## 编译原理PA1-A实验报告

2015011287 方闽杰

本次实验的主要内容是在已有的编译器框架上，利用JFlex和Yacc工具，完成Decaf编译器的词法分析和语法分析部分。实验最大的难点在于上手比较慢，需要弄懂lexer.l和parser.y脚本使用的语法规则，以及各个java文件中需要修改的部分。熟悉了流程以后，实验过程就会轻松很多。

我在添加新的语言特性时，基本上按照如下的过程进行：

1. 在lexer.l中添加相应的规则，来识别新加入的关键字与操作符
2. 如果有需要，在BaseLexer.java中添加相应方法（本次实验中，添加复数虚部识别时加入了imgConst()方法）
3. 在parser.y中对新加入的关键字和操作符进行声明，添加新的语法规则，在每条识别规则后面留出大括号包围的空间，留给后面添加动作使用
4. 在Tree.java中，根据语法识别的需要，加入和parser.y中加入的非终结符对应的类，完成相应的方法
5. 根据Tree中新增加的类，完成parser.y中留白的动作，并且根据需要修改SemValue的内容
6. 运行调试

实验过程中，遇到的一些问题和注意点如下：

1. 有时候加入新的语法特性后运行decaf.jar会出现Syntax error的错误，一般是因为没有进行clean操作。Clean之后重新build可以解决。
2. 加入新的语法特性时，在lexer.l中只加入complex、case等关键字并不够，还需要将必要的操作符（如“:”）也添加进去，才能够正确进行词法分析。
3. 在lexer.l中新加入的内容在parser.y中也应当有相应的声明。实验过程中，我发现parser.y中的所有终结符都必须事先声明。
4. 在parser.y的声明部分中，也应当加入适当的限制。比如%left说明该操作符是做结合的，%nonassoc则说明该操作符不能够连续出现。
5. 在lexer.l中，添加了识别复数虚部常数的规则时，应该在baselexer.java中为它提供相应的方法，并且在Semvalue的输出当中也应当有相应的特别处理。
6. 应当注意parser.y中添加规则时，要避免导致冲突。可以在parser.output中查看是否出现冲突。