

PA3 实验报告

计 63

孙亚男

2016011285

1、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。

2、实验实现

首先将前几次作业中的 frontend 和 typecheck 的部分拷贝过来，将自己修改过的文件中修改部分拷贝到 PA3 对应文件中，再将 Tree.java 中的相关部分整个在这次的文件中，具体内容在此不详细叙述。

对于不同的功能需求，具体的改变如下：

2.1 类的浅复制的支持

修改文件 Tree.java

- 在 SCopyExpr 类中添加 Variable 类型成员变量。
用于存储拷贝目标的 Temp。

修改文件 TypeCheck.java

- 修改 visitSCopyExpr 函数。
将 scopy (x, y) 中 x 的 Symbol 存入 SCopyExpr 类的 Variable 中。

修改文件 TransPass2.java

- 添加 visitSCopyExpr 函数。

通过 getSize 函数得到待拷贝类的大小，为实现浅拷贝，在 for 循环中依次将待拷贝类以一次 4 字节的规模使用 genLoad 函数取出，并使用 genStore 函数存入新建的 Temp 类对象中。最后使用 genLoadVTable 将该类的虚表取出并存入新建的 Temp 类对象中。使用 setTemp 函数将该 Temp 对象存入 SCopyExpr 的 Variable 的 Temp 中。

2.2 sealed 的支持

由于只涉及属性判断，该任务在此次 PA 中无需再做任何操作即可完成。

2.3 支持串行条件卫士

修改 TypeCheck.java

- 修复 PA2 遗留的问题：修改 visitIfSubStmt 函数。
将 IfSubStmt 类中的语句块 stmt 使用 accept 访问。

修改 TransPass2.java

- 添加 visitGuarded 函数。
依次创建开始标签、循环标签、结束标签和退出标签，其中循环标签为 Guarded 类中 subStmt 的大小+1。依次遍历条件卫士中的每一项的条件，若条件满足，则访问其后对应的语句块，否则跳转到下一个条件继续判断。直至所有条件均被遍历，则退出串行条件卫士
- 添加 visitIfSubStmt 函数。
将 IfSubStmt 类中的语句块 stmt 使用 accept 访问。

2.4 支持简单的类型推导

修改 BuildSym.java

- 修复 PA2 遗留的问题：修改 visitVar 函数。
从 table 中得到当前 Var 类对象的 Symbol 并赋给 Var 类对象中 VarDef 类成员变量的 Variable 类成员变量。

修改 TypeCheck.java

- 修复 PA2 遗留的问题：修改 visitVar 函数。
将 Var 类中的 Kind 类对象赋值为 Tree.LValue.Kind.LOCAL_VAR。
- 修改 visitAssign 函数。
当等式左侧表达式的类型为 UNKNOWN 时，将等式右侧对象的 Temp 类对象赋给等式左侧对象的 Temp 类对象。

修改 TransPass2.java

- 修改 visitAssign 函数。
当等式左侧的表达式的 Kind 为 LOCAL_VAR 时，根据左侧对象的类分类讨论，以确定在赋值时的不同强制类型转换类型。
- 添加 visitVar 函数。

新建一个空的 Temp 类对象；将 Var 类对象中 VarDef 类成员变量的 Variable 类成员变量赋给该对象的 Variable 类成员变量；将该 Temp 赋给 Var 类对象中 VarDef 类成员变量的 Variable 类成员变量的 Temp 类成员变量。

2.5 支持数组操作

2.5.1 数组初始化常量表达式

修改 TransPass2.java

- 添加 visitNewSameArray 函数。

根据传入参数的类型设置一个该类型对象的大小。由 readme 可知，在本次实验中，INT、BOOL 和 STRING 类型的对象均为 4 字节大小，而 CLASS 类型的对象大小通过 getSize 函数获得。

故分为两类分别传入参数调用 genNewSameArray 来生成所需数组。

修改 Translator.java

- 添加 genNewSameArray 函数。

仿照已有 genNewArray 的写法，计算所需空间，申请空间，生成循环为数组中成员依次赋值。其中，若为非 CLASS 类型，则可以通过 genStore 直接赋值，若为 CLASS 类型，则需要仿照任务一种 scopy 的浅拷贝方式进行赋值操作。

- 添加 genCheckNewSameArraySize 函数。

为支持检查 n 取值功能，仿照已有 genCheckNewArraySize 的写法，判断数组是否存在 n 的取值小于 0 的情况，若存在，则报运行时错误。

修改 RuntimeError.java

- 添加 NEGATIVE_SAME_ARR_SIZE 成员变量。

根据测例中对于 n 取值小于 0 的错误输出为该成员变量赋值。

2.5.2 数组下标动态访问表达式

修改 TransPass2.java

- 添加 visitDefaultArray 函数。

新建一个空的 Temp 类对象；仿照 genCheckArrayIndex 函数的写法对数组下标进行检查，若合法，则将下标对应对象的 Temp 赋给创建好的 Temp 类

对象，否则将 default 后的表达式的 Temp 赋给创建好的 Temp 类对象；将该 Temp 类对象赋给 DefaultArray 类的 Temp。

2.5.3 数组迭代语句

修改 Tree.java

- 修改 VarBind 类。
添加 Variable 类成员变量。

修改 BuildSym.java

- 修复 PA2 遗留的问题：修改 visitForeachArray 函数。
while 为空情况下，不应 accept 对应 expr，添加对 while 为空情况的特判；将 foreach 对应 Symbol 从声明到对应 scope 中改为声明到 table 中。

修改 TypeCheck.java

- 修复 PA2 遗留的问题：修改 visitForeachArray 函数。
将 Symbol 的 declare 位置提前至 while 对应 expr 调用 accept 函数之前；
增加对 while 为空情况的特判；将 VarBind 对应的 Symbol 赋给新添加的 Variable 类成员变量。

修改 TransPass2.java

- 添加 visitVarBind 函数。
仿照任务四对 Var 的处理方式。新建一个空的 Temp 类对象；将 VarBind 类对象中 Variable 类成员变量赋给该对象的 Variable 类成员变量；将该 Temp 赋给 VarBind 类对象中 Variable 类成员变量的 Temp 类成员变量。
- 添加 visitForeachArray 函数。
仿照 visitForLoop 的写法完成循环，初始条件为下标为 0，限制条件为下标小于数组大小且 while 语句为真（若为空，则一直为真），每次循环下标++；仿照 visitForLoop 的写法完成 break 的支持，在访问 foreach 循环体内语句块之前将结束标签压入栈中，访问结束后将结束标签弹出。

2.6 检测除零非法

修改文件 TransPass2.java

- 添加 checkZero 函数。
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。

- 修改 visitBinary 函数。

在除法和取余运算中先将右侧表达式作为参数调用 checkZero 函数。

3、实验总结

1. 本次实验的目的是生成近似于汇编的 tac 代码，故在实验中保持清醒的头脑，知道哪些语句是需要其出现在 tac 代码中的，哪些是单纯的为实现特定赋值等功能的 java 代码十分重要。在完成实验的过程中，我曾出现过希望 tac 中出现赋值语句，但未调用 genAssign 而是直接使用“=”的问题。

2. genAssign 与 genLoad 分别实现了对值和对相应地址的操作，要熟练区分，不然会增加调试的难度。

3. 本次实验调试难度较大，可以对照 tac 代码进行调试，在 tac 代码中添加特定寄存器的输出以便定位错误位置；同时还可以在生成 tac 代码时使用不同数量的标签定位语句块，便于调试。

4、实验体会

本次实验使我发现了许多 PA2 中存在的漏洞，体会到了这种环环相扣的系列实验中做好每一次实验的重要性，同时将之前学习过的汇编与编译有效结合，使得我对编译原理的了解更加深入。因为本次实验相对难度较大，所以在完成所有功能后，我收获了巨大的成就感！