# Report_Lab1_21302010040

**21302010040 叶天逸**
主机用户名
Win usr@hostname: Lenovo@IVANIVY-SAVIOR
WSL usr@hostname: ivan@IVANIVY-SAVIOR
2024-03-26
13:46

# 主要程序

发现 lex 语法必须在要转换的字符后空一格, 且句子最好顶格:

```
"xxx" {xxxx}
```

# TeaplAst.h 中定义的43个结构

```
pos
program
programElementList
programElement
arithExpr
exprUnit
structDef
varDeclStmt
fnDeclStmt
fnDef
type
varDecl
varDef
rightVal
boolExpr
arithBiOpExpr
arithUExpr
fnCall
indexExpr
arrayExpr
memberExpr
boolUnit
boolBiOpExpr
boolUOpExpr
comExpr
leftVal
assignStmt
rightValList
varDefScalar
```

```
varDefArray
varDeclScalar
varDeclArray
varDeclList
paramDecl
fnDecl
codeBlockStmt
ifStmt
whileStmt
callStmt
returnStmt
codeBlockStmtList
tokenId
tokenNum
```

```
make compiler
./compiler ./tests/temp.tea ./tests/temp.ast
./compiler ./tests/int_io.tea ./tests/int_io.ast
./compiler ./tests/full_conn.tea ./tests/full_conn.ast
./compiler ./tests/insert_order.tea ./tests/insert_order.ast
./compiler ./tests/test03.tea ./tests/test03.ast
```

```
# 比较AST和REFAST
python compareAST.py ./tests/full_conn.refast ./tests/full_conn.ast
```

# 通过调试解决错误

```
Reading a token: Next token is token INT ()
Shifting token INT ()
Entering state 81
Reducing stack by rule 23 (line 333):
   $1 = token INT ()
-> $$ = nterm Type ()
Stack now 0 7 3 21 49 89 124 152
Entering state 159
Reducing stack by rule 74 (line 668):
   $1 = token FN ()
   $2 = token ID ()
   $3 = token LP ()
   $4 = nterm ParamDecl ()
   $5 = token RP ()
   $6 = token RA ()
   $7 = nterm Type ()
-> $$ = nterm FnDecl ()
Stack now 0 7
Entering state 12
Reading a token: Next token is token LC ()
syntax error
Error: popping nterm FnDecl ()
Stack now 0 7
Error: popping nterm ProgramElement ()
Stack now 0
Cleanup: discarding lookahead token LC ()
Stack now 0
0a1,355
> A program
```

这里可以通过吃掉的token和 y.out 的研究发现是 `FnDecl LP ParamDecl RP` 中的 `ParamDecl` 可以为空但是我没写成空
这个心路体现了调试的作用

# 对比树

REFAST

```
279         |--A_arithExpr
280             |--A_arithBiOpExpr
281                 |--A_arithExpr
282                     |--A_exprUnit
283                         |--A_arrayExpr
284                         |--A_leftVal
285                             |--A_memberExpr
286                             |--A_leftVal
287                                 |--A_arrayExpr
288                                 |--A_leftVal a[
289                                     |--A_indexExpr 1].a[
290                             |--A_indexExpr 4]*
291                 |--A_arithExpr
292                     |--A_exprUnit 47+
```

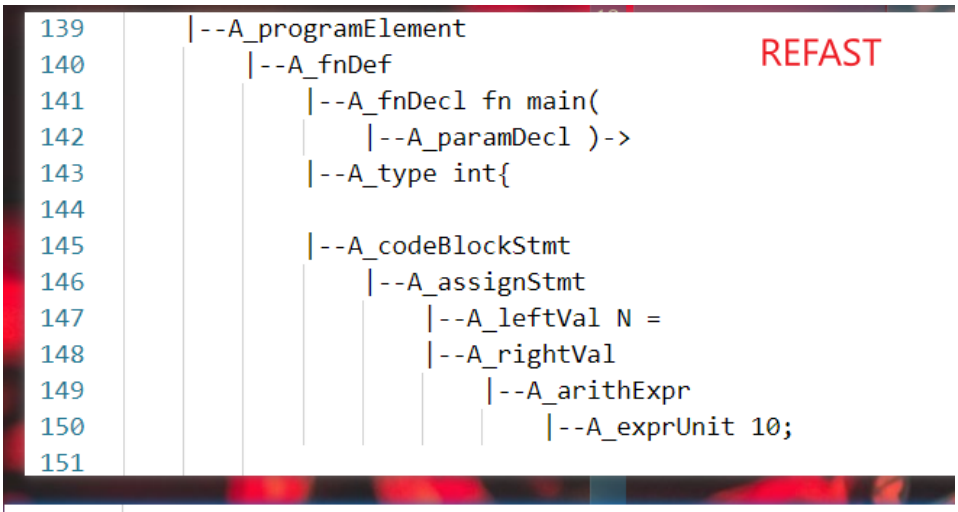
AST

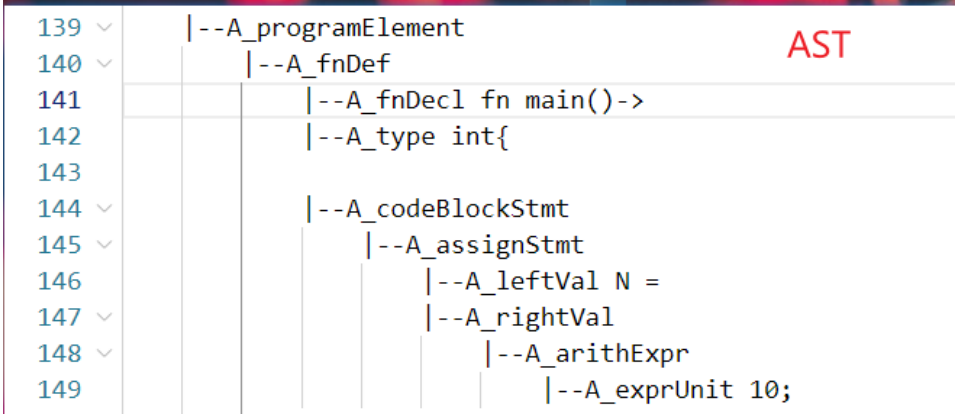
```
329             |--A_arithExpr
330                 |--A_exprUnit
331                     |--A_arrayExpr
332                     |--A_leftVal
333                         |--A_memberExpr
334                         |--A_leftVal
335                             |--A_arrayExpr
336                             |--A_leftVal a[
337                                 |--A_indexExpr 1].a[
338                         |--A_indexExpr 4]*
339         |--A_arithExpr
```

即使能够输出, 树也会有所不同

两个树有所不同, 可以直观看出 REFAST 先解析成 arithBiOpExpr 我们目前还看不出来是什么问题

# insert_order




```
FnDecl: FN ID LP ParamDecl RP RA Type
ParamDecl: VarDeclList
```

根据输出和规则我们可以发现, 这里的 ParamDecl 应该到 VarDeclList 才有空集

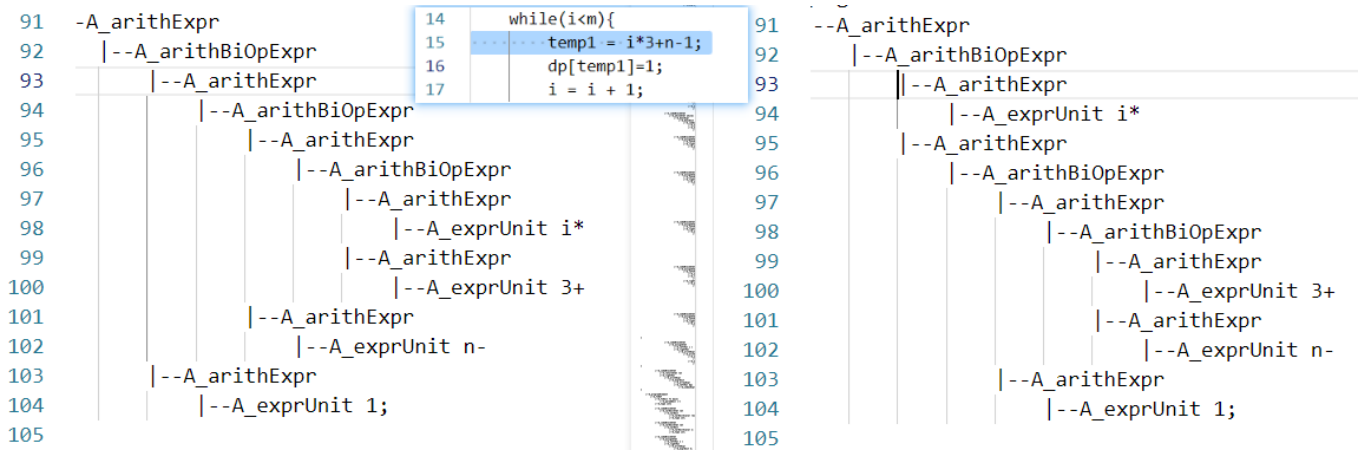
```
VarDeclList:
VarDecl COMMA VarDeclList
{
  $$ = A_VarDeclList($1, $3);
}
| VarDecl
{
  $$ = A_VarDeclList($1, nullptr);
}
|
{
  $$ = nullptr;
}
;
```

通过测试 insert_order

# 优先级:

发现优先级问题:


int_io: 右侧是参考答案,




unique_path: 左侧是参考答案.

发现原来是优先级问题, 之前那个也是. 将乘除放在加减下面行解决

```
clang++ -std=c++17 -g -c compiler.c
clang++ -std=c++17 -g -o compiler y
PASS insert_order
PASS temp
PASS brainfk
PASS test03
PASS unique_path
PASS int_io
PASS int_split
PASS test0
PASS hanoi
PASS test04
PASS register_alloca
PASS jump_game
PASS test07
PASS dijkstra
PASS full_conn
PASS bin_search
PASS DFS
PASS test01
PASS BFS
PASS test06
PASS sort_test5
PASS sort_test7
PASS palindrome_number
PASS test02
PASS expr_eval
PASS test05
PASS line_search
```

21302010040

全过

# 冲突

目前没有冲突了, 之前改空值搞出好多, 发现是优先级问题后又改回来了

unique_path.refast | parser.yacc M | y.output

src > y.output

```
Grammar

    0 $accept: Program $end

    1 Program: ProgramElementList

    2 ProgramElementList: ProgramElement ProgramElementList
    3                   | %empty

    4 ProgramElement: VarDeclStmt
    5               | StructDef
    6               | FnDeclStmt
    7               | FnDef
    8               | SEMI

    9 ArithExpr: ExprUnit
   10          | ArithBiOpExpr
```