

编译原理 AP_1

——实验报告

闫力敏*

1 概述

本次试验，在既有的框架下进行编写，完成了 PA_1 的实验要求，实现了 Complex 类，支持 Case, Super 表达式，对象复制，循环卫士语句，并在完成所有检测后，对一些其他的情况进行了测试，以及分析。

1.1 实验原理

Lexer.l 是词法分析，我们在这里通过正则表达式识别关键字（终结符），然后将所得的终结符回馈给 Parser.y 进行语法分析，Parser.y 在识别的过程中将数据储存到 SemValue 中，当然也根据 SemValue 构建语法节点。

Tree (Expr, 也是 Tree 的子类) 作为基类提供了打印函数的接口，因此可以通过实现不同的子类来实现新特性所要求的输出。

1.2 实验过程

1.2.1 Complex

首先，在 Lexer.l 中，定义"complex"，"PrintComp" 的关键字匹配，image (虚部) 的匹配方式，以及 imageconst 的处理方式，随后在 Parser.y 补充对应的终结符以及相关的一元操作符号，最后在 Tree.java 中添加相关常数以及处理函数。

1.2.2 Case

首先自然在 Lexer.l 中，定义关键字"case"，"default" 然后在 Parser.y 中实现"ACaseExprlist"，"DefaultExpr" 的非终结符的定义以及对应的处理方式，然后在 Tree.java 实现 Expr 的一个新的子类，重载打印函数，实现 Case 语句的输出。此外修改了 SemValue，实现对新子类的信息储存方便 Parser.y 进行处理。

1.2.3 Super

首先关于"Super" 的使用方法我个人觉得应该更像 Call 的使用方式，只是 Super 前面不再是一个可变的 Receiver 而是一个固定的 Super，所以还将 Super 还添加进了 SimpleStmt。

*清华大学计算机系，学号：2015011391，邮编：100084


```

        1:2+3;
        1+2j:3+4;
        3:4+5;
        default:0;
    };
}
}

```

输出结果如下- 检测了 Case 对整数常量的唯一性

```

*** Error at (7,15): syntax error
my-case-test.decaf ERROR!

```

2 总结

本次试验是后面实验的基础，不仅需要认真完成，而且需要借此机会熟悉框架的运行机理，通过此次试验，切实的掌握了 Lex,Yacc 的使用，也复习了不少自动机的知识。