**Pa3 实验报告**

**一、实验要求**

在本阶段，我们将对该输入程序进行翻译，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，即TAC形式。

**二、实验实现**

首先将 PA2 中frontend, tree.java, typecheck的部分拷贝过来，对于不同的功能需求，具体的改变如下：

* 类的浅复制:

修改Transpass1.java文件，仿照vardef，对classdef对象增加vars，是一个list变量，用来记录classdef对象中的变量个数。

修改Translater.java 文件，增加了 genScopy 函数。首先分配一块相应对象的大小的空间，其中还包含了虚表的空间。之后逐一将每一个变量复制到res里即可，最后还要讲虚表复制到res中。

修改Transpass2.java文件，增加visitScopy函数，调用genScopy，将scopy中的ident类型的第一个变量绑定上相应的Temp。

* 串行条件卫士:

修改Transpass2.java，对于Guard类的四个构造函数分类讨论，如果if语句块内只有一个E|||S，则调用genBeqz和genMark函数，写相应逻辑的汇编代码，判定E的真假，branch到相应的位置。其他三个构造函数则分别对于链表中的元素分别进行visit。

* 简单类型推导:

修改Tree.java文件，LValue的Kind中增加UNKNOWN\_VAR变量。

修改Transpass2.java文件，对其中的visitIdent函数的修改是参照visitVarDef函数，对UNKNOWN\_VAR绑定一个Temp变量。对visitAssign函数的修改是增加case UNKNOWN\_VAR，用genAssign函数使得var的赋值可以正确执行，并正确改变var的类型。

修改typecheck.java文件的visitIdent函数，对ident对象的lvKind赋值为UNKNOWN\_VAR。

* 数组操作:  
  初始化常量表达式:

修改Transpass2.java文件，新增 visitArrayRepeat 函数，首先对Expr和intconst进行visit操作，调用genCheckgenCheckRepeatRepeatLessThanZero函数进行intconst的正负检查，通过检查，则调用genNewRepeatArray函数产生相应的汇编代码，最终得到arrayRepeat对象的val值。

修改Translater.java文件，新增genCheckgenCheckRepeatRepeatLessThanZero函数，此函数仿照给出的checkArrayIndex函数，对下标进行检查。新增genNewRepeatArray函数，仿照genNewArray函数进行汇编代码的生成，对示例代码的改动在于对基本类型和Class类型的分别判定，如果是class类型，则需要调用浅复制的genScopy函数获得新对象的值store进arrayRepeat对象的val中。普通类型直接store即可。

下标动态访问表达式:

修改 Transpass2.java 文件，新增 visitDynamicAccess 函数，对于表达式中的三个expr分别进行accept之后的分析工作。分别得到三个expr的Temp对象，调用genCheckgenCheckArrayIndex（重载后的函数），得到dynamicAccess对象的val值。

修改Translater.java，对genCheckgenCheckArrayIndex函数重载，此函数仿照同名函数进行改写，增加了下标合法时的Label的汇编代码块，下标不合法的Label和退出Label仍然沿用同名函数的写法。将对应下标的数组的元素的值取出赋值给目标节点，这部分的代码仿照visitIndexed函数对于数组的处理的方式进行改写；如果下标不合法，则直接返回default的表达式的值。

数组迭代语句：

修改 Transpass2.java 文件，新增 visitForeach 函数，分while是否存在的情况进行分类讨论，分别调用重载了的函数genForeach进行汇编代码的产生。新增genForeach函数，仿照forloop函数，按照foreach的语法进行汇编代码的产生。需要得到array的length，array的index一开始设置为0，之后递增1。首先产生loop的标号，紧接着判定index是否满足数组下标范围，如果不满足则直接调到exit。之后对x进行赋值，将数组中的元素取出来的值作为setTemp函数的参数绑定到x上。再进行while块的处理，需要注意的是每次bool都需要accept一次（因为变量会不断地变化）。再进行S中每一个元素的accept。由于在pa2中这些元素已经被加入到同一个作用域，用list实现，所以直接循环accept即可。最后进行index的递增和exit标号的产生，汇编代码到此为止就结束了。

修改TypeCheck.java，更正了pa2中没有while块的bug，同时新增了symbol的赋值，如果是新建的variable则直接赋值，如果是已有的variable，则lookup出来。比较重要的改动是在symbol赋值之后再进行while块的accept，因为symbol赋值之前如果while块中含有var的变量的话，会出现UNKNOWN类型未经改变的问题，所以while块的accept要放在后面一些。进行的修改还包括仿照forloop写的break语句，这也是在pa2里漏写的。