四元式优化测试

目录

[1.测试内容： 1](#_Toc11282)

[2.覆盖情况 2](#_Toc3527)

[3.代数恒等式优化 2](#_Toc12825)

[3.1测试用例： 2](#_Toc2434)

[3.2测试结果 3](#_Toc5145)

[4.删除冗余临时变量 4](#_Toc26633)

[4.1测试用例 4](#_Toc24203)

[4.2测试结果 4](#_Toc31680)

[5删除公共子表达式 5](#_Toc23269)

[5.1测试用例： 5](#_Toc11867)

[5.2测试结果： 5](#_Toc25015)

[6.常量传播 7](#_Toc32020)

[6.1测试用例： 7](#_Toc3741)

[6.2测试结果： 7](#_Toc25429)

[7.集成测试： 8](#_Toc27814)

# 1.测试内容：

局部优化。

在进行测试用例设计的时候，既要设计可以优化的用例，确保用例被优化。

又要设计不能被优化的用例。

还有对于公共子表达式和常量传播的测试的时候，要生成中间的跳转指令，函数结束指令等语句对基本块的划分进行测试，以及修改公共子表达式的值，来检测判定是否正确。

# 2.覆盖情况

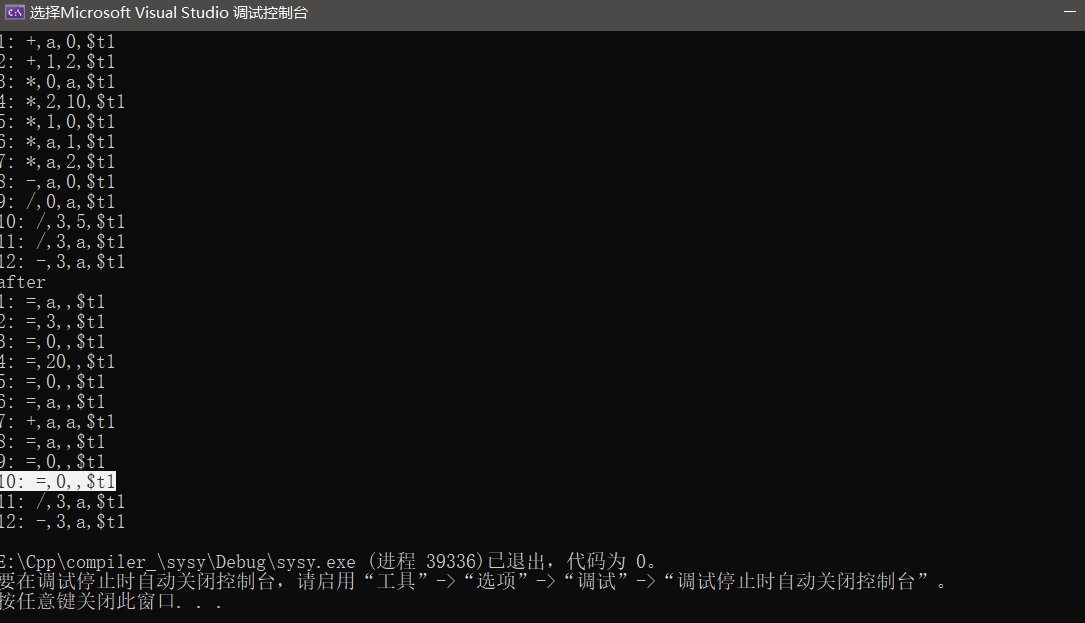
|  |  |
| --- | --- |
| 测试项目 | 是否通过 |
| 代数恒等式优化 | 通过 |
| 删除冗余临时变量 | 通过 |
| 删除公共子表达式 | 通过 |
| 常量传播 | 通过 |
| 集成测试 | 通过 |

# 3.代数恒等式优化

## 3.1测试用例：

|  |  |  |
| --- | --- | --- |
| 操作 | 优化前 | 优化后 |
| 加法 | (+,a,0,$t1) | (=,a,,$t1) |
| (+,0,a,$t1) | (=,a,,$t1) |
| 减法 | (-,a,0,$t1) | (=,a,0,$t1) |
| 乘法 | (\*,2,a,$t1 | (+,a,a,$t1) |
| (\*,a,2,$t1) | (+,a,a,$t1) |
| (\*,1,a,$t1) | (=,a,,$t1) |
| (\*,a,1,$t1) | (=,a,,$t1) |
| (\*,0,a,$t1) | (=,0,,$t1) |
| (\*,a,0,$t1) | (=,a,,$t1) |
| 除法 | (/,0,a,$t1) | (=,0,,$t1) |
| (/,a,1,$t1) | (=,a,,$t1) |
| (%,a,0,$t1) | (=,0,,$t1) |
| 常量合并 | (op,1,2,$t1) | (=,1+2,,$t1) |

## 3.2测试结果



设计测试用例的时候，针对优化的算法设计的都是关于算数运算的操作，四元式的两个源操作数有数字，标识符的组合，操作符有+,-,\*,/的组合。

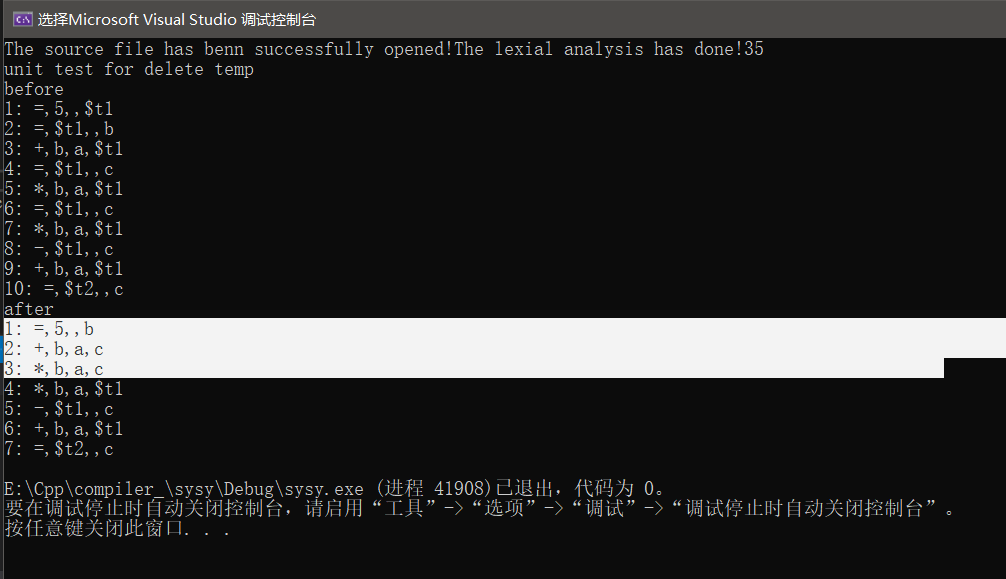
看到第十条测试用例，在使用整数除法的时候采用向下取整，因为sysy中没有浮点数的数据类型，所以这里不是错误，在生成mips汇编的时候也采取相应的规则。

# 4.删除冗余临时变量

## 4.1测试用例

|  |  |  |
| --- | --- | --- |
| 操作 | 优化前 | 优化后 |
| 赋值 | (=,5,,$t1) | (=,5,,b) |
| (=,$t1,,b) |
| 算术运算 | (+,a,b,$t1) | (+,a,b,c) |
| (=,$t1,,c) |

## 4.2测试结果



对于生成的冗余临时变量进行删除

对四元式的前六条优化后（标注），生成了优化后的中间向量。

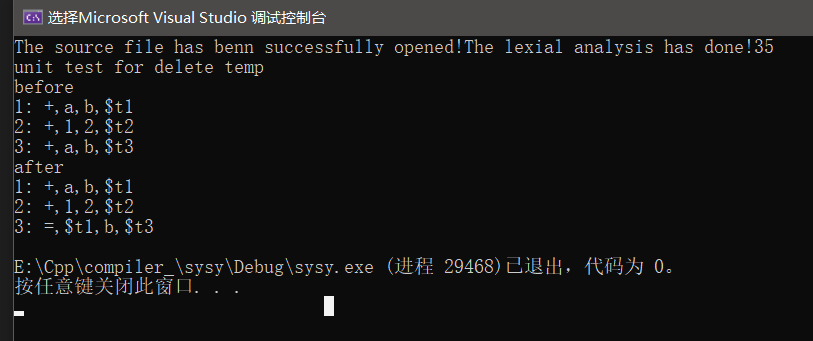
原先四元式的7-10无法优化

# 5删除公共子表达式

## 5.1测试用例：

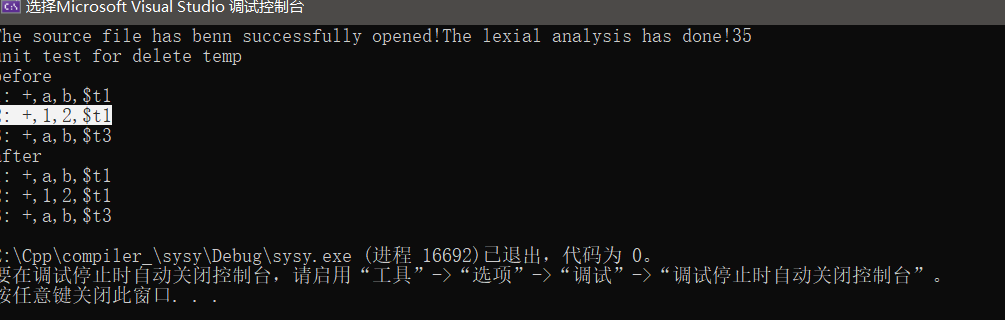
|  |  |
| --- | --- |
| 优化前 | 优化后 |
| （+,a,b,$t1） | (+,a,b,$t1) |
| 没有对a,b,t1的修改 | 没有对a,b,t1的修改 |
| (+,a,b,$t3) | (=,$t1,,$t3) |

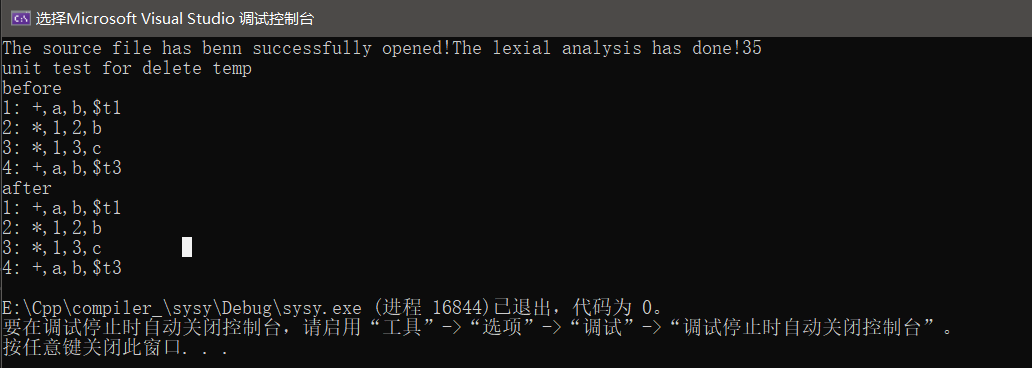
如下为一组可以通过执行的测试用例，在同一基本块中且没有对表达式的任意变量产生修改。



## 5.2测试结果：

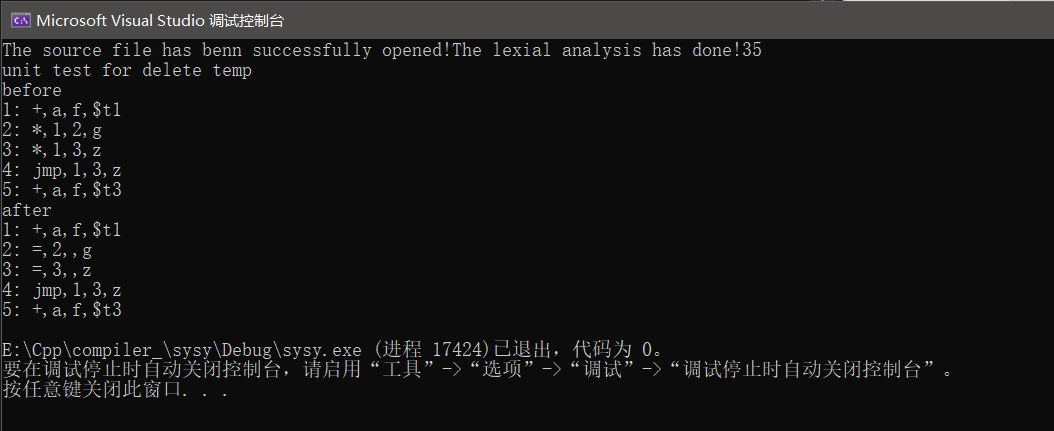
当对表达式中的任意一个操作数修改之后，都不能删除公共子表达式。



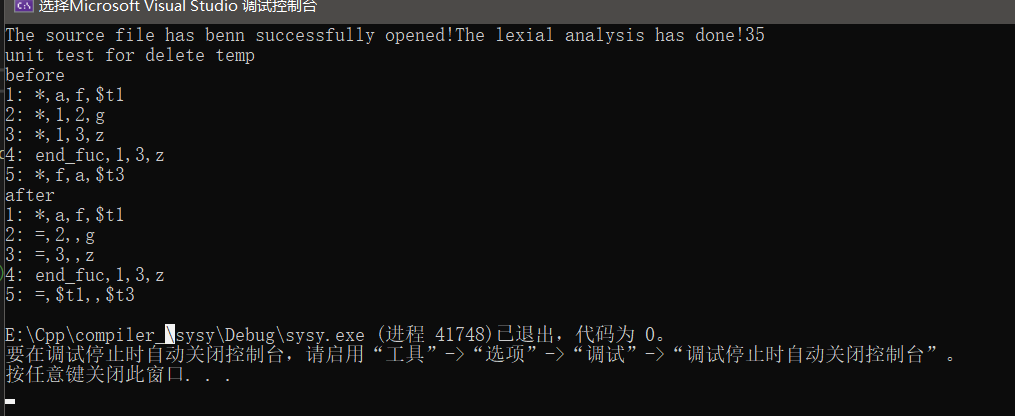


当中间的四元式没有对第一个公共子表达式进行修改的时候，则可以删除第二个公共子表达式。

当不在同一个基本快中的时候，不能进行删除，因为这里做的是局部优化。



交换律测试：



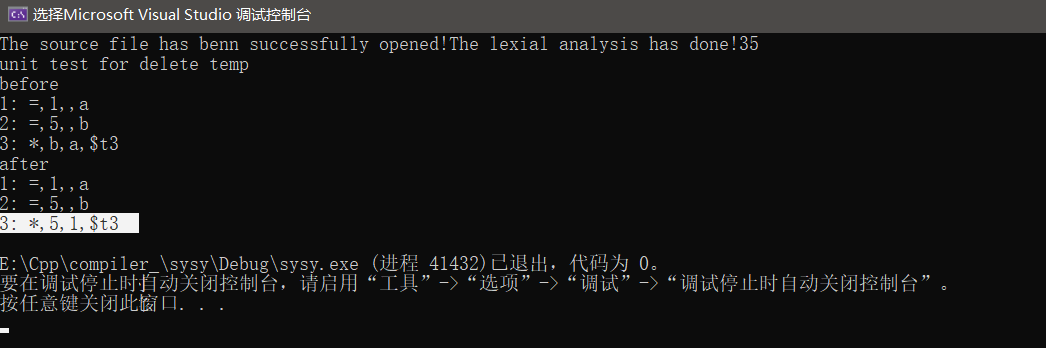
# 6.常量传播

## 6.1测试用例：

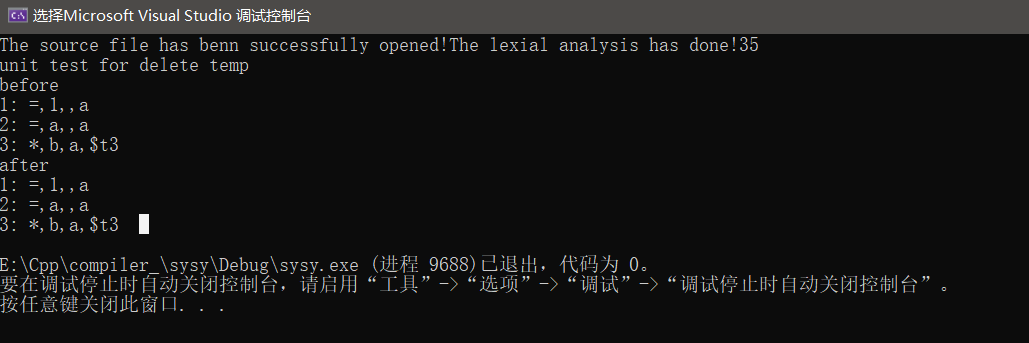
|  |  |  |
| --- | --- | --- |
| 操作 | 优化前 | 优化后 |
| 基本类型 | (=,1,,a) | (=,1,,a) |
| (=,1,,b) | (=,1,,b) |
| (+,a,b,c) | (+,1,1,c) |
| 数组 | (=[],5,1,a) | (=[],5,1,a) |
| (=,a,1,$t) | (=,a,1,$t1) |
| (=,$t1,,b) | (=,$5,,) |
|  |  |  |

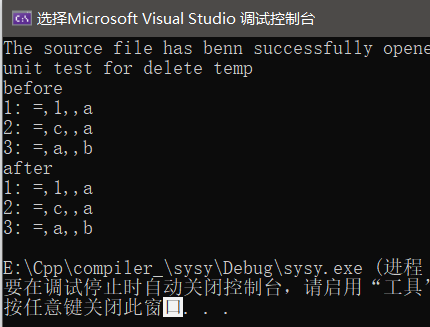
## 6.2测试结果：

可以进行常量传播的情况



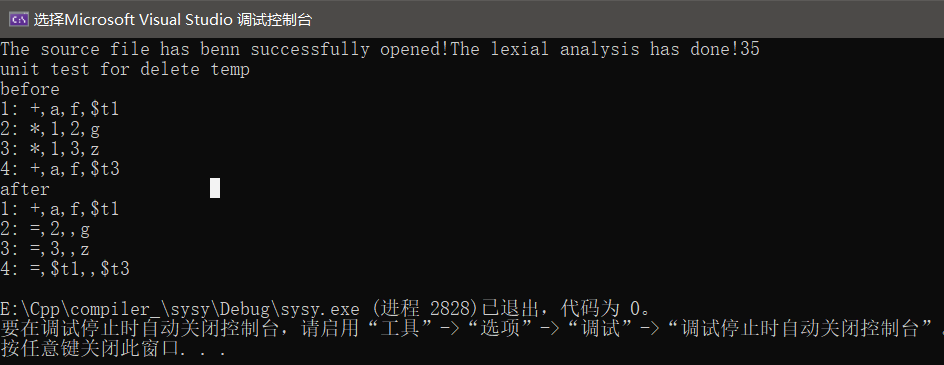
若修改了操作数，则无法进行常量传播





对于基本块的判定和删除公共子表达式相同。因此不再重复测试

# 7.集成测试：



因为各个优化模块之间单独作用，没有数据传递和调用关系，可以认为集成测试不会产生新的bug。因此没有进行大规模的集成测试，