hw11

1

a

B2,B3,B4,B5

B3,B4

b

对a复写传播并替换：B2中c=a+b ； B2中d=c-a ;B5中b=a+b ;B5中e=c-a ; B4中d=a+b

对b复写传播并替换：无

c

循环B2,B3,B4,B5的公共子表达式

a+b c-a

包括B2中c=a+b ； B2中d=c-a ;B5中b=a+b ;B5中e=c-a ; B4中d=a+b

B3,B4的公共子表达式：无

d

B2,B3,B4,B5的归纳变量： B2中c=c+1 B5中b=b+1

B3,B4的归纳变量： B4中e=e+1

e

B2,B3,B4,B5的循环不变计算：无

B3,B4的循环不变计算：d=a+b

2

b

	前驱	后继
ENTRY	NULL	B1
B1	ENTRY	B2
B2	B1 B5	B3 B5
B3	B2 B4	B4 B5
B4	B3	B3
B5	B2 B3	B2 B6
B6	B5	EXIT
EXIT	B6	NULL

	e_gen	e_kill
ENTRY	NULL	NULL
B1	1,2	a+b,c-a,b+d,b*d,a-d
B2	a+b,c-a	b+d,b*d,a-d
B3	NULL	b+d,b*d,a-d
B4	a+b	b+d,b*d,a-d,e+1
B5	c-a	a+b,b+d,b*d,a-d,e+1
B6	a-d	a+b,c-a,a+b,b+d,b*d
EXIT	NULL	NULL

全部表达式U：1,2,a+b,c-a,b+d,b*d,a-d,e+1

深度优先序：B1 B2 B3 B4 B5 B6 EXIT

第一次迭代：

	IN	OUT
ENTRY	NULL	NULL
B1	NULL	1,2
B2	1,2	1,2,a+b,c-a
B3	1,2,a+b,c-a	1,2,a+b,c-a
B4	1,2,a+b,c-a	1,2,a+b,c-a
B5	1,2,a+b,c-a	1,2,c-a
B6	1,2,c-a	1,2,a-d
EXIT	1,2,a-d	1,2,a-d

第二次迭代：不变

	IN	OUT
ENTRY	NULL	NULL
B1	NULL	1,2
B2	1,2	1,2,a+b,c-a
B3	1,2,a+b,c-a	1,2,a+b,c-a
B4	1,2,a+b,c-a	1,2,a+b,c-a
B5	1,2,a+b,c-a	1,2,c-a
B6	1,2,c-a	1,2,a-d
EXIT	1,2,a-d	1,2,a-d

C

	use[B]	def[B]
B1	NULL	a,b
B2	a,b	c,d
B3	b,d	NULL
B4	a,b,e	d
B5	a,b,c	e
B6	b,d	a

第一次迭代：

	OUT[B]	IN[B]
B6	NULL	b,d
B5	b,d	a,b,c,d
B4	NULL	a,b,e
B3	a,b,c,d,e	a,b,c,d,e
B2	a,b,c,d,e	a,b,e
B1	a,b,e	e

第二次迭代：

	OUT[B]	IN[B]
B6	NULL	b,d
B5	a,b,e,d	a,b,c,d
B4	a,b,c,d,e	a,b,c,e
B3	a,b,c,d,e	a,b,c,d,e
B2	a,b,c,d,e	a,b,e
B1	a,b,e	e

第三次迭代：保持不变

	OUT[B]	IN[B]
B6	NULL	b,d
B5	a,b,d,e	a,b,c,d
B4	a,b,c,d,e	a,b,c,e
B3	a,b,c,d,e	a,b,c,d,e
B2	a,b,c,d,e	a,b,e
B1	a,b,e	e

3

环境：GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0

未优化

- 运行时的表现：段错误

```
root@LAPTOP-HRJHHKLT:/mnt/d/mywork/compiler_lab/hw11# ./func1
Segmentation fault
```

- 无限的递归调用导致运行栈的溢出，引起段错误。
- 函数f对应汇编码如下

```
f:
.LFB0:
    .cfi_startproc
    endbr64
    pushq   %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    subq    $16, %rsp
    movq    %rdi, -8(%rbp)
    movq    -8(%rbp), %rax
    movq    -8(%rbp), %rdx
    movq    %rax, %rdi
    movl    $0, %eax
    call    *%rdx
    leave
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
```

采用二级优化

- 运行时的表现：一直运行，不终止

```
root@LAPTOP-HRJHHKLT:/mnt/d/mywork/compiler_lab/hw11# ./func2
```

- 在开启二级优化后，编译器得知函数f的return语句中的函数调用是尾递归调用。函数f被优化为把当前活动记录作为原本要新增的活动记录，并把调用指令`call`改为跳转指令`jmp`，故程序进入死循环，但不会出现运行栈的溢出。
- 函数f对应汇编码如下

```
f:
.LFB0:
    .cfi_startproc
    endbr64
    xorl    %eax, %eax
    jmp *%rdi   #rdi存放f的入口地址
    .cfi_endproc
```