

HW1-实验报告

胥昊天 22307110206

技术路线

1. 实现 `constantPropagater` :

- 实现方式与 `MinusIntConverter` 类似。
- 首先，我们默认调用 `constantPropagater` 之前先调用了 `MinusIntConverter`，这样就不存在单元操作中多个负号合并的情况，只需要考虑二元操作中常量的计算。

2. 实现 `Executor` :

- `Executor` 中我实现了“减号转负号”的功能，所以调用 `Executor` 之前并不需要先调用别的 visitor。
- 我们维护一个符号表 `map<string, int> symtab;`，以记录程序中已声明的变量的值。
- 对于二元表达式，如果都是常量，我们采取类似常量传播的方法；如果包含变量，我们则查询符号表（如果符号表中没有相应变量，则报错，并跳过该语句，继续执行后续代码），计算出二元表达式的值，并返回 `IntExp` 作为新的节点。
- 类似于二元表达式的执行方式，其余 AST 节点也采用此法：“只要能计算出值，就用计算出的整数值 `IntExp` 代替原节点”。
- `Esc` 表达式也是类似处理：先执行完 `Stm` 列表中的语句，再计算 `Exp` 的值，返回计算出的整数值 `IntExp` 代替原节点。
- 如此计算执行后，`Return` 节点的 `Exp` 一定是 `IntExp` 或者 `IdExp`，在 `int execute(Program *root);`，所以不难通过查询字符表得到返回值。

代码实现

[详见代码仓库](#)

代码里有详细注释。