

PA3实验报告

2017011307 张晨

实验原理

动态错误检查

在 `visitBinary` 里，如果当前运算是除法或取模，则加入“如果除数是0，则报错”的逻辑。代码与“数组大小非法”极为类似。

抽象类

因为在PA2里已经做过相关检查，所以在PA3里可以将逻辑简单的写为“如果有函数体，才遍历它”，不需要判断是否是抽象的。

局部类型推导

正如实验指导书所说，不用做任何事情。

扩展call

与调用非静态成员函数基本相同，最大的区别是前四个字节直接是函数指针，而不是VTable，故可以直接访存得到要call的地址。在生成这部分TAC之前，需要先遍历call对象不包含括号的部分。

获取新建函数的函数指针

我将所有新建的函数写到同一个虚表中，函数命名规则为“对象的名字 + '_' + 编号”，其中编号是一个每次加一的数，它保证了所有函数的名字不同。在 `ProgramWriter` 初始化时，便通过声明一个类的方式声明这个虚表。在 `ctx` 里用一个 `LinkedHashSet` 存所有新建函数的函数名。在遍历完所有内容后，定义这个类，将所有新建函数视为累的非静态成员函数的类。使用 `LinkedHashSet` 是为了保证虚表中函数的顺序与插入顺序相同，从而在向 `LinkedHashSet` 里插入新的函数时，即可通过size知道其在虚表中的偏移量，方便代码的生成。

将方法名直接当做函数使用

在 `visitVarSel` 中，`symbol` 是 `methodSymbol` 时进行如下处理。

- 非静态函数
 1. 新建一个函数，获取其 `FuncVisitor`，内容为从第一个参数偏移量为4的位置取出该方法对应的成员对象，作为第一个变量，将调用该新建函数的所有变量依次作为之后的变量，调用 `visitMemberCall`，`visitMemberCall` 会通过成员变量的虚表获取相应的函数指针。最后调用 `visitReturn`
 2. 原有位置转成的TAC代码的功能为，申请8字节内存，前四字节存函数指针，函数指针从“新建函数”的虚表的特定位置取出。后四字节存成员对象。PA2中已经处理了缺省this的情况，故这里的成员对象就是 `receiver` 的 `temp`。
- 静态函数

1. 新建一个函数，获取其 `FuncVisitor`，内容为将调用它的所有参数（除去第一个参数）作为参数，调用 `visitStaticCall`。之后调用 `visitReturn`。
2. 原有位置转成的TAC代码的功能为，申请4字节内存，存函数指针，函数指针从“新建函数”的虚表的特定位置取出。

Lambda表达式

- 获取捕获变量。在 `Typer` 中，为每一个 `Scope` (不限于 `LambdaScope`) 维护它依照捕获的规则捕获的变量，判断方法为只要不是这个 `Scope` 自身定义的变量，就是捕获的。在内层 `Scope` 关闭时，将其捕获的变量中不是其父 `Scope` 定义的全部加入父 `Scope` 的捕获变量中。为保证捕获的变量没有owner，仅在对象没有receiver的时候进行捕获操作，若该对象需要增加this，则捕获this，否则捕获自己。
- 新生成的函数。其功能为首先通过访存读出所有捕获的变量，存到相应的`symbol.temp`里。之后和 `visitExpr/visitBlock` 就没有区别了。在处理新生成的函数之前，需要存下被捕获的变量原来的`symbol.temp`，并在处理完新生成的函数之后恢复。
- 原本位置。申请 $(1 + \text{被捕获的变量数目} * 4)$ 的内存，偏移量0处存函数指针，函数指针从“新建函数”的虚表中获得。之后的位置依次存所有的被捕获变量。如果被捕获的是this，要通过下面“获取this”的方法获取Temp并存下来，如果是局部变量，直接存它的Temp。
- 获取this。在每个 `FuncVisitor` 里，显式存第一个参数是否是this，以及不是this时，this相对于第一个参数的偏移量。因为this相对于第一个参数的偏移量一定不是0，所以可以将偏移量为0视为第一个参数就是this，从而在 `FuncVisitor` 里只需要存偏移量。`FuncVisitor` 的默认偏移量是0，当在lambda中且捕获了this时，将其设为相应偏移量。

其他

在PA2中，我为每一个Expr及其子类都分配了一个symbol。所以在PA3中，没有使用 `Expr.val` 作为temp，而是使用 `Expr.symbol.temp` 作为temp。为支持这种操作，我也将 `temp` 从 `VarSymbol` 类移到了 `Symbol` 类里（以减少工作量）。

遇到的困难

主要是修PA2的bug。由于输出的信息有限，这些bug不会导致PA2输出的内容出错。

1. `Symbol` 的 `type` 是常量类型，故推导出来之后也无法修改。我当时的做法是新建一个 `Symbol`，并在查找表中更新，这足以完成PA2的要求，但实际上需要更新的 `Symbol` 还有很多。这导致逻辑上的同一个 `Symbol` 在实际并不是一个，故它们的 `temp` 无法共享。我的修改方法是将 `Symbol` 改为非常量类型。
2. 没有完全做好为成员变量增加this的操作。将其完善了。