

1. 二、详细分工说明

1.1 1 成员 A：项目负责人 / 架构设计

主要职责：

- 设计 Compiler-Compiler 的整体系统架构
- 定义模块接口 (Scanner / Parser / CodeGen)
- 实现语法分析器自动生成模块：
 - LL(1) 或 LR (如 SLR / LR(1))
 - FIRST / FOLLOW 集计算
 - 分析表构造
- 负责模块集成与最终可运行版本
- 协调团队进度与代码合并 (Merge)

主要贡献代码：

- grammar_parser/
- parser_generator/
- main_driver/

1.2 2 成员 B：词法分析负责人

主要职责：

- 设计并实现扫描器自动生成模块
- 支持基于形式化词法规则的输入 (正则表达式 / 正则文法)
- 实现经典词法分析算法：
 - Thompson 构造 ($\text{Regex} \rightarrow \text{NFA}$)
 - 子集构造法 ($\text{NFA} \rightarrow \text{DFA}$)
 - 状态等价化方法 (DFA 最小化)
- 输出可被语法分析器调用的 Token 流
- 提供词法分析错误处理机制

主要贡献代码：

- lexer_generator/
- nfa/
- dfa/
- dfa_minimizer/

1.3 ③ 成员 C：中间代码生成负责人

主要职责：

- 设计测试语言的语法制导翻译方案
- 定义中间代码形式（如三地址码）
- 实现：
 - 属性文法
 - 语法制导翻译规则
- 在语法分析过程中同步生成中间代码
- 将中间代码保存到文件并展示结果

主要贡献代码：

- semantic/
- ir/
- tac_generator/

1.4 ④ 成员 D：测试与文档负责人

主要职责：

- 设计完整的测试方案：
 - 编译器生成器本身的测试（多语言规则）
 - 生成编译器后的源程序测试
- 编写测试用例与结果分析
- 负责项目文档撰写：
 - 程序设计文档（Word / PDF）
 - README（GitHub/Gitee）
- 制作答辩 PPT
- 汇总 GitHub 协作与贡献情况

主要贡献代码 / 文档：

- tests/
- docs/
- README.md
- slides/