

编译原理 lab5 实验报告

李维璇 201250058

实验思路

- 首先将原有符号表的Type转变为LLVMValueRef，其余大致可以不变。
- 添加 ParseTreeProperty<LLVMValueRef> 字段，保存每个节点的值，主要是为表达式所用。
- 接着，就可以顺着实验要求写了。

精巧的设计

- ParseTreeProperty<LLVMValueRef> 字段为求表达式的值方便了不少，并且支持递归，递归的终点是lval、函数调用或者number。
- 由于使用的是listener，对scope的支持也很好。

遇到的困难

排除一开始在console输出的问题后，遇到了以下两个问题并解决：

- 所有函数必须通过ret退出，否则会报错。这常出现于void类型的函数。我之前的ret都是在访问return_stmt时输出。
- 在访问lval时，出现了空指针异常。排查发现是有时候数组未加到符号表中。我误解了这句话：

（我们保证测试用例中的数组变量不以 `const` 修饰，且数组定义中数组的长度仅为常量表达式）。

我以为是数组均不会通过const声明。其实是数组变量不以const修饰，而会有数组常量的。但我感觉这一步 是在语义阶段检查，这里也不必强调说明。