

编译原理实验报告

PA2

姓名：谢兴宇

学号：2017011326

2019 年 12 月

1 工作简述

在本次实验中，我从单独框架切换到了整体框架。

首先合并了 PA1-A 的工作，添加了对于抽象类的支持。

为了支持局部类型推导，我添加了一个新的特殊的Type: `VarType`，其对应的TypeLit为`TVar`，来将对以`var`为类型关键字声明的局部变量的类型推导延迟到第二趟遍历对其初始化表达式的类型推导之后。

为了支持 lambda 表达式，我定义了 lambda 表达式需要的`LambdaScope`和`LambdaSymbol`。为了创建 lambda 表达式的作用域和符号，我在第一趟遍历中加入了对`LValue`和`Expr`的解析。当有 lambda 表达式存在时，可能有多个局部变量都处于正在初始化的过程中，为了正确处理对这些符号的查找，我在遍历时额外维护了所有当前正在初始化的符号。

除此之外，还有诸多细节上的处理。

2 问题回答

1. 实验框架中是如何实现根据符号名在作用域中查找该符号的？在符号定义和符号引用时的查找有何不同？

通过当前的作用域栈中，自栈顶到栈底在每一个作用域中查找符号，直到找到为止。对于非类域，仅查找位置在当前被查找符号位置之前的符号。

符号定义时的查找在第一趟遍历;符号引用时的查找在第二趟遍历,且若被查找符号在一个初始化表达式中,对于非类作用域,需要查找出现位置在正在声明的符号之前。

2. 对 AST 的两趟遍历分别做了什么事? 分别确定了哪些节点的类型?

第一趟遍历创建了符号表和作用域,第二趟遍历做了类型检查。

第一遍确定了定义的类、方法、成员变量和局部变量的类型;第二遍确定了剩余所有节点的类型。

3. 在遍历 AST 时,是如何实现对不同类型的 AST 节点分发相应的处理函数的? 请简要分析。

每一种 AST 节点是某一个`trait`的`case class`,这样我们便可以用一个 pattern matching 来处理一个`trait`。