

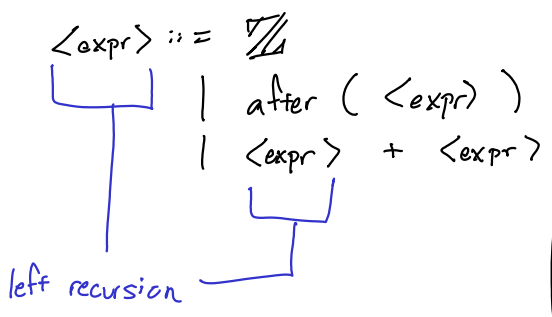
true  
tr ue  
true1

Lexing: ~~make~~ series of mini-lexers and then combine into one

input: char list  
output: token & char list

Parser inters structure

Overall: started w char list  
got token list

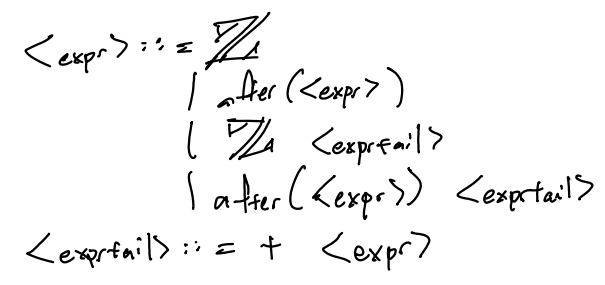


L L parser  
 ↑ ↑  
 reads builds  
 from from  
 left left

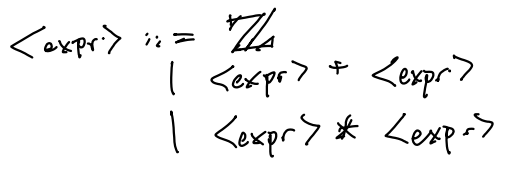
cannot handle left-recursion

Solution: transform grammar to eliminate left-recursion

1. Copy every production w/o left rec
2. Create new rule for all left-recursive cases, leaving out left-recursive nonterminal
3. Copy every production again, but add the new rule



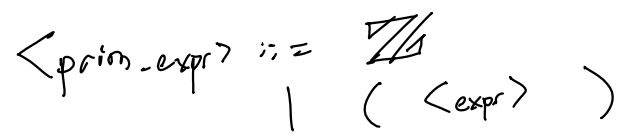
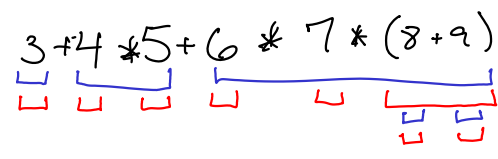
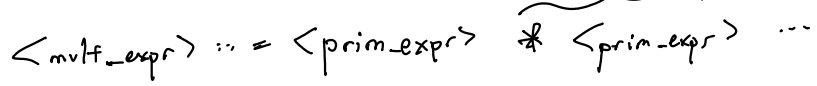
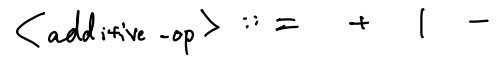
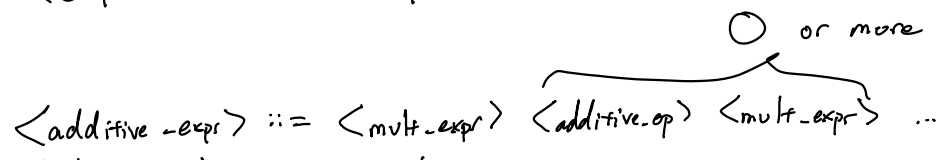
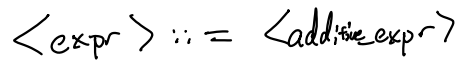
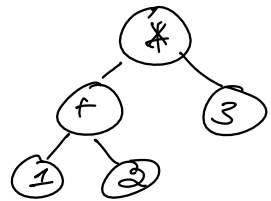
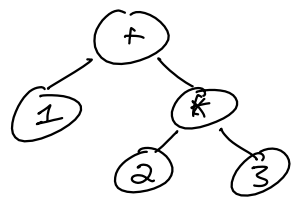
For your lab, grammar is nearly equivalent



Precedence (associativity)

$x || (y \&\& z)$

$1 + 2 * 3$



$4 - 1 - 2$

