

# Decaf PA2 Report

王琛 计65 2016011360

## 一、特性的增加

### 1. 类的浅复制

- `TransPass.java` : 增加了 `visitScopy` 函数, 访问源class和目的class, 获得size。然后进行拷贝, 首先分配size的新内存空间, 然后调用`genAssign`赋值给目的class (不用`genAssign`目的class拷贝会失败, 无法实现浅复制)。最后逐个字将源class赋值给目的class。

### 3. 串行条件卫士语句

- `TransPass2.java` :
  - 增加 `visitIfSub` 函数: 对串行条件卫士单条语句的处理。首先访问Expr, 其次生成条件转移的三地址码, 然后是条件为真的部分, 直接accept即可, 最后调用`genMark` (条件为假转移到的地方)。

### 4. 简单的类型推导

- `Tree.java` : 在Lvalue的Kind的中新增了AUTO\_VAR类型, 用于自动推导的情况
- `TypeCheck.java` : 在visitIdent中判断为var时, 将lvKind设置为AUTO\_VAR
- `TransPass2.java` : 在visitAssign函数中新增了对AUTO\_VAR情况的处理。首先设置symbol对应的Temp对象, 然后和LOCAL\_VAR相同, 将`assign.left` (实际上是Tree.Ident类型)的Temp和`assign.expr.val`作为参数调用`genAssign`生成三地址码

### 5. 若干与一维数组有关的表达式或语句

#### (1) 数组初始化常量表达式

- `TransPass2.java` : 增加了`visitArrayRepeat`函数, 如果不是class类型, 调用`genNewArray`生成, 如果是class类型则调用`genNewClassArray`函数
- `Translator.java` :
  - 修改了 `genCheckNewArraySize`, 添加了参数int opt。如果opt为1则报数组初始化n取值小于0的错, 否则和之前一样报NEGATIVE\_ARRAY\_SIZE的错、
  - 修改了 `genNewArray` 函数: 添加了Temp value和int opt参数, 其中value用来是数组初始化的值, 正常情况下为0, 当适用E%n时值为E的值, opt含义和上面的`genCheckNewArraySize`相同。

- 增加了 `genNewClassArray` 函数，和 `genNewArray` 类似，首先分配数组空间，填入长度，然后每次循环分配class类型的空间，将数组中相应位置指向该空间，并且进行类的浅拷贝。

## (2) 数组动态下标访问表达式

- `TransPass2.java`：增加了 `visitDefault` 函数，对数组的下标进行判断，首先判断是否小于0，然后判断是否大于length，如果没有越界，使用 `genLoad` 得到相应的值，否则使用默认值。

## (3) 数组迭代语句

- `TransPass2.java`：增加了 `visitForeach` 函数，取出数组长度，进行循环，循环变量每次加1直到等于数组长度为止。如果有while部分直接调用相应的 `accept`，如果while为真调用相应的 `accept` 语句块。至于break的支持，仿照 `visitForLoop` 设置 `loopExits` 就可以了。
- `BuildSym.java` 和 `TypeCheck.java` 修正了PA2的一些错误

## “除零非法”检测

`Translator.java`：

- 增加了 `genCheckDivByZero`，参数为除数，如果除数为0，则报错并且停机，否则直接退出。（写法和 `genCheckNewArraySize` 基本相同）。
- 在 `genMul` 和 `genMod` 调用上述函数进行判断。

## Acknowledgement

本次PA中，柳瑞阳同学和刘应天同学给予了我代码上的指导。