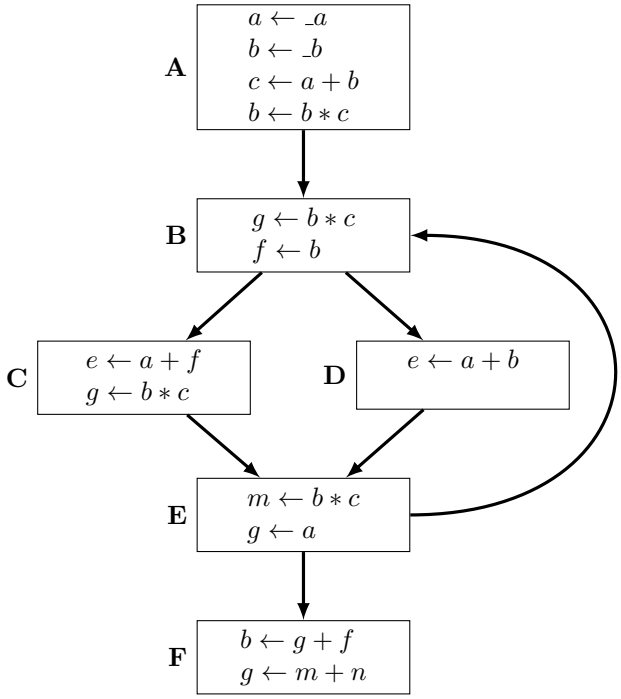
Assignment 6
(Due: Feb 22 23:59)
Ivan Neto

Problem 1: Convert the following CFG to a semi-pruned SSA form, following the three major steps:

1) Compute the dominance frontier for each basic block;
2) Insert phi-functions based on Dominance Frontier and Global Names;
3) Rename variables with counters and stacks.

[Note: to find global names, you may skip the entry block, whose UEVar does not generate any actual global names.]



Solution 1):

Block	**A**	**B**	**C**	**D**	**E**	**F**
DF	-	B	E	E	B	-

Solution 2):
The first step in inserting the ϕ-functions is calculating the global names - names that live across blocks.

To do this, I use the following algorithm given in lecture:

```
Globals <- emptyset
for each block B in the CFG
    VarKill <- emptyset
    for each operator x <- y op z in B, in order
        if y not in VarKill then
            Globals <- Globals union {y}
        if z not in varKill then
            Globals <- Globals union {z}
        VarKill <- VarKill union {x}
        Blocks{x} <- Blocks{x} union {B} // defining blocks
```

Running this algorithm, the following definition blocks are computed:

name	a	b	c	f	g	e	m	n
DB(s)	$\{A\}$	$\{A, F\}$	$\{A\}$	$\{B\}$	$\{CEFB\}$	$\{C, D\}$	$\{E\}$	-

Note: DB(s) means defining block(s)

Similarly, the following global names are computed

Globals = $\{a, b, c, g, f, m, n\}$

Note: I skip the entry block because UEVar does not generate any actual global names.

I use the following algorithm given in the lecture recording to insert the ϕ-functions:

```
/* Already done immediately before this */
compute global names Globals and Blocks(x)
for each block B in CFG
    compute DF(B)

for each name x in Globals
    WorkList <- Blocks(x)
    while WorkList != emptyset
        remove block B from WorkList
        for each block B_df in DF(B)
            if B_df has no phi-function for x then
                insert phi-function for x in B_df
                WorkList <- WorkList union {B_df}
```

I show each global name being inserted into the CFG:

For global name b:

No ϕ-functions are inserted because both blocks A and F do not have a dominance frontier. Therefore we cannot run the inner for loop in the algorithm.

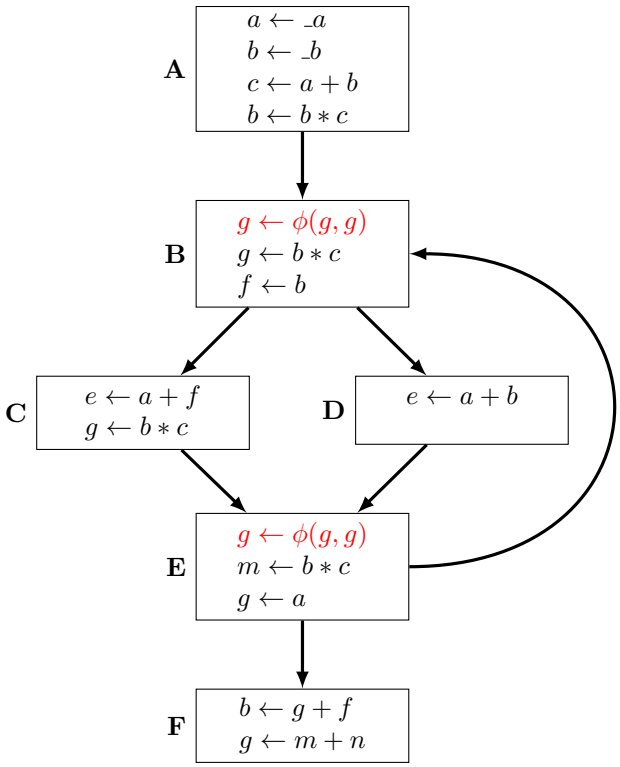
For global name a:

No ϕ-functions are inserted because block A does not have a dominance frontier. Therefore we cannot run the inner for loop in the algorithm.

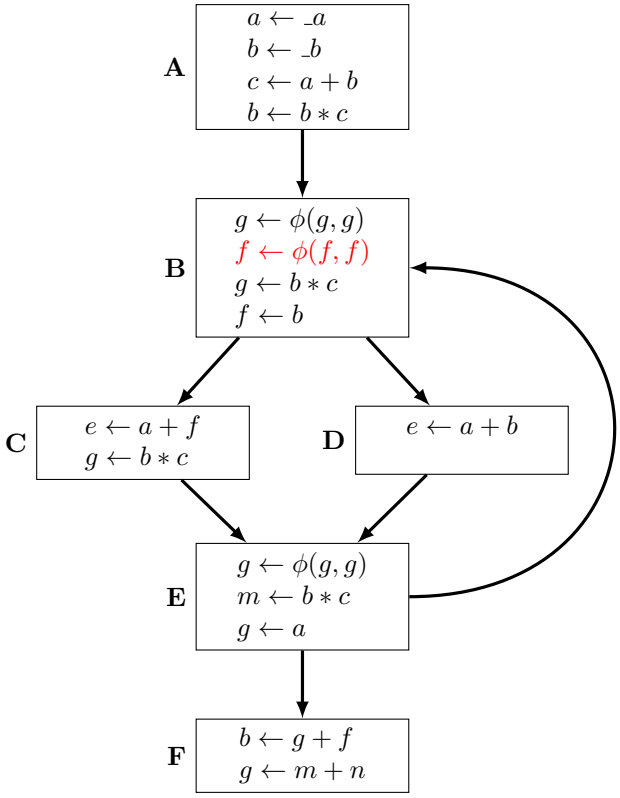
For global name c:

No ϕ-functions are inserted because block A does not have a dominance frontier. Therefore we cannot run the inner for loop in the algorithm.

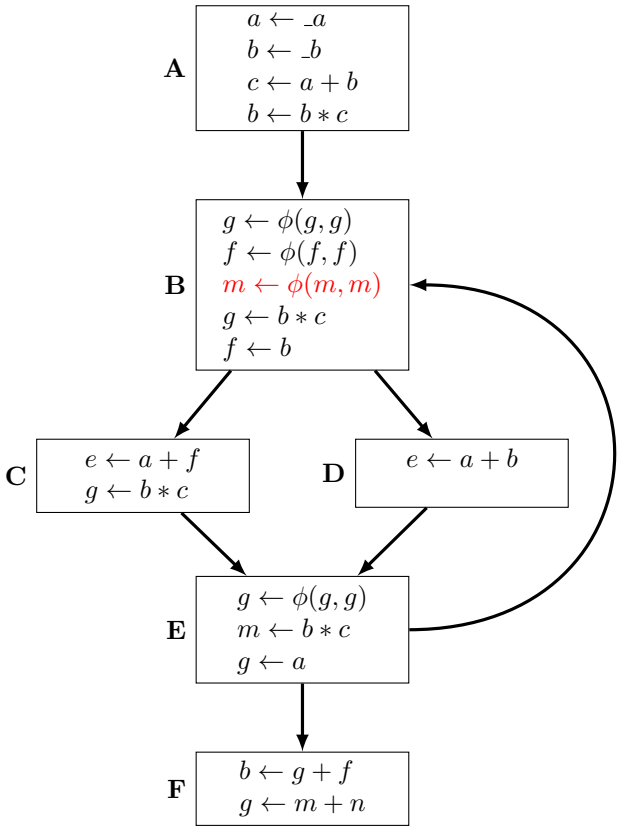
For global name g:



For global name f:



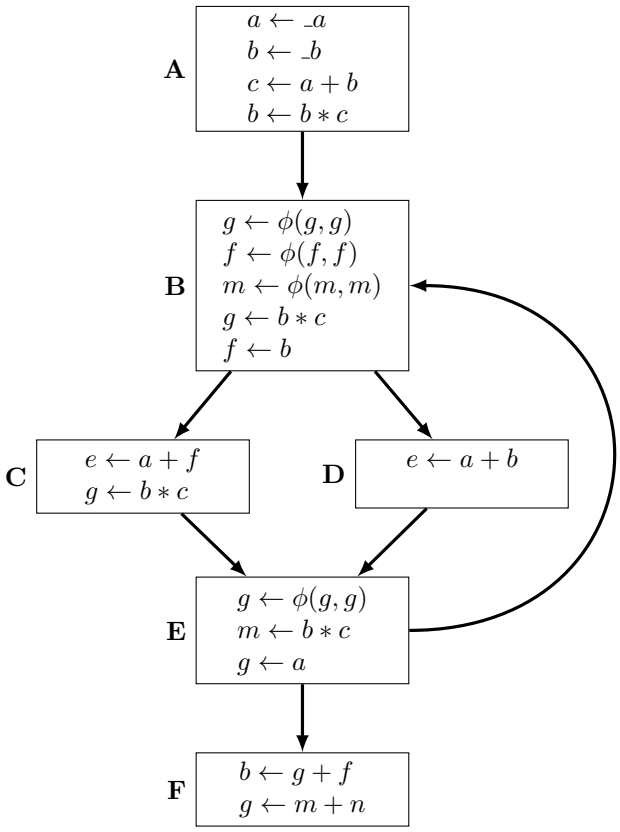
For global name m:



For global name n:

Nothing is done here because n does not have any defining blocks.

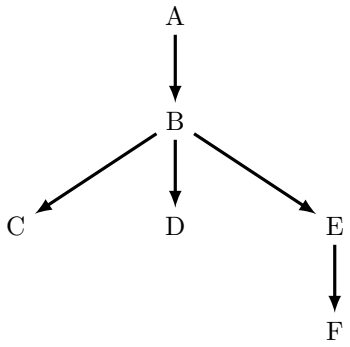
Final CFG:



Note: Although it looks like e should contain a ϕ-function at block E, e is not a global variable as it is not utilized in any subsequent blocks. Therefore we do not need a ϕ-function for e at block E.

Solution 3):

I start the process of renaming the ϕ-function variables by building the dominance tree:



To rename the variables, I follow the order of a preorder traversal of the dominance tree:

Pre(DomTree) $= \{A, B, C, D, E, F\}$

I will be using the following function definitions as provided by the lecture recording:

StartRename():

```
StartRename():
    for each global name i
        counter[i] <- 0
        stack[i] <- emptyset
    Rename(B_0)
```

NewName(n):

```
i <- counter[n]
push n_i onto stack[n]
counter[n]++
return n_i
```

Rename(B):

```
for each phi-function in B, x <- phi(...)
    rename x as NewName(x)

for each operation "x <- y op z" in B
    rewrite y as top (stack[y])
    rewrite z as top (stack[z])
    rewrite x as NewName(x)
```

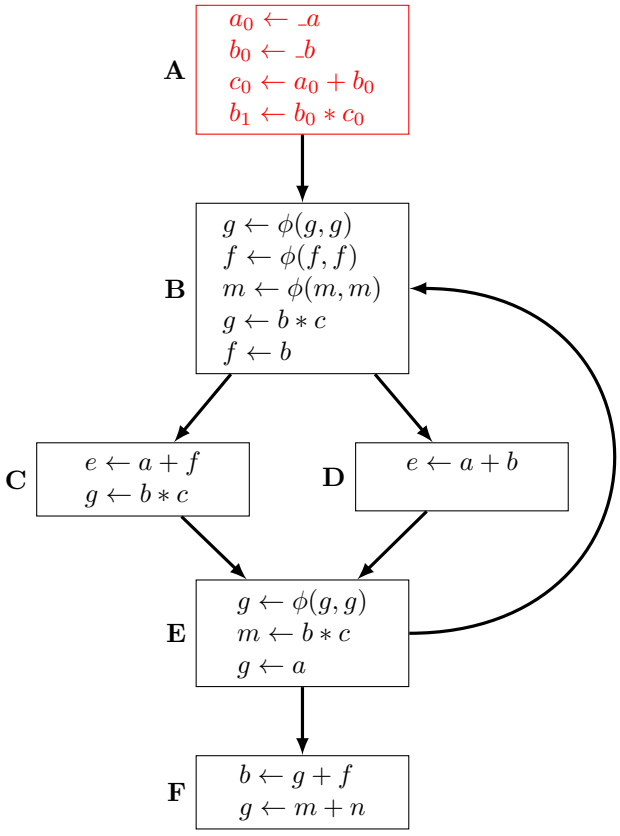
```
for each successor of B in the CFG
    rewrite appropriate phi parameters

// pre-order traversal on dom tree
for each successor S of B in the dom tree
    Rename(S)

// back to previous scope
for each operation "x <- y op z" in B
    pop(stack[x])
```

I use the above algorithm and show, for each iteration of the preorder traversal, the transformation at each block:

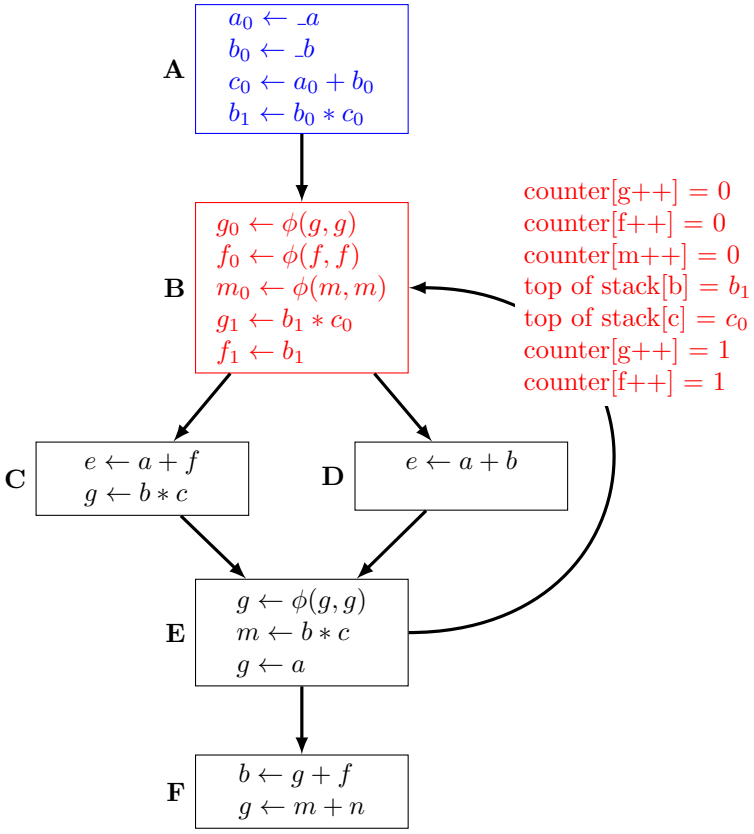
For A:



Since block A is the first block, I renamed each variable to its 0th version. Since no variables are defined that are used in the parameters for any ϕ-functions in B, I do not have to rewrite those parameters in B appropriately.

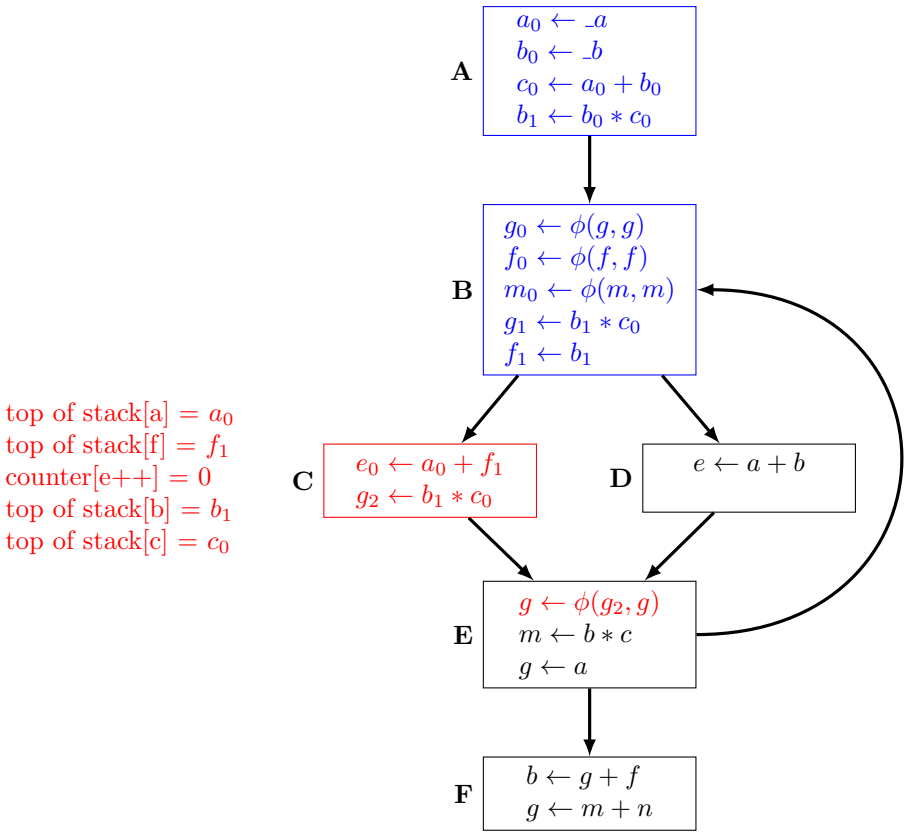
name	a	b	c	f	g	e	m	n
stack	$\{a_0\}$	$\{b_0, b_1\}$	$\{c_0\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
count	1	2	1	0	0	0	0	0

For B:



name	a	b	c	f	g	e	m	n
stack	$\{a_0\}$	$\{b_0, b_1\}$	$\{c_0\}$	$\{f_0, f_1\}$	$\{g_0, g_1\}$	$\emptyset$	$\{m_0\}$	$\emptyset$
count	1	2	1	2	2	0	1	0

For C:

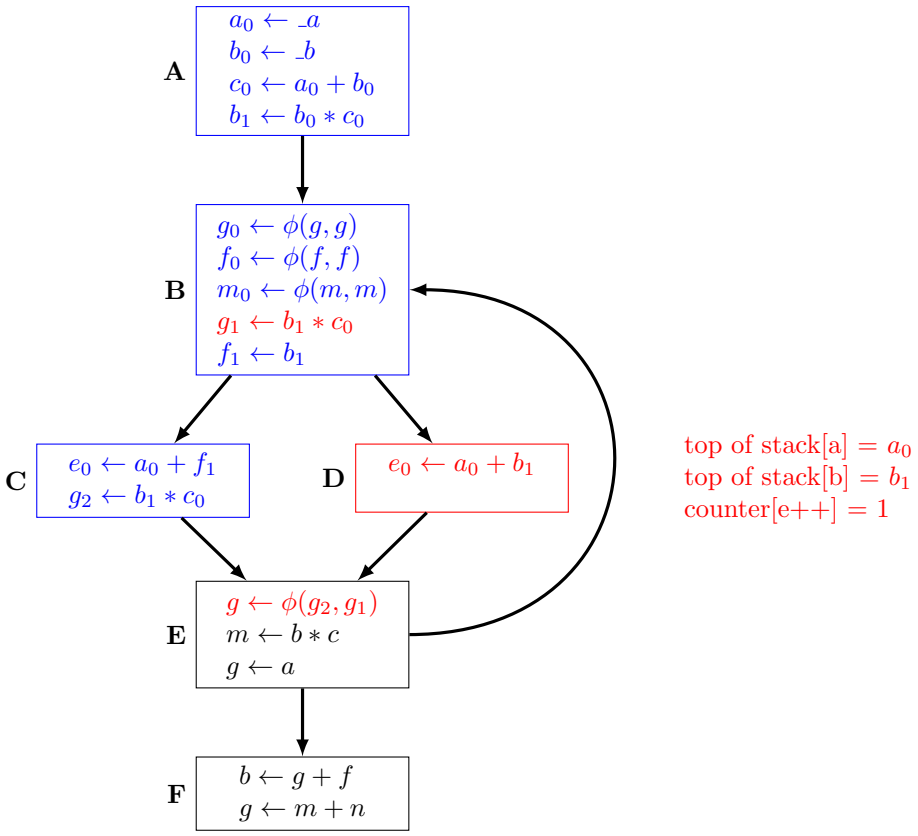


name	a	b	c	f	g	e	m	n
stack	$\{a_0\}$	$\{b_0, b_1\}$	$\{c_0\}$	$\{f_0, f_1\}$	$\{g_0, g_1, g_2\}$	$\{e_0\}$	$\{m_0\}$	$\emptyset$
count	1	2	1	2	3	1	1	0

Since on the dominance tree C is a leaf, we will need to pop all the x in C in the form $x \leftarrow y$ op z:

name	a	b	c	f	g	e	m	n
stack	$\{a_0\}$	$\{b_0, b_1\}$	$\{c_0\}$	$\{f_0, f_1\}$	$\{g_0, g_1\}$	$\emptyset$	$\{m_0\}$	$\emptyset$
count	1	2	1	2	2	0	1	0

For D:

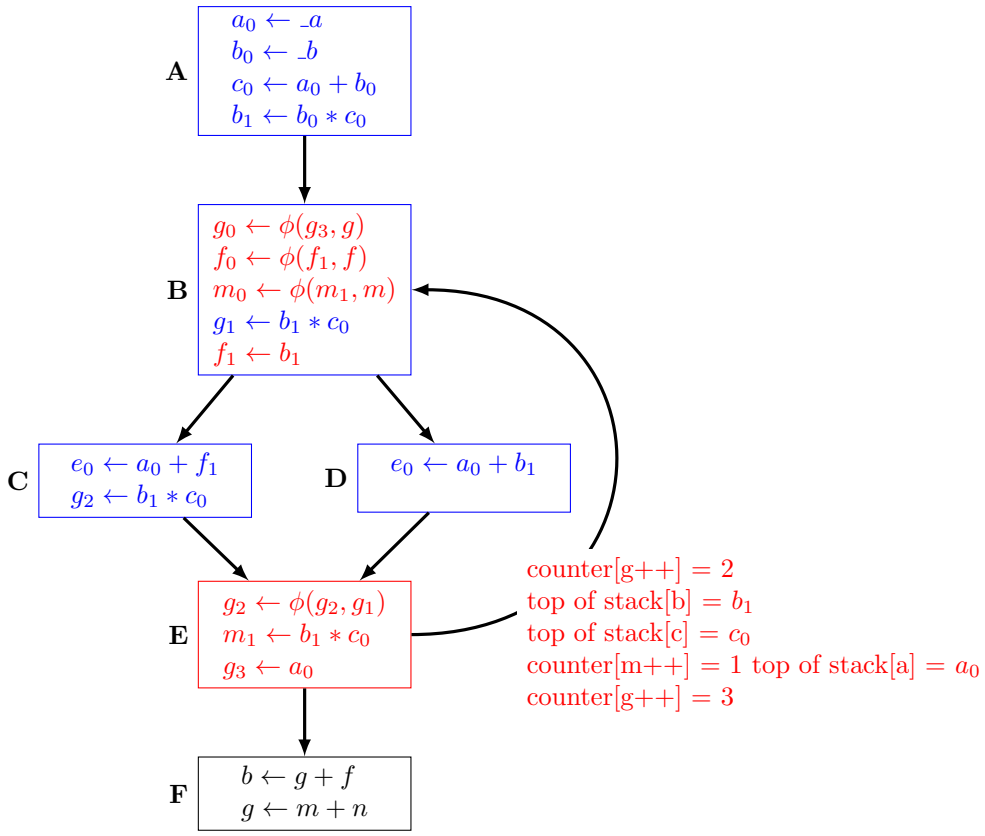


name	a	b	c	f	g	e	m	n
stack	$\{a_0\}$	$\{b_0, b_1\}$	$\{c_0\}$	$\{f_0, f_1\}$	$\{g_0, g_1\}$	$\{e_0\}$	$\{m_0\}$	$\emptyset$
count	1	2	1	2	2	1	1	0

Since on the dominance tree D is a leaf, we will need to pop all the x in C in the form $x \leftarrow y$ op z:

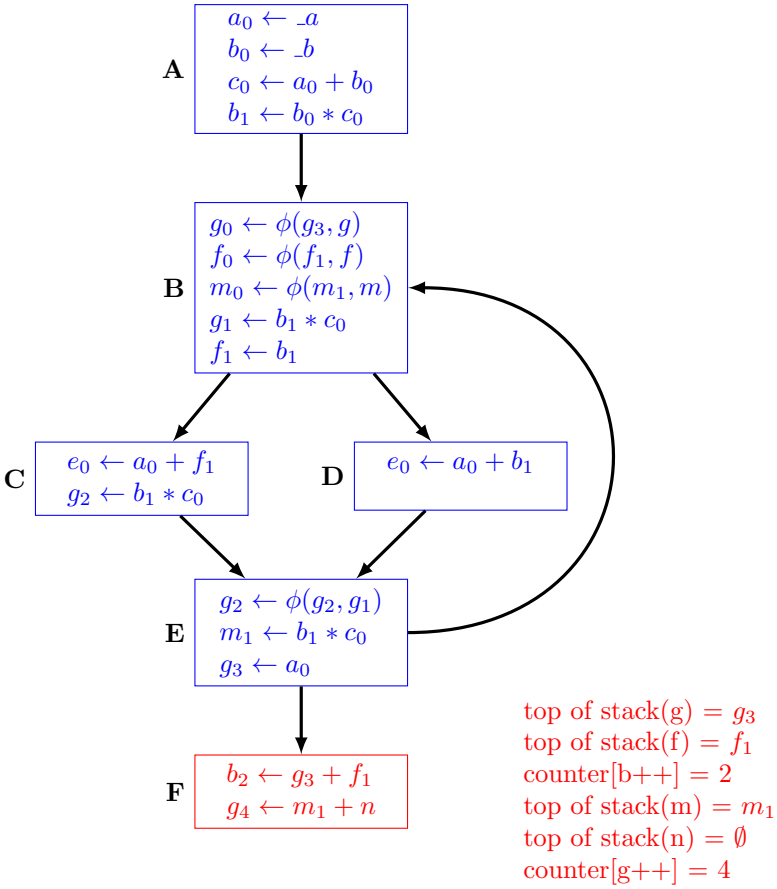
name	a	b	c	f	g	e	m	n
stack	$\{a_0\}$	$\{b_0, b_1\}$	$\{c_0\}$	$\{f_0, f_1\}$	$\{g_0, g_1\}$	$\emptyset$	$\{m_0\}$	$\emptyset$
count	1	2	1	2	2	0	1	0

For E:



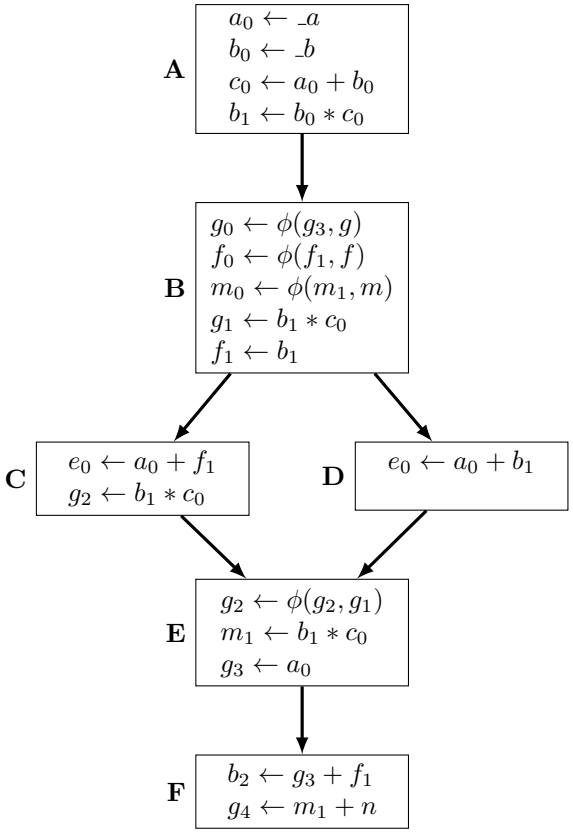
name	a	b	c	f	g	e	m	n
stack	$\{a_0\}$	$\{b_0, b_1\}$	$\{c_0\}$	$\{f_0, f_1\}$	$\{g_0 g_1 g_2 g_3\}$	$\emptyset$	$\{m_0, m_1\}$	$\emptyset$
count	1	2	1	2	4	0	2	0

For F:



name	a	b	c	f	g	e	m	n
stack	$\{a_0\}$	$\{b_0, b_1, b_2\}$	$\{c_0\}$	$\{f_0, f_1\}$	$\{g_0 ... g_4\}$	$\emptyset$	$\{m_0, m_1\}$	$\emptyset$
count	1	3	1	2	5	0	2	0

The final CFG:


A
$a_0 \leftarrow \_a$
$b_0 \leftarrow \_b$
$c_0 \leftarrow a_0 + b_0$
$b_1 \leftarrow b_0 * c_0$
B
$g_0 \leftarrow \phi(g_3, g)$
$f_0 \leftarrow \phi(f_1, f)$
$m_0 \leftarrow \phi(m_1, m)$
$g_1 \leftarrow b_1 * c_0$
$f_1 \leftarrow b_1$
C
$e_0 \leftarrow a_0 + f_1$
$g_2 \leftarrow b_1 * c_0$
D
$e_0 \leftarrow a_0 + b_1$
E
$g_2 \leftarrow \phi(g_2, g_1)$
$m_1 \leftarrow b_1 * c_0$
$g_3 \leftarrow a_0$
F
$b_2 \leftarrow g_3 + f_1$
$g_4 \leftarrow m_1 + n$