

2025 年编译原理与技术实践课程报告

姓名：张梓卫

学号：10235101526

课程名称：编译原理与技术实践

指导教师：张敏

实验日期：2026/01/07

实践亮点

- 零构建验证：**验证无需编译、部署，依托于 [Github Pages](#) 静态站点托管静态 HTML 页面，公网访问，即点即用！（验证前端：<https://deralive.github.io/Compiler>）
- 零后端架构：**摒弃传统控制台输出，采用 WebAssembly 与 Emscripten 技术链将 C++ 代码编译为 `.wasm` 二进制文件作用户分发，在网页端输入测试用例，即可调用写好的 C++ 接口，无需依赖后端服务器，体验去中心化的毫秒级响应。
- 用户友好输入环境：**基于 `CodeMirror` 组件，提供语法高亮、行号显示、自动缩进等功能，给调试者类 VSCode 的舒适体验。自带可点击的示例代码库，涵盖各实验重点测试用例。
- (Lab 1) Thompson 可视化：**在词法分析实验中，为便于学习 Thompson 算法，采用 `Viz.js` 图形库，附加实现从正则表达式到 NFA 状态机的图形化渲染。能够将中缀正则表达式解析为后缀表达式。支持 `*`、`|`、`.` 等基本运算符的优先级处理与括号嵌套。
- (Lab 2) 交互式 FIRST 与 FOLLOW 集计算：**在 LL(1) 分析器的基础上封装了独立的集合计算模块，能够处理包含空串 (ϵ) 推导、非终结符依赖等复杂情况，实现了对任意文法 FIRST、FOLLOW 集的实时计算与可视化展示，辅助验证算法细节。
- (Lab 2) 语法树可视化：**LL(1) 语法分析器中集成 `echarts.js`，动态渲染可交互的语法推导树 (Parse Tree)，支持节点折叠、滚轮缩放与拖拽移动，直观展示递归下降的推导过程。
- (Lab 3) SLR(1) 分析表生成器：**支持输入上下文无关文法，自动计算闭包与 GOTO 表，动态生成由 `EChart` 模块驱动的 SLR(1) 预测分析表，通过颜色直观区分移进、归约状态。
- (Lab 4) 动态符号表与栈帧监视：**构建了可视化的内存布局监视器。能依据变量类型精确模拟栈帧分配：为 `int` 分配 4 字节，`real` 分配 8 字节，并自动计算 `Offset`。同时，实时追踪变量的定义行号。
- (Lab 2 / 3 / 4) 鲁棒性纠错：**在 LL(1) Lab 中利用恐慌模式，以同步词集合定位并跳过非法输入；在 SLR(1) Lab 中，针对常见的漏写分号等错误进行智能检测与自动补全。在语义分析 Lab 中，针对语义错误，程序会注入默认安全值，确保核心不崩溃，能一次分析多处错误。
- 工程实践：**使用类的形式定义 `Lexer, Parser, Grammar`，便于后续功能扩展与维护。用工程实践经验和单元测试策略，对词法、语法、语义功能进行隔离验证，确保模块间的低耦合性，每个实验增加至 10 个测试用例，自动化测试脚本可以批量进行单元测试，且涵盖多种报错情况。本实验报告采用 Typst 编写。

项目地址：<https://github.com/deralive/Compiler>