



廣東工業大學

实验报告

课程名称 编译原理

题目名称 PL/0 编译程序的修改扩充

学生学院 先进制造学院

专业班级 22 人工智能 (3)

学 号 3122009224

学生姓名 施乔

指导教师 康培培

2025 年 6 月 16 日

目录

编译原理实验报告：PL/0 编译器功能扩展	3
一、实验目的	3
二、实验要求	3
（一）功能要求	3
（二）技术实现要求	3
三、PL/0 编译器概述	3
（一）核心架构	3
（二）关键数据结构	3
四、关键功能实现	3
（一）词法分析扩展	4
（二）条件语句 ELSE 子句实现	4
（三）语义分析与中间代码生成	4
五、错误处理机制	5
六、测试分析	5
（一）测试程序 1	5
（二）测试程序 2	5
七、总结	6

编译原理实验报告：PL/0 编译器功能扩展

一、实验目的

本实验基于 PL/0 编译器进行功能扩展，旨在深入理解编译器工作流程，包括词法分析、语法分析、语义分析和中间代码生成等阶段。重点实现以下扩展功能：

- 增加保留字（ELSE, FOR, STEP, UNTIL, DO, RETURN）和运算符（*=, /=, &, |, !）；
- 将不等号#改为<>；
- 为条件语句增加 ELSE 子句支持。通过代码修改与测试，掌握编译器扩展的核心技术与错误处理机制。

二、实验要求

（一）功能要求

- 词法扩展：支持新保留字和运算符的识别，修改不等号逻辑；
- 语法扩展：实现 IF-THEN-ELSE 双分支结构，完善条件语句的编译逻辑；
- 语义支持：确保新语法结构的中间代码正确生成与执行。

（二）技术实现要求

- 基于递归下降分析法修改语法分析程序；
- 通过符号表管理变量作用域；
- 设计错误恢复机制处理语法错误。

三、PL/0 编译器概述

（一）核心架构

PL/0 编译器采用以语法分析为核心的单遍扫描架构，各模块调用关系如下：



其中词法分析和代码生成作为子程序被语法分析调用。

（二）关键数据结构

数据结构	作用	示例
符号表 (TABLE)	存储标识符属性	变量地址、常量值、过程层级
中间代码 (CODE)	存储类 P-code 指令	JMP 0 13（无条件跳转）
符号集合 (SYMSET)	管理语法分析上下文	STATBEGSYS 标记语句开始符号集

四、关键功能实现

（一）词法分析扩展

在 GetSym()函数中增加以下逻辑：

```
// 保留字识别：扩展关键字表 KWORD 与符号映射 WSYM
strcpy(KWORD[5], "ELSE");      WSYM[5] = ELSESYM;
strcpy(KWORD[7], "FOR");      WSYM[7] = FORSYM;
strcpy(KWORD[14], "STEP");     WSYM[14] = STEPSYM;
strcpy(KWORD[16], "UNTIL");    WSYM[16] = UNTILSYM;
strcpy(KWORD[15], "DO");      WSYM[15] = DOSYM;
strcpy(KWORD[13], "RETURN");  WSYM[13] = RETURNSYM;

// 运算符识别
case '&':
    GetCh();
    SYM = ANDSYM; // 逻辑与
    break;
case '|':
    GetCh();
    SYM = ORSYM;  // 逻辑或
    break;
case '!':
    GetCh();
    SYM = NOTESYM; // 逻辑非
    break;

// 不等号修改
case '<':
    GetCh();
    if (CH == '>') {
        SYM = NEQ; // 将"<>"识别为不等号
        GetCh();
    } else {
        SYM = LSS; // 小于号
    }
    break;
```

（二）条件语句 ELSE 子句实现

在 STATEMENT()函数中修改 IF 语句处理逻辑：

```
case IFSYM:
    GetSym();

    CONDITION(SymSetUnion(SymSetNew(THENSYM, DOSYM), FSYS), LEV, TX); // 解析条件表达式

    if (SYM == THENSYM)
        GetSym();
    else
        Error(16); // 缺少 THEN 报错

    CX1 = CX;
    GEN(JPC, 0, 0); // 生成条件跳转指令（地址待回填）

STATEMENT(SymSetUnion(SymSetNew(ELSESYM), FSYS), LEV, TX); // 解析 THEN 分支

    if (SYM != ELSESYM) {
        CODE[CX1].A = CX; // 无 ELSE 时回填跳转地址
    } else {
        GetSym();
        CX2 = CX;
        GEN(JMP, 0, 0); // 生成跳过 ELSE 的跳转
        CODE[CX1].A = CX; // 回填条件跳转至 ELSE 块起始
        STATEMENT(FSYS, LEV, TX); // 解析 ELSE 分支
        CODE[CX2].A = CX; // 回填 JMP 至 ELSE 块结束
    }
    break;
```

关键点：通过 JPC+JMP 指令组合实现分支控制，CX1/CX2 记录跳转地址用于回填。

（三）语义分析与中间代码生成

- 符号表管理：在 ENTER()函数中记录标识符类型（常量/变量/过程）及内存地址；
- 代码生成：GEN()函数根据语法分析结果生成三类指令：
 - 数据操作（LIT/STO）

流程控制（JPC/JMP）

运算指令（OPR）

指令类型	功能	示例
LOD	加载变量	LOD 0 3（加载第 0 层偏移 3 的变量）
OPR	执行运算	OPR 0 8（执行等于判断）
JPC	条件假跳转	JPC 0 11（条件假跳转至地址 11）

五、错误处理机制

通过 TEST()函数实现错误恢复：

```
void TEST(SYMSET S1, SYMSET S2, int N) {  
    if (!SymIn(SYM, S1)) {  
        Error(N); // 报错（如 Error(16)表示缺少  
THEN）  
        while (!SymIn(SYM, SymSetUnion(S1, S2)))  
            GetSym(); // 跳过非法符号直至合法符  
号出现  
    }  
}
```

错误类型示例：

Error(16): IF 后缺少 THEN

Error(22): 因子缺少右括号

Error(38): FOR 语句缺少 STEP

Error(39): FOR 语句缺少 UNTIL

六、测试分析

（一）测试程序 1

```
PROGRAM EX01;  
VAR A;  
BEGIN  
    READ(A);  
    IF A>3  
    THEN IF A<10  
    THEN WRITE(1)  
    ELSE WRITE(2)  
    ELSE WRITE(3)  
END.
```



（二）测试程序 2

```
PROGRAM EX02;  
VAR A,B,C;  
BEGIN  
    A /= 2; // 测试复合赋值  
    B *= 3;  
    C = A & B; // 测试逻辑运算符  
  
    // 测试新增保留字  
    ELSE;  
    FOR I:=1 STEP 1 UNTIL 5 DO  
        WRITE(I);  
    RETURN;  
END.
```



七、总结

本实验通过扩展 PL/0 编译器实现了 ELSE 子句支持、新增保留字与运算符，并优化了不等号语法。关键技术点包括：

递归下降分析法：通过修改 STATEMENT() 函数实现双分支条件语句的语法解析；

地址回填技术：利用 CX1/CX2 记录跳转地址，解决目标代码生成中的前向引用问题；

符号表层级管理：支持嵌套作用域下的变量寻址（如 LOD 指令的层级参数）。

实现的功能模块包含：

保留字扩展模块（FOR、STEP、UNTIL 等）

运算符扩展模块（*=、/=、&等）

实验表明，扩展后的编译器能正确识别新语法结构并生成高效类 P-code 代码，错误处理机制有效提升健壮性。测试用例验证了所有新增功能的有效性，包括条件分支、循环控制和复合赋值等功能。