

# 编译原理 PA2 作业报告

黄家晖 2014011330

PA2 主要实现静态语义检查，包括符号表的建立和类型的检查，为 decaf 编译器增加新的语法特性。

## 1 新增加的数据结构和函数

1. 在 Error 中新增加了错误类型，包括 Continue 语句不在循环体内的错误、选择三元运算符的算子不符合要求的错误、Case 的标签不是整形常量错误等；
2. 在 BuildSym 的 Visitor 类中添加访问 repeat 语句和 switch, case 语句的符号表建立方法，对于 switch 来说，其作用域直接和外层相同（根据 new\_test\_q3.decaf 文件的标准输出推断可知），而对于 repeat 语句，作用域为本地的新建作用域（参照 C 的语言规范）；
3. 在 TypeCheck 的 Visitor 类中添加三元运算符的语义检查、continue 的语义检查以及二元运算符 << 的语义检查。

## 2 遇到的问题 and 解决方法

对于建立符号表的过程，注意到实验文档中给出 repeat 的参考文法是 `repeat Stmt* until (BoolExpr)`，而只有 Block 才能关联本地作用域，因此采取一个巧妙的做法，在建立 repeat 抽象语法树的时候就将 Stmt\* 放在 Block 当中，实际上为 AST 嵌套了一层 Block 节点，这样一来，repeat 内的所有语句也能像 if 那样被看做一个单独的 Scope 进行了处理。

对于类型检查的过程，逻辑比较复杂的是三元运算符部分，代码中我尽量使用已经定义好的函数。首先判断条件语句是否为 Bool 型，如果不是则报错，但不影响三元运算符语法节点的属性文法；接着看 ? 后面的选择语句，如果二者是 compatible 的（这个函数考虑了更多的情况），且都不是错误类型，那么返回左边的属性，如果有一个是错误类型，属性文法就整体为错误；如果二者不能 compatible，那么输出报错，属性文法为错误。

另外对二元运算符 << 计算最小公共父类采用最简单的  $O(n^2)$  进行查找，详情请参见函数 `TypeCheck.getMinCommonAncestor()`。