

Decaf 实验 PA1 实验报告

计 35 吴之璟 2013011392

一. 实验内容

对 decaf 语言增加新的语言特性，完成对新语言特性的词法语法分析，以及相关的语法规书构造。

二. 基本思路

1. ++, --运算符

- a) 使用语法: $\text{Expr} ::= \text{Expr} ++ \mid ++ \text{Expr} \mid \text{Expr} -- \mid -- \text{Expr}$
- b) 实现: 在 Parser.y 的 Expr 后面添加上述四种情况的对应语法, 在 Tree.java 中的 Unary 类的方法 printTo 中通过 tag 区分四种情况从而做出不同的输出。

2. 三元运算符

- a) 使用语法: $\text{Expr} ::= \text{Expr} ? \text{Expr} : \text{Expr}$
- b) 实现: 同样在 Parser.y 的 Expr 中添加上述语法, 在 Tree.java 中新增继承 Expr 的类 Ternary 来处理。

3. 反射运算

- a) 使用语法: $\text{Expr} ::= \text{numinstances}(\text{identifier})$
- b) 实现: 同样在 Parser.y 的 Expr 中添加上述语法, 在 Tree.java 中新增继承 Expr 的类 Numinstances 来处理。

4. 串行条件卫士语句

- a) 使用语法:
 $\text{GuardedStmts} ::= \text{GuardedStmts} \mid \mid \text{Expr} : \text{Stmt} \mid \text{Expr} : \text{Stmt}$
 $\text{Stmt} ::= \text{GuardedIfStmt}$
 $\text{GuardedIfStmt} ::= \text{if } \text{GuardedStmts} \text{ fi}$
- b) 实现: 在 Parser.y 的 Stmt 后添加对应语法, 并新加 GuardedStmts 和 GuardedIfStmt 两类, 同时在 Tree.java 内新增继承 Tree 的类 GuardedIf 来处理。

5. 串行循环卫士语句

- a) 使用语法:
 $\text{GuardedStmts} ::= \text{GuardedStmts} \mid \mid \text{Expr} : \text{Stmt} \mid \text{Expr} : \text{Stmt}$
 $\text{Stmt} ::= \text{GuardedDoStmt}$
 $\text{GuardedDoStmt} ::= \text{do } \text{GuardedStmts} \text{ od}$
- b) 实现: 在 Parser.y 的 Stmt 后添加对应语法, 并新加 GuardedIfStmt, 同时在 Tree.java 内新增继承 Tree 的类 GuardedDo 来处理。

三. 遇到的问题

- 1. 最初在增加自加自减的语言特性时, 使用了老师给定的参考语法 $\text{identifier}++$ 和 $++\text{identifier}$, 但在把语法规则转换为语法分析程序的过程中存在冲突, 无法消除。然后把 identifier 全部使用 Expr 代替, 虽然解决了冲突, 但可能会造成错误地正确识别出一些不符合规范的语句, 比如 $(a+b)++$ 。
- 2. 在输出对应的语法分析树时要先观察已给定的 result 文件夹中对应文件的内容, 之后才能确定自己要输出的文字信息。