**PA3实验报告**

2015010207 计65 钱姿

**实验内容：**

本阶段的主要工作是在PA2的基础上，对输入程序进行翻译，把使用带属性修饰的抽象语法树来表示的输入程序翻译成适合后期处理的另一种中间表示方式。

**实验任务以及主要实现：**

**1、类的浅复制的支持**

**修改文件 TransPass2.java：**

·重写函数visitScopy, 首先获得需要拷贝的对象所占空间大小后，调用Intrinsic.ALLOCATE函数生成新的空间，然后把需要拷贝的对象的每一项都复制到新生成的空间里，再把虚表地址拷贝过来，最后将拷贝完成的对象的指针赋值给目的对象。

**2、sealed的支持**

不需要任何修改

**3、串行条件卫士语句**

**修改文件TransPass2.java:**

·重写函数visitGuardedStmt,建立若干标签，主要是能够成功跳转各个分支，并在建立标签后，对条件语句进行判断，如果成功则跳转执行语句进行访问，重复以上步骤依次访问所有的语句块。

4**、支持简单的类型推导**

**修改文件TypeCheck.java**

·修改visitAssign函数，增加了对lvalue的成员变量lvKind的赋值，将其赋值为LOCAL\_VAR

**修改文件TransPass2.java**

* 修改了visitAssign函数，增加了当lvKind为LOCAL\_VAR类型时的类型判断，如果是Var相关语句，则进行强制转换为VarStmt类型。

·重写了visitVarStmt函数，新建了Temp，并将其于varstmt的symbol中的temp绑定。

**5、支持数组初始化常量表达式**

**修改文件TransPass2.java**

·重写visitInitArray函数，对赋值的数组元素进行判断，如果是Class类型，那么就需要调用针对初始化class类型的函数，进行浅拷贝，而不能仅仅赋值指针。否则调用普通的初始化函数

**修改文件Translater.java**

·增加函数genNewInitArray，通过标签建立一个赋值循环，首先申请一块以4为单位的length为长度的空间，随后通过循环为空间每个值赋值为想要设置的初始值，参考原代码中的genNewArray函数

·增加函数genNewInitClassArray,与上一个函数最大的区别是，在赋值过程中采用了和浅拷贝一样的方式，而不是单纯的拷贝指针。

**修改文件RuntmeError.java**

·增加错误NEGATIVE\_ARR\_INIT\_SIZE，如果初始化的时候n不合法，则报该错误。

**6、支持数组下标动态访问表达式**

**修改文件TransPass2.java**

·重写函数visitDefault,在该函数中，首先对数组和数组index进行访问，随后对访问是否合法进行判断，如果超过了数组长度或者小于0，那么都视为不合法，将跳到default语句，此处通过建立标签来实现。否则将数组中的指定数取出来，并将Default的val值设置为该值

**7、支持数组迭代语句**

**修改文件TransPass2.java**

·重写visitForeachStmt函数，首先建立一个循环变量和一个用来往上累加的值为1的temp，随后对数组名和Type x(Var x)进行访问，并建立一个exit标签，并将其推入loopExits,实现break语句，随后建立循环标签，对循环是否合法进行判断，如果合法则对语句块进行访问。

·重写visitNormalStmt函数，新建了Temp，并将其于varstmt的symbol中的temp绑定。

**修改文件TypeCheck.java**

·修改visitForeachStmt函数，弥补PA2的弥天大锅\_(:з)∠)\_。首先增加了对Bool语句块是否为空对判断。调换了foreach之前的语句和Bool语句块的访问顺序。

**8、支持“除零”非法**

**修改文件RuntmeError.java**

·增加错误DIVIDED\_BY\_ZERO，存储被零除的错误信息

**修改文件TransPass2.java**

·修改visitBinary函数，在除法和取模运算时首先调用checkIfZero函数

·增加checkIfZero函数，在TAC中添加一个条件判断，如果相应结果是0，则输出错误信息，并且调用Intrinsic.HALT函数