Lab01 / Lab02

Lab03 — parser

C   lab01   data structure, map
                file I/o
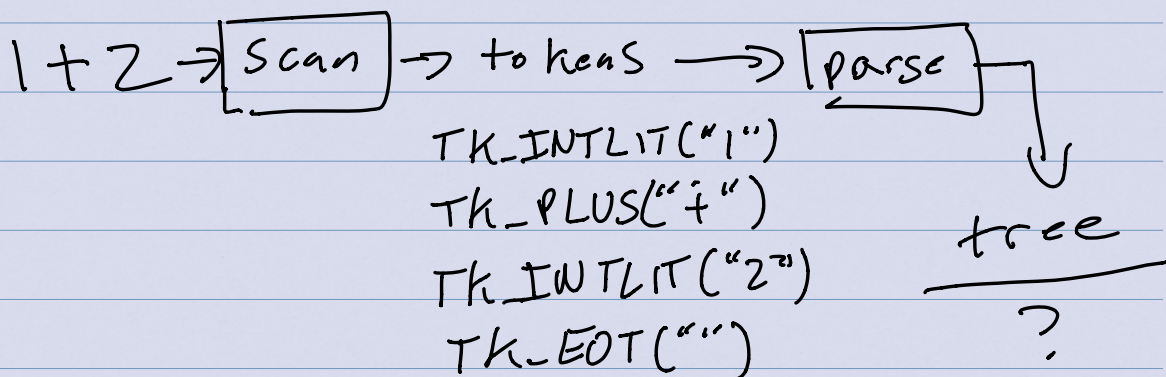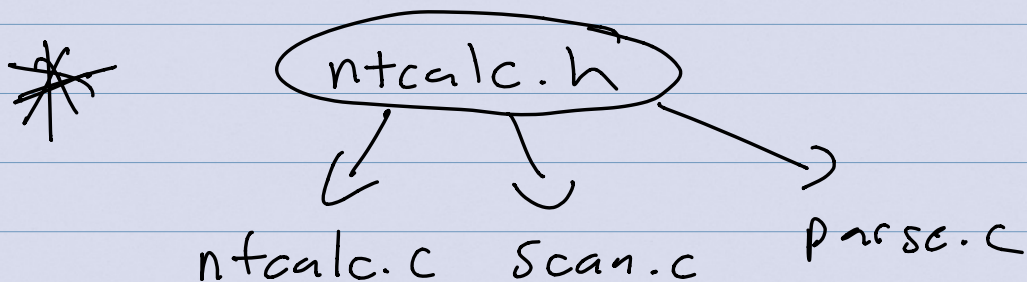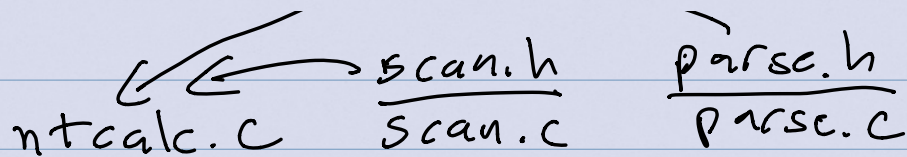Scanner   Lab02   text → tokens
parser   Lab03   tokens → tree

Prep for Project02

ntcalc → simplify
ntlang

EBNF for a scanner is called
the micro syntax

EBNF for a parser
        syntax
tokens

① ⊕ ②

ntcalc.c ← ← → scan.h | parse.h
⎯⎯⎯⎯ | ⎯⎯⎯⎯
scan.c | parse.c

✳ (ntcalc.h)
↙ ↓ ↘
ntcalc.c scan.c parse.c

1+2 → [Scan] → tokens ⟶ [parse] ⌐
TK_INTLIT("1")
TK_PLUS("+")   tree
TK_IWTLIT("2")   ⎯⎯⎯⎯
TK_EOT("")   ?

EBNF Syntax

                                    TK_PUS
                                    TK_MINUS...
                                       ↑
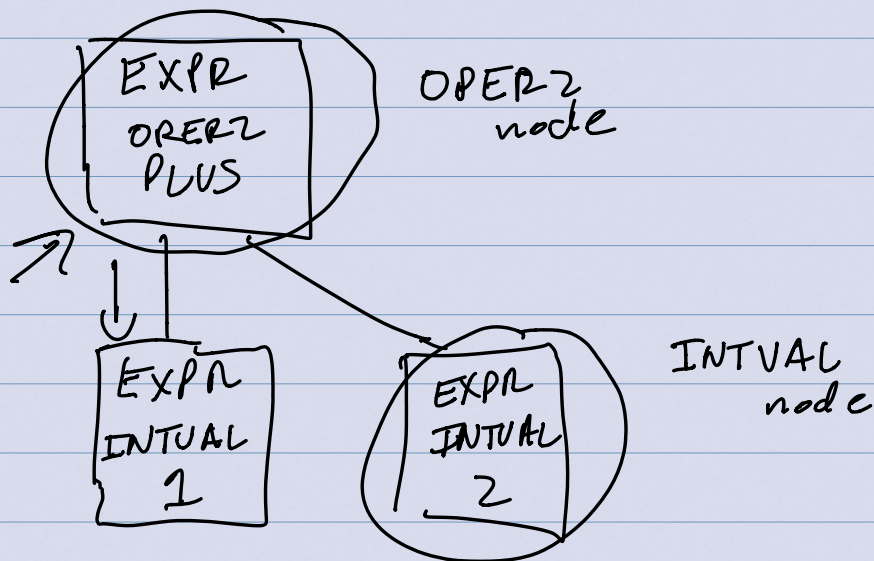
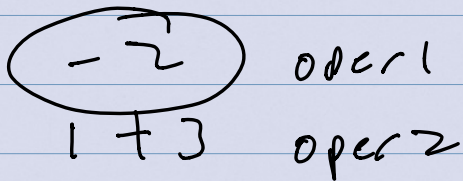expression ::= operand ([operator] operand)✳
operand ::= integer TK_INTLIT
          | '-' operand
          | '(' expression ')'

openwl operator | '(' expression ')'
      ↙
①+② → operand

2 ✳ (1+3)

"1 ⊕ 2"

EXPR
ORER2
PLUS

OPER2
node

EXPR
INTUAL
1

EXPR
INTUAL
2

INTVAL
node

---

trees in C   structs and unions

(- 2) oper1

1 + 3 oper2

tokens → parse tree

TK_INTLIT	TK_PLUS	TK_INTLIT	TK_EOT
"1"	"+"	"2"	" "

1 + 2

Recursive descent parser