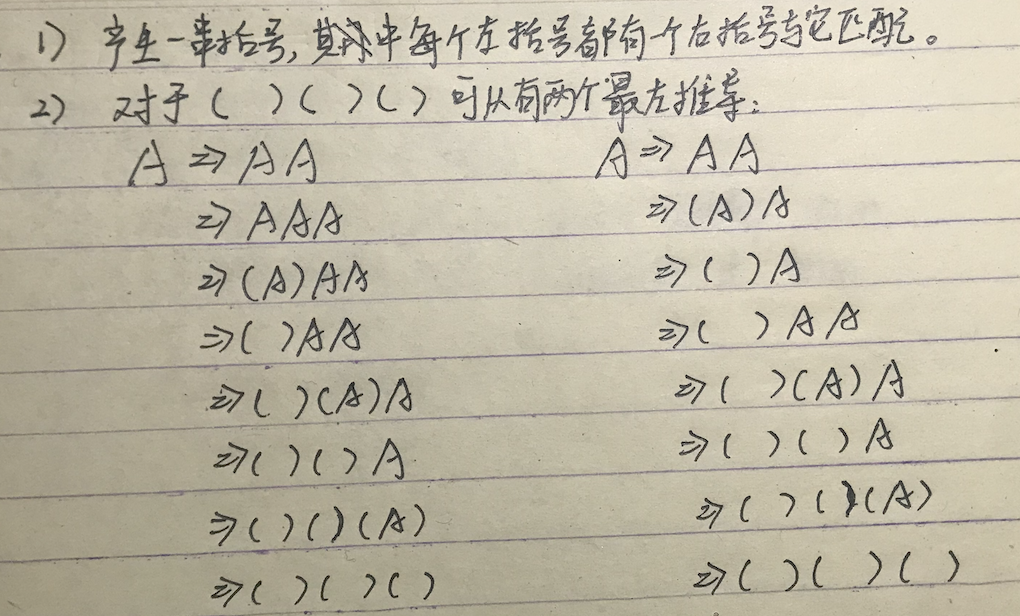
3.2



3.3

1. 3+4\*5-6

leftmost derivation:

exp => exp addop term

=> exp addop term addop term

=> term addop term addop term

=> factor addop term addop term

=> number addop term addop term

=> number + term addop term

=> number + term mulop factor addop term

=> number + factor mulop factor addop term

=> number + number mulop factor addop term

=> number + number \* factor addop term

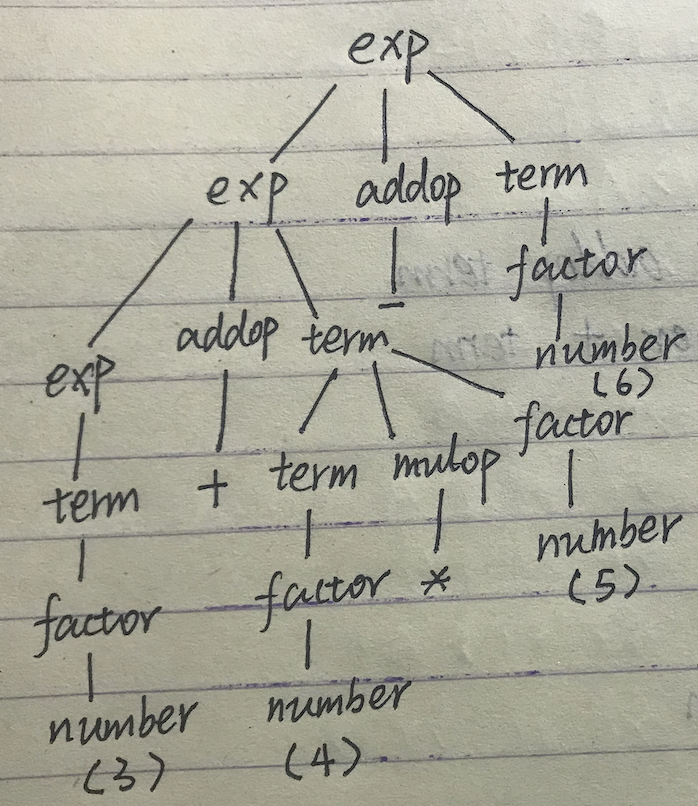
=> number + number \* number addop term

=> number + number \* number - term

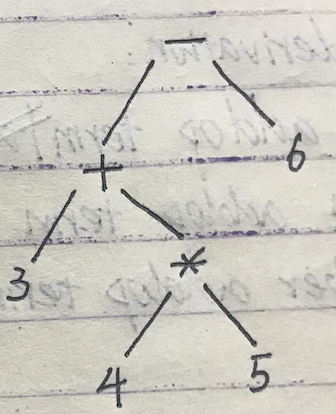
=> number + number \* number - factor

=> number + number \* number - number

parse tree:



abstract syntax tree:



1. 3\*(4-5+6)

leftmost derivation:

exp => term

=> term mulop factor

=> factor mulop factor

=> number mulop factor

=> number \* factor

=> number \* ( exp )

=> number \* ( exp addop term )

=> number \* ( exp addop term addop term )

=> number \* ( term addop term addop term )

=> number \* ( factor addop term addop term )

=> number \* ( number addop term addop term )

=> number \* ( number - term addop term )

=> number \* ( number - factor addop term )

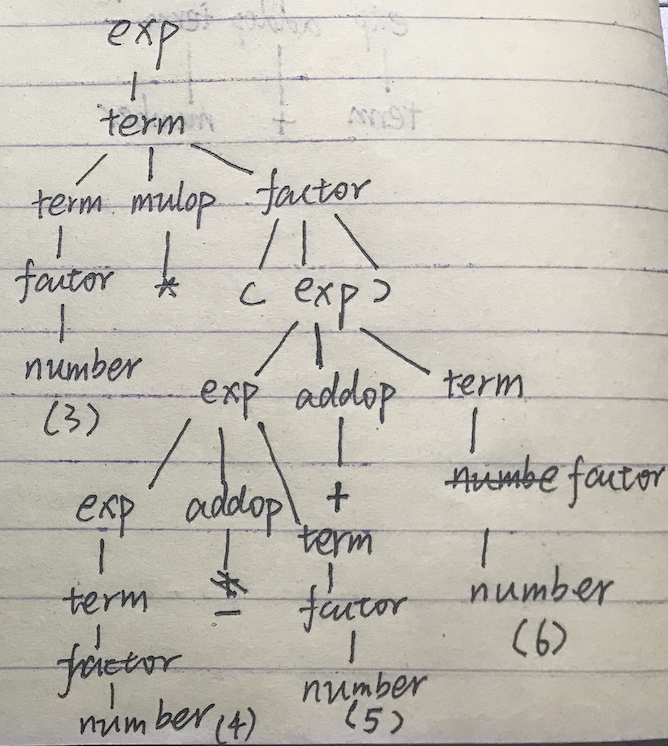
=> number \* ( number - number addop term )

=> number \* ( number - number + term )

=> number \* ( number - number + factor )

=> number \* ( number - number + number )

parse tree:



abstract syntax tree:



1. 3-(4+5\*6)

leftmost derivation:

exp => exp addop term

=> term addop term

=> factor addop term

=> number addop term

=> number – term

=> number – ( exp )

=> number – ( exp addop term )

=> number – ( term addop term )

=> number – ( factor addop term )

=> number – ( number addop term )

=> number – ( number + term )

=> number – ( number + term mulop factor )

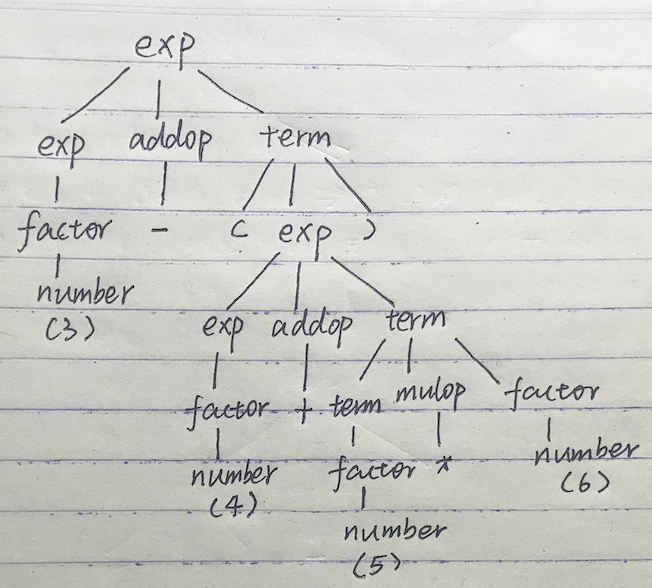
=> number – ( number + factor mulop factor )

=> number – ( number + number mulop factor )

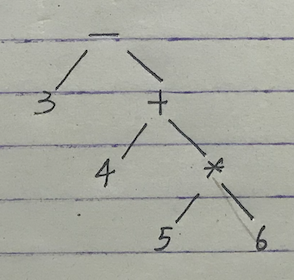
=> number – ( number + number \* factor )

=> number – ( number + number \* number)

parse tree:



abstract syntax tree:



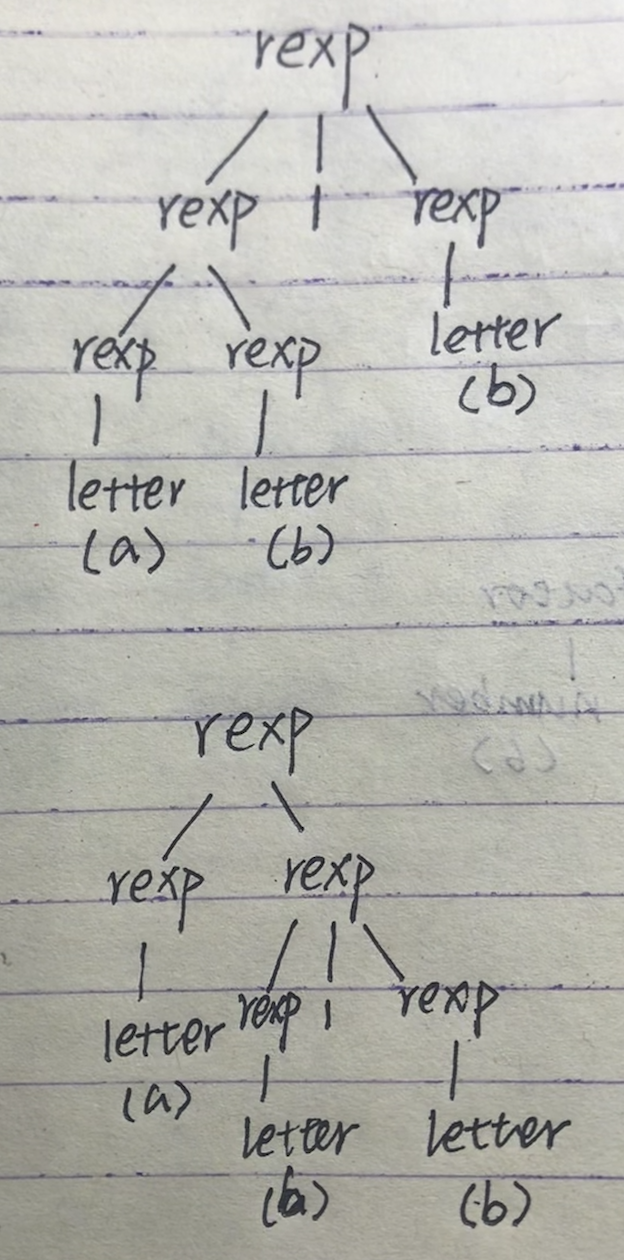
3.4

a.

rexp => rexp \* => (rexp) \* => (rexp | rexp) \* => (rexp rexp | rexp) \* => (a rexp | rexp) \*

=> (a b | rexp) \* => (a b | b) \*

b. ab|b:



c.

rexp -> rexp “|” term | term

term -> term factor | factor

factor -> high\* | high

high -> “(” rexp “)” | letter

d.

左结合，因为采取左递归的形式，所以二元运算符| 采用的是左结合的方式，一元运算符\*和左边的结合。