

2025 年编译原理与技术实践课程实验报告

姓名: 郭雅颖

学号: 10235101570

课程名称: 编译原理与技术实践

指导教师: 张敏

实验日期: 2026 年 1 月 14 日

实践亮点

- 在 LR 分析器 中, 利用 maker.cpp 从文法自动生成 FIRST/FOLLOW 集和 SLR 分析表, 并导出为头文件 LRTTable.h, 避免人工推导和运行时建表开销。 (3)
- 在 语法树可视化 模块中, 实现了自动生成 AST 的 Graphviz DOT 语言与图片导出, 直观对比 LL 与 LR 的建树差异。 (2、3)
- 在 符号管理 模块中, 对不同阶段选择了最佳实践的数据结构 (map、Trie + 哈希表双向映射、二维数组、unordered_map, 实现时间复杂度从 $O(1)$ 到 $O(\log n)$ 级的增删改查, 提升效率的同时平衡可读性易用性。 (1、2、3、4)
- 在 词法与语法分析 阶段, 采用 DFA + 表驱动架构 : 词法分析器用 DFA 实现复杂 Token 识别(1), LL 分析器基于预测分析表实现无回溯的自顶向下解析(2), LR 分析器依托 ACTION/GOTO 表完成自底向上的移进 - 规约解析(3)。 (1、2、3)
- 在 语义分析 阶段, 采用了解释器+中间代码生成并行, 实现了变量类型自动提升, 并支持中间代码和符号表快照导出。 (4)
- 在 错误处理环节中, 设计了多层次错误恢复机制: 语法层采用恐慌模式、虚拟插入与前瞻验证(2、3); 语义层实现了类型检查与运行时错误检测(4)。 (2、3、4)
- 在 错误定位环节 中, 设计了 精确行号追踪机制, 在存在空行或跨行语句时仍能将错误定位到实际出错语句行。 (2、3)
- 在 测试与验证 中, 构造了覆盖正确和多类错误用例的测试集, 编写了自动化测试流水线, 实现四个模块脚本一键编译、批量测试和结果归档。 (1、2、3、4)
- 在 项目架构上, 采用高内聚低耦合的模块化结构: LR 分析表形成生产者-消费者模式; 可视化模块独立封装, 通过统一接口串联成可扩展流水线。 (1、2、3、4)
- 在 项目管理 中, 使用 Git 托管到 GitHub 仓库, 便于版本管理 (<https://github.com/ying-2626/Principles-of-Compilation>)。 (1、2、3、4)