

编译原理实践课程报告

李一帆

10184102137

- (1) 代码精简且典型，写法规范，可读性较高
- (2) 完成时间较为提前，词法分析和 LL1 文法在布置下来几天后成功通过所有样例
- (3) 代码性能较高，能在较短时间内对大量代码进行编译
- (4) 分析表存储结构自然且直接，通过数组方式在 $O(1)$ 时间内读写
- (5) 使用面向对象思想将终结符、非终结符、状态等要素归类，使用的函数功能划分明确，耦合度低，可移植性高
- (6) 代码采用的操作较底层，没有特定环境需求，方便多个平台进行使用
- (7) 采用 C++ 编写，易与其他 linux 开源软件以及其他例如 llvm 之类的编译器相结合，没有语言障碍问题
- (8) 语法分析的分析表结构均为代码自动生成，由于文件不可修改等问题并未直接上传，相应代码库为：
<https://github.com/YiFan-Evan/parsing-table.git>
- (9) 代码的测试经过更多的样例测试，覆盖了每种产生式被使用的情况，保证多种情况不会出错。具体测试集为：
<https://github.com/YiFan-Evan/compiler-test-benchmark.git>