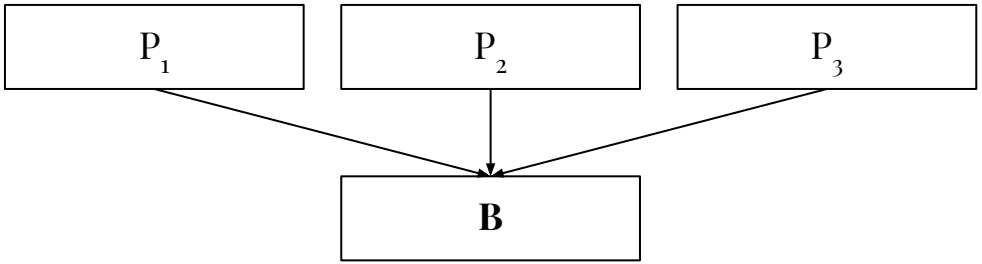
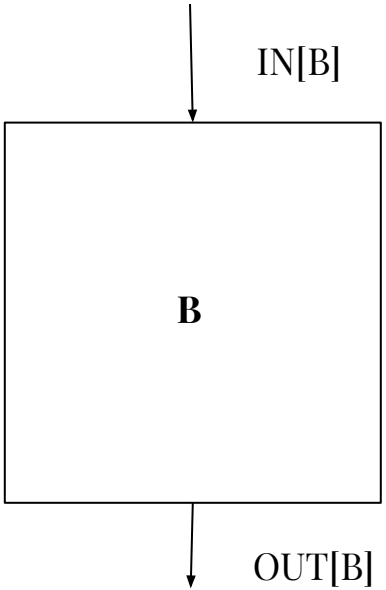
Data-flow analysis

Sudakshina Dutta



Control-flow equation



Transfer function

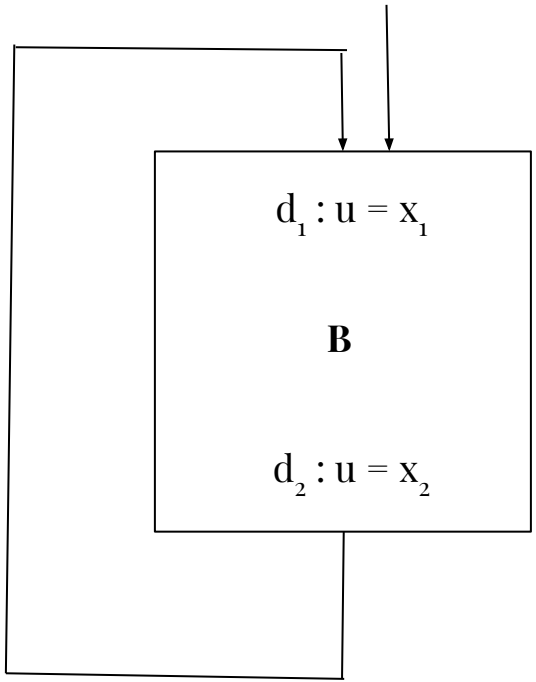
$$IN[B[= \ OUT[P_1] \cup OUT[P_2] \cup OUT[P_3]$$

$$OUT[B] = gen_B \cup (IN[B] - kill_B)$$

NOT REQUIRED.

$gen_B = \{d_2\}$

$kill_B = \{d_1, d_2\}$

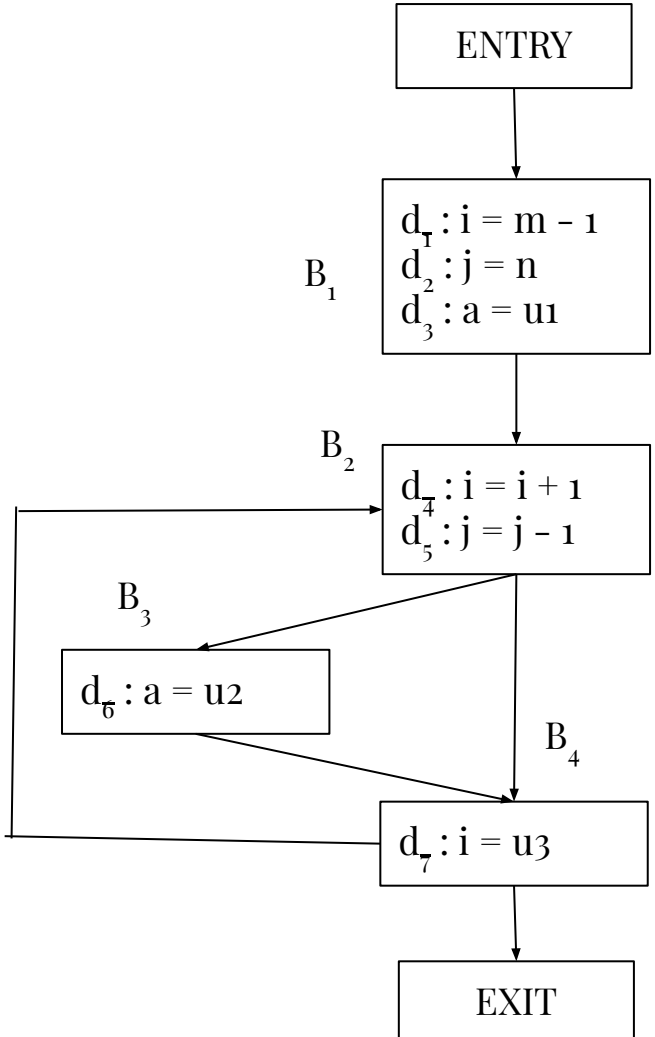


$IN[B]^1 = phi$

$OUT[B]^1 = gen_B \cup (IN[B]^1 - kill_B)$
$= \{d_2\}$

$IN[B]^2 = \{d_2\}$

$OUT[B]^2 = gen_B \cup (IN[B]^2 - kill_B)$
$= \{d_2\} \cup (\{d_2\} - \{d_1, d_2\})$
$= \{d_2\}$



$gen_{B1} = \{d_1, d_2, d_3\}$

$kill_{B1} = \{d_4, d_5, d_6, , d_7\}$

$gen_{B2} = \{d_4, d_5\}$

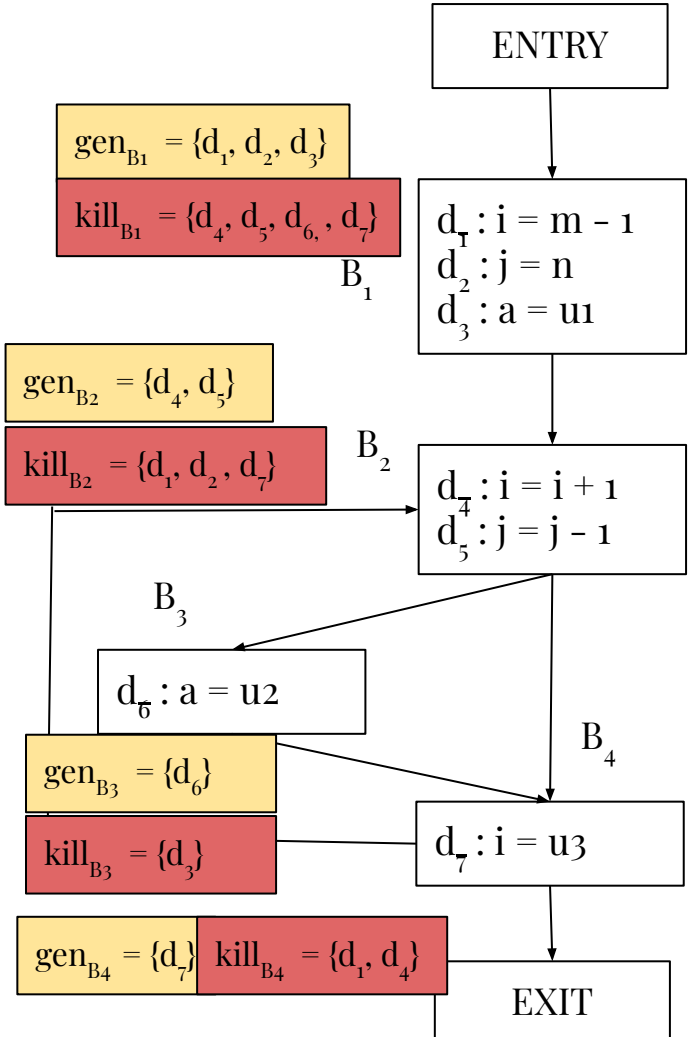
$kill_{B2} = \{d_1, d_2 , d_7\}$

$gen_{B3} = \{d_6\}$

$kill_{B3} = \{d_3\}$

$gen_{B4} = \{d_7\}$

$kill_{B4} = \{d_1, d_4\}$



$$IN[B[= OUT[P_1] \cup OUT[P_2] \cup OUT[P_3]$$

$$OUT[B] = gen_B \cup (IN[B] - kill_B)$$

$In[B_1]^1 = phi$

$Out[B_1]^1 = \{d_1, d_2, d_3\}$

$In[B_2]^1 = \{d_1, d_2, d_3\}$

$Out[B_2]^1 = \{d_3, d_4, d_5\}$

$In[B_3]^1 = \{d_3, d_4, d_5\}$

$Out[B_3]^1 = \{d_4, d_5, d_6\}$

$In[B_4]^1 = \{d_3, d_4, d_5, d_6\}$

$Out[B_4]^1 = \{d_3, d_5, d_6, d_7\}$

$In[B_2]^2 = \{d_1, d_2, d_3, d_5, d_6, d_7\}$

$Out[B_2]^2 = \{d_3, d_4, d_5, d_6\}$

$In[B_3]^2 = \{d_3, d_4, d_5, d_6\}$

$Out[B_3]^2 = \{d_4, d_5, d_6\}$

$In[B_4]^2 = \{d_3, d_4, d_5, d_6\}$

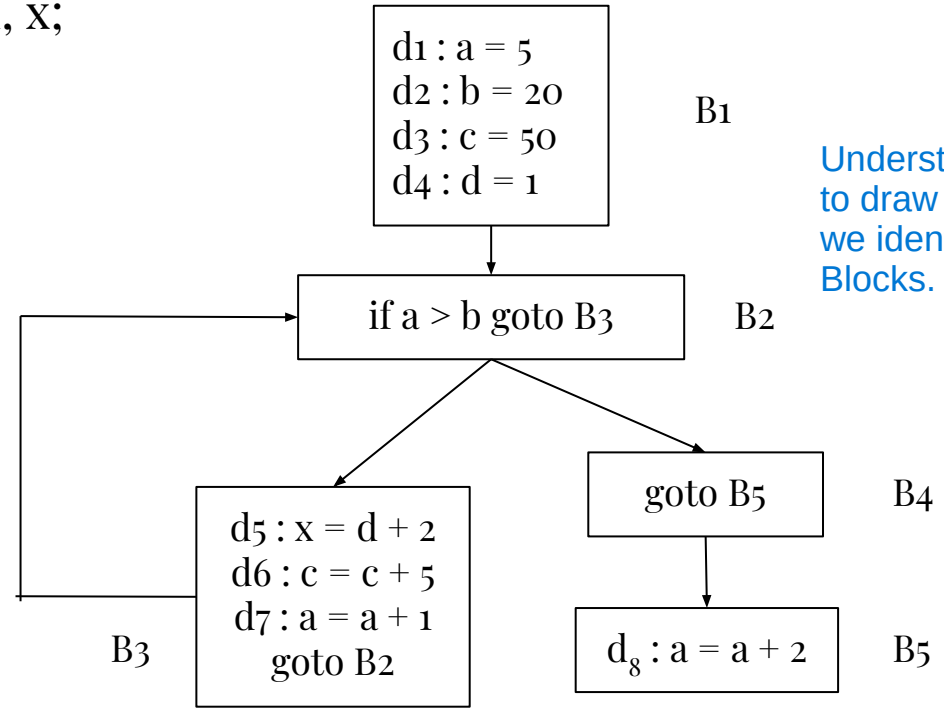
$Out[B_4]^2 = \{d_3, d_5, d_6, d_7\}$

Example

```
int a = 5, b = 20, c = 50, d = 1, x;
while (a < b)
{
    x = d + 2;
    c = c + 5;
    a = a + 1;
}
a = a + 2;
```

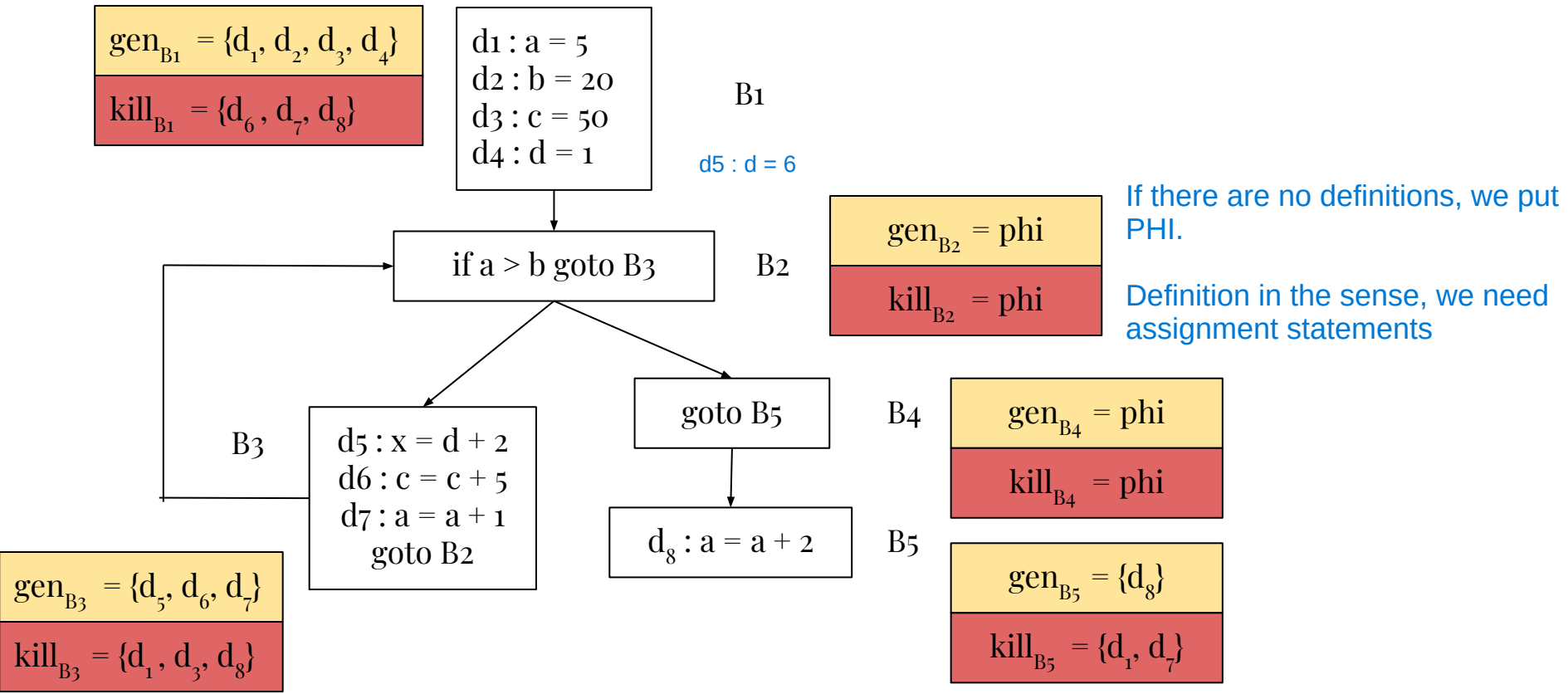
First convert the LHS Code to 3 address code & then see how this below graph came up.

Also, we can understand at what iteration to stop by seeing how outputs are changing.



Understand/Learn how to draw the graphs once we identify the Basic Blocks.

Example



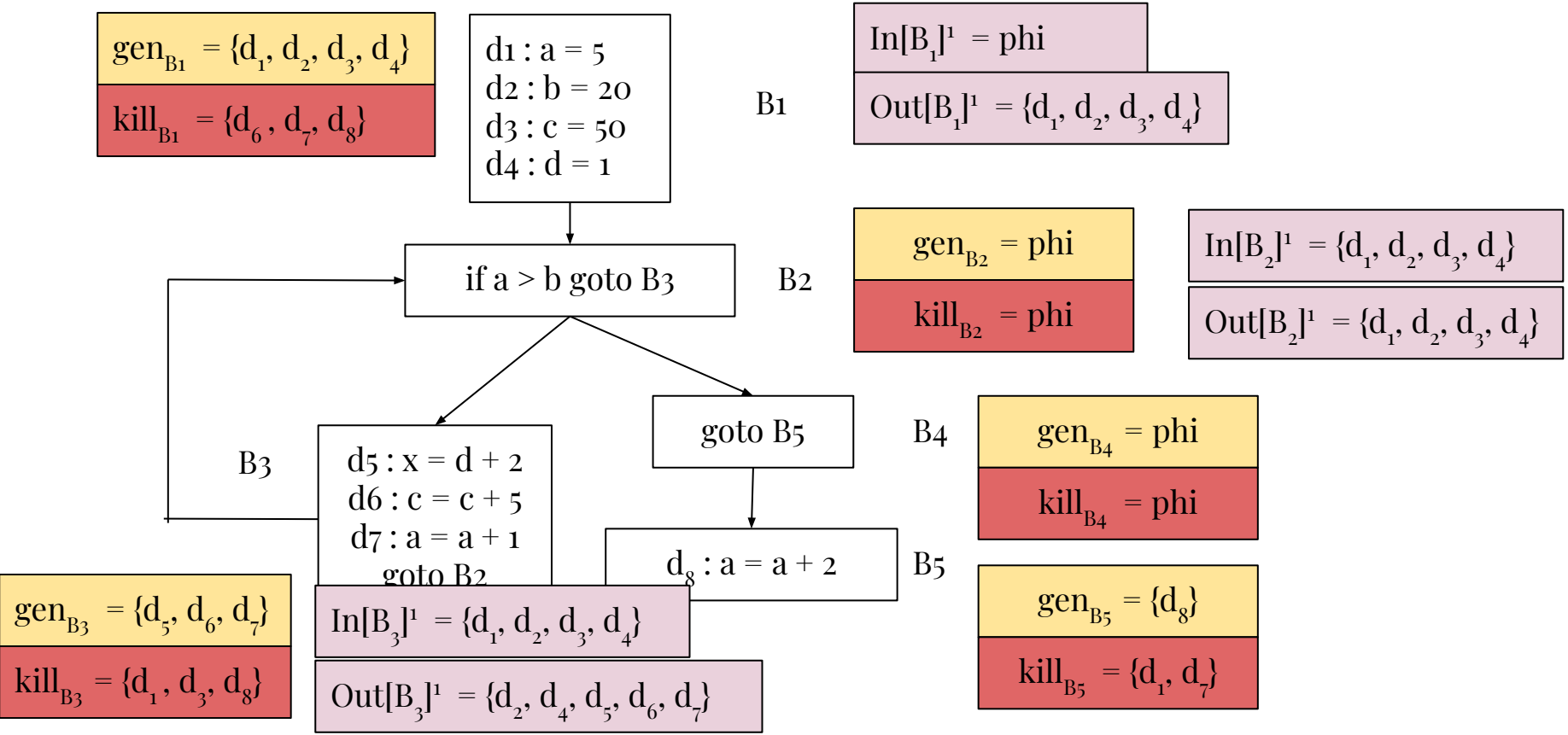
If there are no definitions, we put PHI.

Definition in the sense, we need assignment statements

Example

$IN[B[= OUT[P_1] \cup OUT[P_2] \cup OUT[P_3]$

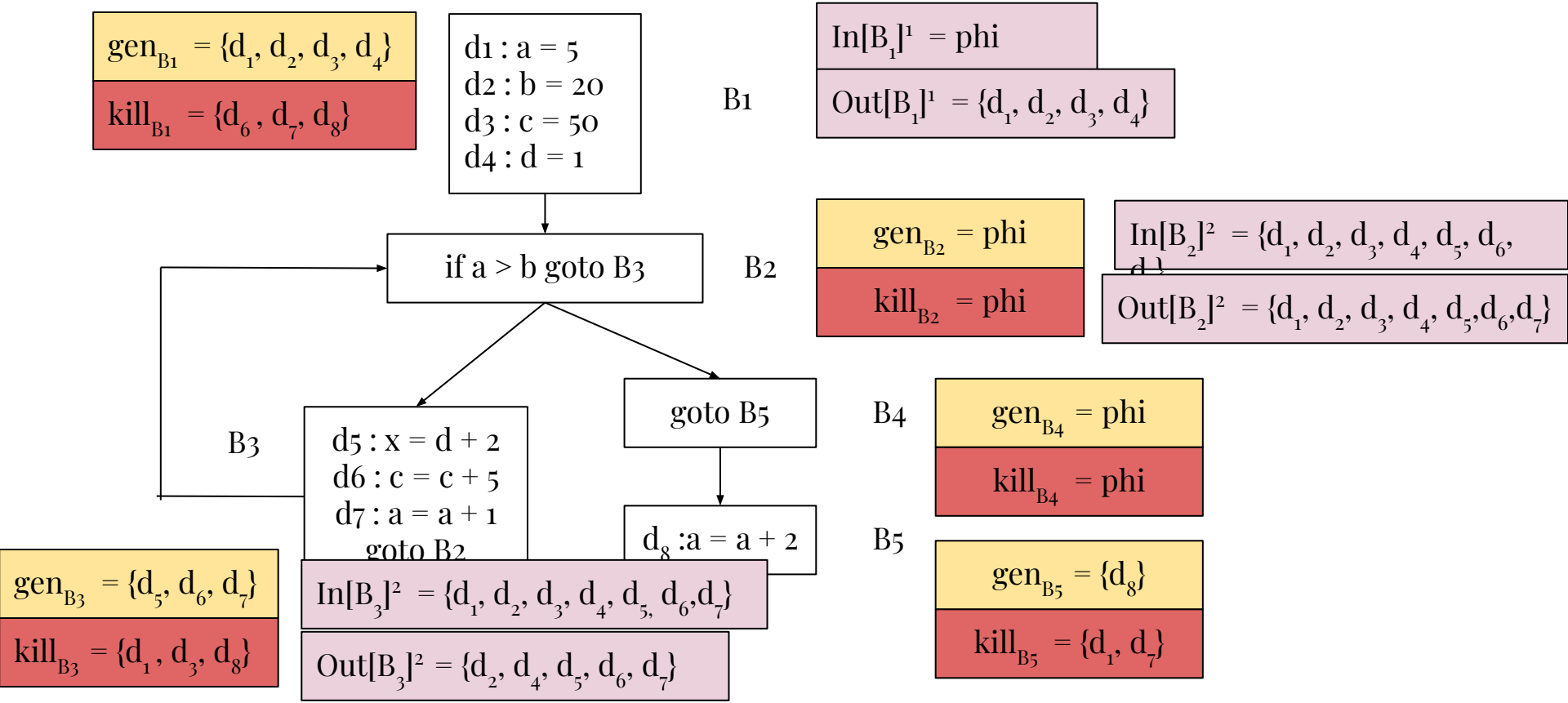
$OUT[B] = gen_B \cup (IN[B] - kill_B)$



Example

$IN[B[= OUT[P_1] \cup OUT[P_2] \cup OUT[P_3]$

$OUT[B] = gen_B \cup (IN[B] - kill_B)$



After optimization

```
int a = 5, b = 20, c = 50, d = 1, x;
if(a < b)
{
   x = d + 2;
   while (a < b)
   {
      c = c + 5;
      a = a + 1;
   }
}
a = a + 2;
```

This pre header should not be missed. Since this runs only once we go to while loop atleast once.