PA3 实验报告

计 76 沈诣博 2017011427

本阶段的工作分为如下几部分：
主要部分：修改了提供的 translate 部分和 error 部分代码，增添了对于新特性的支持。
其他部分：在 PA2 基础上针对两次 PA 特性的不一致 (foreach 语句) 部分进行了修改。
首先说说主要部分。我修改了 translate 部分的代码，实现了浅复制，卫士语句，变量类型推导，数组特性和除 0 报错的实现。

1. 浅复制

浅复制需要如下几步：首先，注意到 Scope 类里有一个包含域的所有成员标识的 Map<String, Symbol> symbols，在需要知晓一个类的全部成员时可以调用 symbols.variables() 进行遍历。
然后可以将浅复制的过程分为以下部分：
1. 算出新建该类对象的空间并进行申请：

一个类对象实体需要的空间大小是 4× 类成员的总数，因此在第一次遍历类成员的时候算出该实体的空间，并且调用 alloc 库函数进行申请。
2. 找到每一个成员的地址并进行复制：

由于在 Transpass1 类里已经对于对于任意一个类的成员的 Variable 标识赋予了指针偏置量，在浅复制的时候只需遍历他们进行偏置即可找到相应的寄存器。对于每一个寄存器内的内容进行复制。
3. 将申请空间的头指针地址给存进需要浅复制的对象所处的寄存器内。

2. 条件卫士语句

对条件卫士语句的每一条子语句进行遍历，对于子语句的遍历过程如下：
1. 生成一个“结束”标签，如果子语句条件部分为负就跳至标签处
2. 遍历执行语句，并将标签附在执行语句之后。

3. 变量类型推导

支持变量类型推导的修改比较简单：由于已经在第二阶段实现语法检测，直接对于相关的变量新增 tac 即可。

4. 支持一维数组

1. 对于数组初始化常量表达式，我使用了以下的方式生成中间代码：
首先，检查新增数组的大小是否小于等于 0，是则报错退出，否则申请空间并继续。
然后，检查数组元素的类型是否是 class, 是则调用浅复制函数，否则直接复制。
在本段代码的最后，生成一个退出标签；在复制元素之前，生成一个循环标签并从后到前进行数组的逐元素复制。在一轮复制之前进行判断，若需要复制的元素偏置小于 0，则跳到退出标签，否则复制完之后跳到循环标签。

2. 对于数组下标动态访问表达式，我使用了以下的方式生成中间代码：
对于返回的不同值，设置两个标签，标出两个不同的中间代码分支：
首先，检查下标是否合法：获取数组 E 的大小，并与下标进行比较，若下标小于 0 或者大于数组 E 的体积，则跳到第二个分支，否则跳到第一个分支。第一个分支返回数组相应元素，而第二个分支返回表达式的值。

3. 对于数组数组迭代语句，我采用了以下的方式生成中间代码：
首先，生成一个 loop 循环，在循环开始之前查看当前绑定的值下标是否在范围内，是则将迭代值 x 绑定为数组的相关元素，否则跳到 exit 标签。
然后，若有 while 条件判断语句，则遍历该语句，如果值为假则跳到 exit 标签。
最后，遍历代码块 S。

实现除 0 错误的识别

我对 translate 类的 genDiv() 和 genMod() 方法进行了如下的改变：
首先判断被除/模数是否为 0，是则报错退出，否则继续。

此外，对于 error 文件夹内的 RunTimeError 类，我增加了除 0 报错和数组初始化报错两条，从而支持示例类的具体报错语句。

对于次要部分进行的修改如下：

1. 将 PA2 中的 frontend,tree 以及 typecheck 复制进来，并且根据复制进来代码实现了 sealed 特性。
2. 在 tree.java 中，我在 Expr 子类里添加了标识中间码的 tac 变量，即对于原有的 tree.java 的新加特性进行 merge。
3. 修改了 foreach 语句的特性，使之支持条件语句缺省的情况，并且修复了 PA2 中未能支持 break 语句的 bug。