

编译原理 实验 2: 语义分析

冯诗伟 161220039

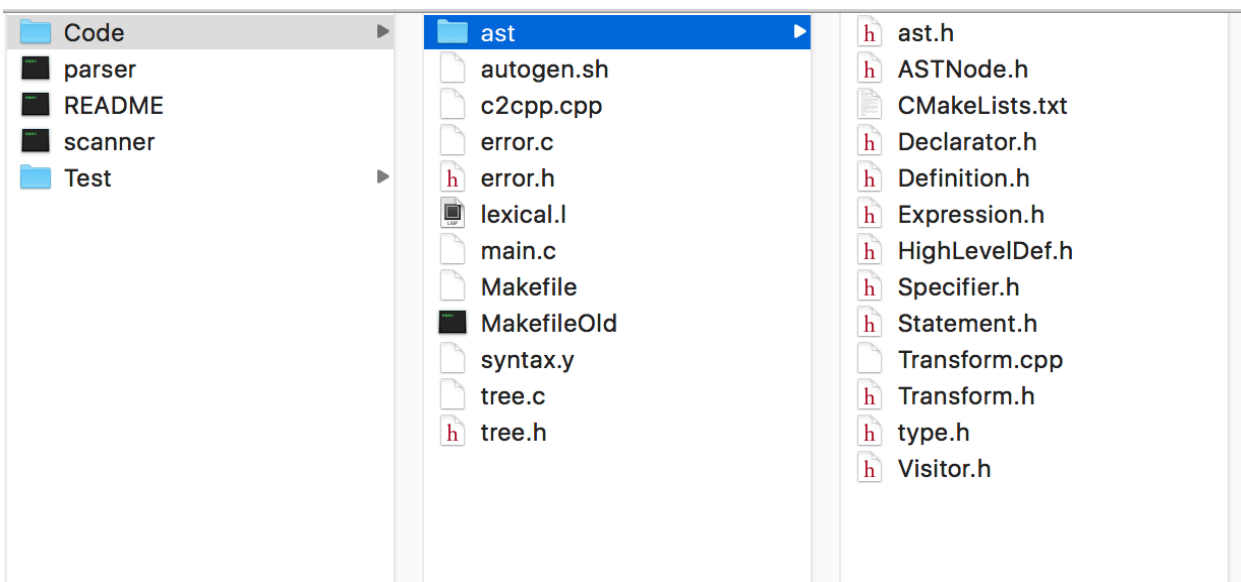
1 记一次惨痛的实验经历

由于拖延症又犯了，导致给实验二留的时间严重不足，于是想着能不能减少一点实验的工作量从而实现ddl的冲刺，然后就进了一个大坑：把语法书重构成抽象语法树 + 使用 visitor 模式。因为 visitor 模式非常适合对于语法树这种不常变动、节点种类较多且操作种类较多的结构。

由于对自己的进度和水平有着过于乐观的估计，实验在有限的时间内进度十分缓慢，只是把 C 的语法树转成了抽象语法树，用 visitor 来对 AST 进行操作分析的部分尚未写完，最后未能完成语义分析。

2 如何编译

由于改成了 C++ 代码，所以也擅自修改了 Makefile(如下图)，里面使用了 g++。MakefileOld 是助教之前给的文件，Makefile 是用来编译 C/C++ 的新文件。



3 文件结构

从上图可以看到，Code 文件夹下 c2cpp.cpp 是用来从 C 转化到 C++ 的代码；剩下的都是实验一中的代码。

Code/ast 文件夹下：

- ASTNoode.h: 定义了 AST 的节点结构的抽象类

```
1      struct ASTNode{  
2          int beginLine;  
3          virtual void accept(Visitor &visitor) = 0;  
4          virtual ~ASTNode(){};  
5      };
```

- Declarator.h Definition.h Expression.h HighLevelDef.h Specifier.h Statement.h 是根据文法定义了具体的 AST 中节点的不同类型，他们都是 ASTNode 的子类。
- Transform.h Transform.cpp 是把语法树转换为 AST 的代码
- Visitor.h 定义了 visitor 们的抽象类，至于具体定义什么功能，会定义它的子类来实现。这一部分的代码还没实现完整，也还通不过编译，所以就没有提交。