Compiler Lab3 Type and rename

201250104 苏致成

December 2022

1 目标

1.1 类型检查

1. 声明与定义：变量未声明、函数未定义、变量重复声明、函数重复定义。
2. 类型检查：赋值号两侧类型不匹配、运算符需求类型与提供类型不匹配、返回值类型不匹配、函数参数不适用。
3. 数组问题：对非数组使用下标运算符。
4. 函数调用：对变量使用函数调用。
5. 左值问题：赋值号左侧非变量或数组元素。

1.2 重命名

输入为行、列、新命名。对源文件中选中的 IDENT 及其同一作用域的 IDENT 进行重命名，输出重命名后的语法分析树。注意 IDENT 可能是定义的变量、使用的变量、函数参数等。

2 实现

2.1 使用工具

git、Antlr、Intellij idea、JDK11、Makefile

2.2 实现过程

2.2.1 设计类图

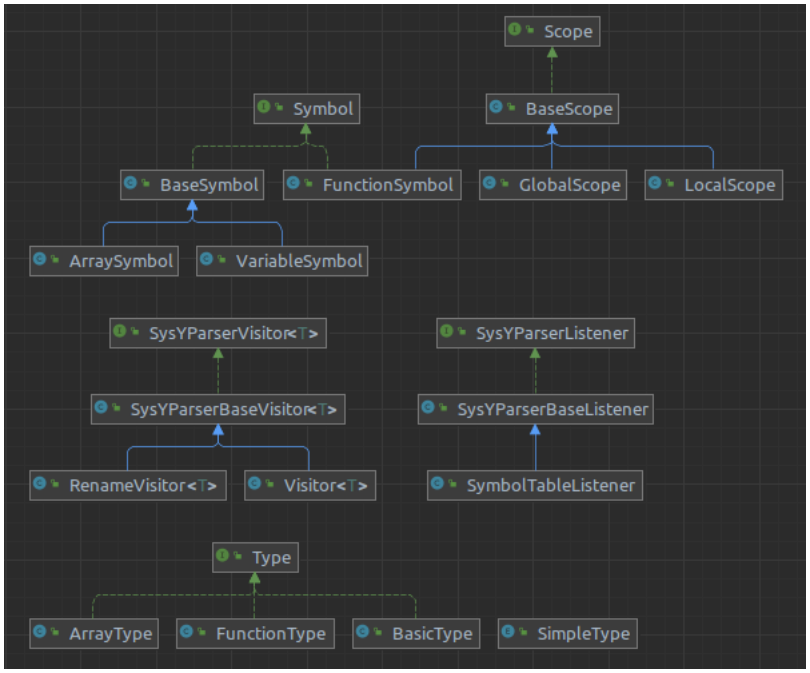

图 1: 类图设计

2.2.2 设计详述

1. Symbol 设计。Symbol 接口含有 BaseSymbol 和 FunctionSymbol 实现类。ArraySymbol 和 VariableSymbol 为 BaseSymbol 的子类。
2. Type 设计。Type 接口含有 ArrayType、FunctionType、BasicType 实现类。其中 BasicType 内含有枚举类 SimpleType 。
3. Scope 设计。Scope 接口含有 BaseScope 实现类，而 FunctionSymbol、GlobalSymbol、LocalSymbol 为 BaseScope 的子类。
4. 数组设计。ArraySymbol 含有 ArrayType 类型，ArrayType 含有维度、数量、子类型的属性，最内层为 BasicType，因此 ArrayType 的维度至少为一维。
5. 重命名设计。第一遍遍历时，若遇到当前 IDENT 行列为 main 中参数，则字符串记录当前作用域名字和 IDENT 名称。第二次遍历时，若遇到作用域和 IDENT 都完全相同的，则进行输出。
6. 节点传递参数。关于如何在函数之间传递参数的问题，使用泛型类可以解决。本 Lab 中需要传递类型的参数（如判断运算符两遍类型是否匹配）。

3 遇到障碍

3.1 局部变量定义

问题描述：如果是 vardecl，resolve 会导致无法命名与全局变量同名的局部变量。
解决方式：不使用 resolve()，而是使用 currentScope.getSymbol() 判断当前作用域是否存在该 Symbol。

3.2 return 语句无法得知函数定义的返回类型

解决方式：定义全局的 retType，因为不存在函数嵌套的情况，因此可以在 returnStmt 中获得；之后意识到可以获取当前作用域，因此可以将其强转为 FunctionSymbol ，获取其返回值类型。

3.3 函数和 block 重复

问题描述：按照原先的 SysYParser.g4 文件，发现在进入函数定义的 block 后会跳出 FunctionSymbol 这个作用域，进入 block 这个作用域。

解决方式：因为确定函数声明和定义一体，因此引入 next 变量，若进入 block 之前发现其前为 FunctionSymbol ，则不开启新的作用域。

3.4 重命名设计

问题描述：rename 打印树会重复。因为在获取类型等情况下对子节点先进行了遍历，之后又用 super 方法（不用的话会导致如 L_BRACE 符号打印不完整）重新遍历，所以导致重复打印。

解决方式：判断是否为第二次遍历，若是，则对子节点的遍历全部仅使用 super 的方式即可，防止重复遍历。

衍生问题描述：第二次遍历时无论哪个 IDENT 所处的作用域均为 Global 作用域。

解决方式：对进出作用域的操作予以放行，仅需保证不会发生重复遍历的操作即可。

3.5 定义形参时机

问题描述：发现部分形参定义在全局变量。原来是因为 FunctionSymbol 未完全定义及进入时即 define 了该形参。

解决方式：将形参列表记录下来，待进入 FunctionSymbol 后再逐个 define 形参，保证作用域正确。

3.6 函数重定义时无法跳过整个函数

解决方式：从使用 listener 转为使用 visitor，可以自定义访问，如果重定义直接不访问即可。后续发现 listener 也能实现。

3.7 重命名时无法得知 symbol 被定义的作用域

问题描述：如上所述，重命名需要获取其作用域作为标识，但现有函数无法支持获取 symbol 解析出来的是哪个作用域的信息。

解决方式：添加方法，逻辑与 resolve 基本一致，但需要返回该作用域的名字。

3.8 全局函数与局部变量冲突

问题描述：例如当局部作用域中存在 int a;，全局函数中存在 int a(int b), 则解析时选取局部变量，报错“对变量使用函数调用”，但实际上合法。

解决方式：在 CallFuncExp 中首先尝试解析全局函数，若发现不存在该函数再使用 currentScope.resolve(); 进行解析。