

1. 实验过程

只通过了 parser-1。

模板参考自 [SYSU-lang/parser.y 模板](#)，思路是差不多的，但是定义了自己的数据结构。

按照 [实验二语法分析 Wiki](#) 中的文法编写语法规则段，通过对比 clang 的语法分析输出确定 kind、name 和 value，通过对 parser-1 的单元测试不断调整代码，主要工作大部分在类型判断和 ImplicitCastExpr 的判断上。

2. 遇到的问题及解决方法

1. IF-Else 的移入/规约冲突

解决办法：参考 [Bison IF-ELSE 移入/规约冲突解决办法](#)

2. 我觉得 ImplicitCastExpr 的判断是 parser-1 中比较难的地方，除了左值到右值的转换，还有类型转换（Integer到Float，Float到Integer等等），后者除了变量赋值时需要考虑，还有函数传参时也要考虑，通过建立符号表记录变量的类型和函数的返回值、参数类型就可以解决。

函数的符号表是容易建立的，而变量的符号表不容易，因为需要考虑全局变量和局部变量，更详细一点，如果一个变量在一个Block（花括号{}包起来的语句块）之前被定义，那么在该Block中再定义一个相同名字的变量，Block中变量的类型就会覆盖Block之前相同名字变量的类型。

解决方法：对每一个变量使用一个栈来存储类型，每当遇到一个变量的定义时，就在该变量对应的栈中，将此定义中的类型压栈，之后每次需要使用一个变量的类型时，就按照其栈顶的类型来处理。在一个Block规约完成时，代表该Block结束了，那么就在该Block中，寻找在此Block中被定义的所有的变量（寻找kind为VarDecl的节点），然后将这些变量对应的栈都进行一次出栈操作，弹出栈顶的类型。

3. 自动评测结果

```
zhou@LAPTOP-B0ER6N6Q:~/SYSU-lang$ CTEST_OUTPUT_ON_FAILURE=1 cmake --build $HOME/sysu/build -t test
[0/1] Running tests...
Test project /home/zhou/sysu/build
  Start 1: lexer-0
1/10 Test #1: lexer-0 ..... Passed    0.84 sec
  Start 2: lexer-1
2/10 Test #2: lexer-1 ..... Passed  344.64 sec
  Start 3: lexer-2
3/10 Test #3: lexer-2 ..... Passed  346.29 sec
  Start 4: lexer-3
4/10 Test #4: lexer-3 ..... Passed  347.20 sec
  Start 5: parser-0
5/10 Test #5: parser-0 ..... Passed    0.95 sec
  Start 6: parser-1
6/10 Test #6: parser-1 ..... Passed 1006.16 sec
  Start 7: parser-2
7/10 Test #7: parser-2 ..... ***Failed    1.72 sec
[0/55] Done /home/zhou/SysU-Lang/lexer/lexer-0, lexer-1, lexer-2, lexer-3, parser-0, parser-1, parser-2
```