2015/12/12 沈哲言 计 34 2013011371

DECAF 实验阶段三（PA3）

1. 主翻译器 TRANSLATER 的修改

此文件修改的部分不多，主要增加了除零的运行时错误的检查 genCheckDivider。在阅读了数组下标越界和数组大小非法之后的处理代码之后，对错误处理的逻辑有了一定的了解。除零检查主要针对除法和取模运算，将被除数做零检查，即产生等于零的分支跳转语句，进入错误处理段（简单的输出错误信息，需要在 RuntimeError 类中增加相应的错误字符串）

2. 第一趟扫描 TRANSPASS1 的修改

这部分修改主要针对的是 numinstances 反射运算，首先在 Class 类中新增一个 Temp 类型的变量来记录每一个类的实例数量，然后在第一趟扫描遍历类声明的函数中对这个 Temp 类型变量初始化（即分配一个变量寄存器），因为代码框架不具备对于全局类声明的统计，故这里的变量寄存器只适用于当前函数作用域（实际上只要记录 main 函数内的声明即可）。

3. 第二趟扫描 TRANSPASS2 的修改

这部分修改主要针对新增的运算，增加了五个访问函数分别对应五个新的运算。其中自增自减运算主要仿照二元运算来实现，值得注意的是自增自减除了返回值以外还包含一个自加和自减的赋值，与二元运算稍有区别，另外前置和后置在返回值上也有细微差别；三元运算符主要仿照 If 语句来实现，设置判断条件为假的 Label，进行分支跳转；反射运算的实现方式是在 new 语句的遍历时对 Class 的实例数量进行加 1，然后在反射时直接返回这个 Temp 值；串行条件卫士和循环卫士的实现主要仿照 For 语句，对每一个语句块进行遍历并设置条件成立或者失败的 Label，最后统一设置出口标签 exit，需要注意的是串行循环卫士语句中可能包含 break 语句，故需要将出口标签 exit 加入堆栈 LoopExits 中以便跳出时的跳转。

4. 总结

通过第三阶段的实验，我对中间代码生成这一步骤有了更深入的理解。这一步中的TAC 码实际上已经和汇编代码非常相近了，整个编译器前端的工作也就到此结束了，至此，编译器已经能从分析阶段转入真正运行程序了。