

我的程序实现了对程序的语法检查和重命名问题。

对于语法检查，我采用了listener的模式，并使用ParseTreeProperty记录各节点的信息，如果该节点上有出错，则利用ParseTreeProperty在该节点上标注NoneType(Type的一个子类)，父节点可以在exit时根据子节点标注的信息，来确定是否报错（如果子节点已经报错，即标记NoneType，则父节点无需再报错）。

对于重命名，采用了visitor模式，有三种模式，分别为构造模式、查找模式、打印模式。第一次构造符号表，第二次查找需要被替换的符号（隐含其作用域信息），第三次打印语法树时进行替换。

在实验中遇到的比较显著的bug是visitor作用域的切换时机不正确引起的。因为int a(int a)要区分两个a，则两者必须位于不同的作用域。回想resolve()是自底向上解析的。一开始我的做法是在访问FuncDef的子节点之前就将作用域切为函数作用域。这样两个a都会位于同一作用域。正确的做法是在访问FuncFParams的子节点前再切换作用域，这样访问第一个terminal a的时候仍然处在全局作用域。

但注意当没有参数时visitor是不会调用visitFuncFparams的方法，所以需要做一些作用域的调整。

还有一个bug是对分支语句不完善的override导致的。