

Lab 01 / Lab 02 check off
Lab 03 - parser

Lab 01 C data structures, map-
file I/O.

Lab 02 Scanner text \rightarrow tokens

Lab 03 Parser tokens \rightarrow tree

Prep for Project 02

ntcalc \rightarrow simplify
ntlang

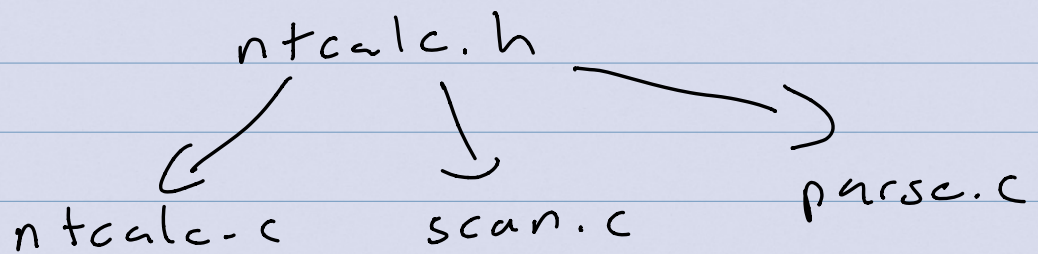
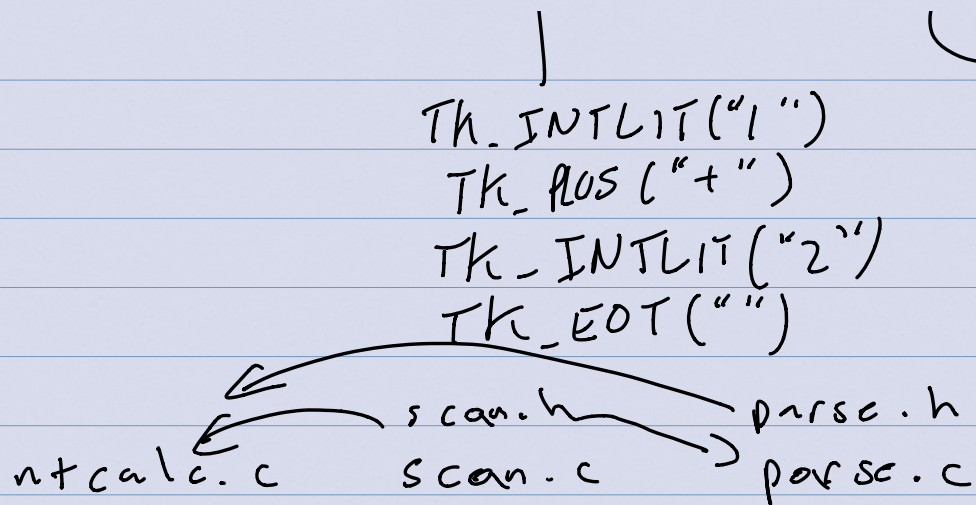
EBNF for a scanner is called
the "micro syntax"

EBNF for a parser
 \uparrow the syntax

grammars

"1 + 2" \rightarrow tokens \rightarrow tree





EBNF Syntax

$1 + 2$ `" + + 2 "`
 $1 + 3 + 4$
 $(1 + 2) * 7$

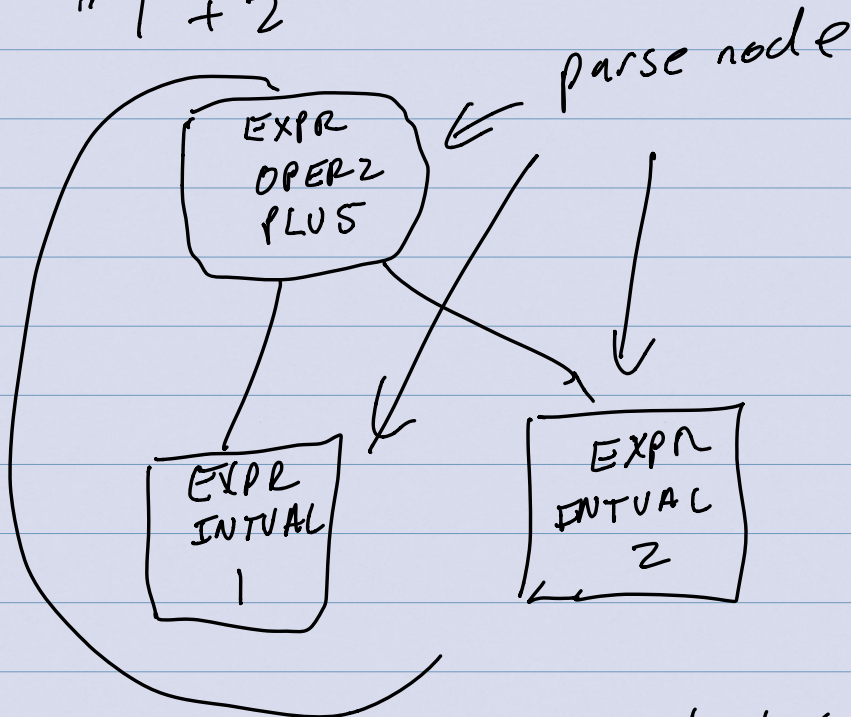
`expression ::= operand (operator operand)*`
`EOT`

`operand ::= integer (TK_INTLIT)`
 `| '-' operand`
 `| '(' expression ')'`

$$1 + 2 + 3 + 4 + 5$$

$$(((1 + 2) * 4) + 5)$$

"1 + 2"



parse tree → abstract syntax tree

① 1 + 2

2 - 2

2 @ @ 2

tokens → parse

TK_INTVAL	TK_PLUS	TK_INTVAL	TK_EOF	
"1"	"+"	"2"	" "	

recursive descent parsing

"1+2"

parse_expression(...)

←
parse_operand()

accept(TK_PLUS)

parse_operand()

parse_operand(...)