李嬁珩 2017050025

编译原理 PA2 实验报告

2019年11月20日

实验报告

编译原理 PA2

一、特性实现方法

1、增加抽象类

关于添加抽象方法，在 ClassSymbol 和 MethodSymbol 中添加判断抽象属性的方法 isAbstract（），另外注意 abstract 的函数体为空，需要跳过分析，避免空指针错误。

关于关键字打印，ClassSymbol 中需要手动判断当前方法是否为 Abstrct ；MethodSymbol 中已经有了 Modifiers ，故不需要额外操作。

关于错误处理，需要处理以下四点：

- Main 类不能为抽象；
- 不允许抽象方法重写基类方法；
- 类中含有抽象方法必须声明为抽象类；
- 抽象类不能实例化；

其中第三个错误处理我采用的是抽象函数表的方法，在 Namer 中解析 ClassScope 前初始化为父类抽象函数表（无父类则为空），解析 ClassScope 结束根据当前类的关键字和表是否为空来判断是否出现错误。

2、Var类型推导

查看树中的节点，如果 typeLit 为空的话，说明当前正在处理 Var 类型变量。我的处理方法是在 Namer 阶段中建立 type 为 BuildInType.WAIT（自己添加的类型） 的 VarSymbol ，在 Typer 阶段根据右边表达式类型进行赋值。

3、函数类型

本次框架中已经包含 FunType 类型， 根据 PA1-A 阶段在 Tree 中建立的 TLambda 类中的成员变量，在 TypeLitVisited 中重写 visitTLambda 函数即可。这部分要处理函数参数为 Void 的错误。

4、Lambda 函数解析

在此阶段，我新建了 LambdaFormalScope 类和 LambdaSymbol 类，仿照已有类实现，作为 Lambda 表达式的作用域和符号。处理的主要流程为：建立 LambdaSymbol ，打开 LambdaFormalScope， 处理参数， 打开LocalScope，处理内部 Stmt，关闭 LocalScope ，关闭 LambdaFormalScope 。

处理内部 Stmt 根据 Lambda 表达式为 Expr 和 Block 分别处理。

Expr 由于只有一个表达式，需要手动开关 LocalScope，Block 处理时会自处理 LocalScope 相关事情。

以上部分均在 Namer 阶段进行。有关返回值推导则放在 Typer 阶段。

此阶段还需增添许多新的 visit 函数，以避免 Typer 阶段找不到 Namer 阶段定义的 LambdaSymbol 。

关于作用域问题，我在 Typer 阶段的 visitAssign 函数中做了判断，借助 ctx 检查当前是否处于 Lambda 函数中，且被赋值的 VarSymbol 是否符合要求。

关于使用正在定义的符号的问题，可以检测符号类型是否为 BuiltInType.WAIT 类型即可。

5、Lambda 返回值推导

Expr 类型的 Lambda 返回值即为表达式的返回值类型。

Block 类型的 Lambda 返回值处理分为三步进行：

- 收集返回值
- 检测分支返回情况
- 分析返回值

对于每一个 Lambda ，采用一个 List<Type> 来收集每个语句的返回值。

对于 Lambda Block 中的每个语句，标记其是否返回。

求得返回值的上界作为 Lambda 的返回值。

其中要注意的点有，如果返回值列表为空，则返回 Void 。如果有返回值，且 Block 里有语句没有返回值，则需报错。求返回值上界时，需要对 null 进行特判。

6、函数赋值和函数调用

本部分均在 Typer 阶段中完成处理。

在 visitVarSel 中，添加对成员变量的支持、静态方法调用非静态方法的权限检查、通过类名访问变量的权限检查这三点，可满足用函数变量存储类的成员函数。

化简 visitCall 函数，在内部检查表达式类型是否为 FunType ，特殊处理 length 函数。

化简 typeCall 函数，在此只需要做参数类型检查即可。

二、回答问题

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?

程序维护了一个作用域栈，包含着当前时刻所有打开的作用域。在 Namer 和 Typer 遍历 AST 时，将这个栈 ctx 作为参数传递。寻找一个符号，只需要在栈上由顶至底逐层寻找即可。

定义时需要根据当前作用域的类型而采用不同的规则去栈中寻找冲突的符号名，引用时只要检查符号存在于栈中即可。

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?

第一次遍历定义变量符号，进行了类定义、类中成员变量定义、类中成员函数定义、Lambda 表达式的定义。确定了符号的名称、作用域、类型（var除外）。

第二次遍历检查符号的类型，进行 var 和 lambda 类型推断，检查符号的引用类型。

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。

采用访问者模式，将使用 visitor 类进行访问节点的动作。Namer 、 Typer 分别继承了该类并重写了其中的函数。visitor 中的函数参数类型不同，java 的动态多态机制可以使得在树节点在调用 accept（）函数时可以采用visitor（namer、typer）中不用的函数进行处理。