

PA2 实验报告

计 63

孙亚男

2016011285

1、实验要求

在本阶段，实验要求手工实现语义分析。具体的，我们需要对语法分析树做两遍扫描，第一趟扫描的时候建立符号表的信息，并且检测符号声明冲突以及跟声明有关的符号引用问题；第二趟扫描的时候检查所有的语句以及表达式的参数的数据类型。

2、实验实现

首先将 PA1-A 中 frontend 的部分拷贝过来，再将 Tree.java 中的相关部分整合在这次的文件中，具体内容在此不详细叙述。

对于不同的功能需求，具体的改变如下：

2.1 类的浅复制的支持

修改文件 TypeCheck.java

- 添加 visitSCopyExpr 函数。

通过 table 的 lookup 函数找到 scopy (ident, E) 中 ident 的 symbol。

按照提示中的逻辑对参数进行分析并按要求报错：判断 symbol 是否为空，

决定是否报错 UndeclVarError；判断 scopy 中两个参数的类型，决定是

否报错 BadScopyArgError。

2.2 sealed 的支持

修改 Class.java

- 添加新的 boolean 类型成员变量 sealed，用来记录该类是否为不可继承类，默认值为 false。
- 添加新的函数 isSealed 和 setSealed 用来得到和修改 sealed 的值。

修改 BuildSym.java

- 在 visitTopLevel 函数对 ClassDef 的遍历中，增加对该类是否为 Sealed

类的判断，若是，则将 sealed 参数置为 true。

- 在 visitTopLevel 函数对 ClassDef 的遍历中，增加对该类的父类是否为 sealed 属性的判断，若是，则报错 BadSealedInherError。

2.3 支持串行条件卫士

修改 TypeCheck.java

- 添加 visitGuarded 函数，将条件中的每一项分别 accept。
- 添加 visitIfSubStmt 函数，对 expr 是否为 BOOL 类型进行判断，若不是，则报错 BadTestExpr。

2.4 支持简单的类型推导

修改 Parser.y

- 修改 loc 对应的参数位置，从 1 (VAR) 改为 2 (IDENTIFIER) 以符合要求中对报错列号的规定。

修改 Tree.java

- 添加 UNKNOWN 常量。
- 在类 Var 中添加 VarDef 类型的变量，并初始化为 UNKNOWN。

修改 BaseType.java

- 添加新的 BaseType UNKNOWN = new BaseType("unknown");
- 修改 visitAssign 函数，判断对象复制前后的类型是否相同。

修改 Symbol.java

- 添加 setType 函数，用来修改 Symbol 中 Type 的值。

修改 BuildSym.java

- 添加 visitAssign 函数，对于 assign 左侧为 Var 类型对象的情况，assign.left.accept(this)；
- 添加 visitVar 函数，使 var 中的 vardef 接受 visit。

修改 TypeCheck.java

- 添加 visitVar 函数，将 var 的类型设置为 UNKNOWN。
- 修改 visitAssign 函数，对于 assign 左侧为 Var 类型对象的情况，从对应域中取出对应 Symbol，构造新的 Symbol，类型为 assign 右侧表达式类型，在将其加入符号表中，并修改 assign.left.type。

2.5 支持数组操作

2.5.1 数组初始化常量表达式

修改 TypeCheck. java

- 添加 visitNewSameArray 函数，根据提示中的逻辑判断 E%n 中的参数类型，并按要求报错：E 不是合法的数组元素的类型，则报错 BadArrElementError；n 不是 INT 类型，则报错 BadArrIndexError。
- 按逻辑为传入参数 newSameArray 设定 type，即：若没有错误，为 E 所对应的数组类型，若有错误，为 ERROR 类型。

2.5.2 数组下标动态访问表达式

修改 TypeCheck. java

- 添加 visitDefaultArray 函数，根据提示中的逻辑判断 E [E1] default E' 中的参数类型，并按要求报错：E1 不是 INT 类型，则报错 BadArrIndexError；E 不是合法的数组元素的类型且 E' 也不是合法的数组元素的类型，则报错 BadArrOperArgError；E 是合法的数组元素的类型但 E' 的类型与 E 不一致，则报错 BadDefError。
- 按逻辑为传入参数 defaultArray 设定 type，即：若没有错误，为 E 的元素类型，若有 E 不是合法的数组而 E' 为合法的数组元素类型，为 E' 的类型，若 E 与 E' 均不符合，为 ERROR 类型。

2.5.3 数组迭代语句

修改 Tree. java

- 在 ForeachArray 类中添加 LocalScope 类对象，用来记录当前作用域。

修改 BuildSym. java

- 添加 visitForeachArray 函数，判断 foreachArray 中 stmt 的类型，若为 Block，则用 stmt 构造 LocalScope，否则新建一个不指向任何实际代码的 Block 构造 LocalScope。
- open table 后 visit 前三个参数，第四个参数 stmt 采用 for 循环遍历的方式 visit 里面各项，close table。

修改 TypeCheck. java

- 添加 visitForeachArray 函数，根据提示中的逻辑判断 foreach (var x

in E while B) S 中的参数类型，并按要求报错：E 不是合法的数组元素的类型，则报错 BadArrOperArgError；B 不是 BOOL 类型，则报错 BadTestExpr。根据提示中的逻辑判断 foreach (Type x in E while B) S 中的参数类型，并按要求报错：E 不是合法的数组元素的类型，则报错 BadArrOperArgError；E 的元素类型和 x 类型不兼容，则报错 BadForeachTypeError；B 不是 BOOL 类型，则报错 BadTestExpr。

- 添加 visitVarBind 函数，用来区分上述两种情况。

3、实验体会

本次实验需要深刻理解 BuildSym 中扫描与 TypeCheck 中扫描的关系，并熟悉 Type, Scope 和 Symbol 之间的联系。在报错时，需要仔细考虑实现过程中的逻辑关系，以免误报或漏报。经过本次实验，我熟练掌握了符号表和作用域等概念，为后续实验打下基础。

4、声明

在完成本次实验第四小问 var 相关语法的过程中，曾就如何在构造符号表是将 var x 中的 x 设置为 UNKNOWN 的问题向计 62 班王逸凡同学请教，得到了在 Var 类中添加 VarDef 类型成员变量来记录的解决方法。