PA3实验报告

---计55 赵嘉霖 2015011344

1. 实验综述：

本次实验需要增加的代码其实并不是很多，难点在于需要结合PA1和PA2，对于抽象语法树、作用域、符号、类型、动态错误等深刻理解其中的逻辑关系。而且对于这五种新增的语法特性，不同于之前实验的开头比较困难，后面只是做一些重复性的工作，本次实验中除了case表达式和串行循环卫士有一定的相似度以外，其他特性都很难相互借鉴。

1. 整复数类型的支持：
2. 复数的内存表示参考了数组，采用动态分配64位的方法，前32位为实部，后32位为虚部，整个复数用指向第一个元素的指针表示。
3. 在Translater中加入genComplexImm4函数，功能与genLoadImm4相似，作用为**加载复数常数**，同时需要在TransPass2中修改visitLiteral，当虚数时调用genComplexImm4。
4. 在Translater中加入genToComp函数，功能为将int转化为复数，实现了**#表达式**，为了更方便的实现复数的加法和乘法运算，genToComp增加了type参数，如果type是int，则转化为复数，如果type是complex，则直接返回。
5. 在TransPass2中修改了visitBinary中的PLUS和MUL，在Translater中加入了genAddComp和genMulComp，当左右不同时为int时，利用genToComp函数将它们转化为复数，再通过genAddComp和genMulComp进行**复数加、乘法运算**。
6. 在Translater中加入genRe和genIm，来将虚数转化为实部的整数、虚部的整数，实现了**@、$表达式**，同时还需要修改TransPass2中的visitUnary函数。
7. 在TransPass2中加入visitPrintComp函数，实现方法参考visitPrint，实现了**复数打印**功能。
8. case表达式的支持：
9. 在TransPass2中加入visitCaseExpr函数，实现方法参考visitIf，每一个（常量，表达式）都相当于一个if语句。
10. 支持super表达式：
11. 在TransPass2中加入visitSuperExpr函数，实现方法与visitThisExpr完全一样，也返回currentThis。
12. 核心是在TransPass2中修改visitCallExpr函数，这部分修改分为三个阶段。
    1. 第一次修改加入了对于receiver是不是super的判断，如果是super则在调用tr.genIndirectCall时传入的参数不是当前对象的虚表，而是当前对象虚表的第一项，也就是当前对象的父类的虚表。这样修改后可以通过前两个test。
    2. 第二次修改是在receiver是super时，直接修改当前对象的虚表指针。方法为先将当前对象的虚表指针存下来，然后将当前对象的虚表指针修改为父类的虚表指针，在结束函数调用之后将当前对象的虚表指针还原。这样修改后可以通过前三个test。
    3. 第三次修改是因为发现不仅仅需要在receiver是super时修改虚表指针，还需要判断当前的ClassScope是否有这个函数，也就是要判断这个函数有没有在这个类重载，如果没有，则相当于再做一次super，直到存在。这样修改后可以通过全部四个test。
13. 支持对象复制：
14. 在TransPass2中加入了visitSCopyExpr函数，方法为先仿照visitNewClass函数动态分配一个新的地址，再仿照Translater中的genNewForClass计算出class的size，并对每个32位进行load和store。
15. 在TransPass2中加入了visitDCopyExpr函数，由于DCopy需要递归，所以在TransPass2中加入了DCopyIter来递归。函数DCopyIter需要输入Temp类型的DCopy源，和源的符号。参考Class类中的resolverFieldOrder实现分析当前对象的每个成员的类型，对于classtype的成员递归，非classtype的成员通SCopy复制。
16. 支持串行循环卫士语句：
17. 在TransPass2中加入了visitDo函数，实现方法类似visitWhileLoop，也类似case，只需注意加入loopExits的入栈出栈实现break功能。
18. 总结：

总体来说super功能实现最难，需要对于对象的内存、虚函数表的表示有深刻的理解。Case和串行最简单，都不需要修改Translater，只需要利用现有的Translater中的函数，也不需要了解内存模式。复数的难点在于需要通过动态分配内存的方式来储存复数。Copy的难点在于需要结合pa2中的classScope、type、symbol等模块，获取对于当前对象的成员的类型。