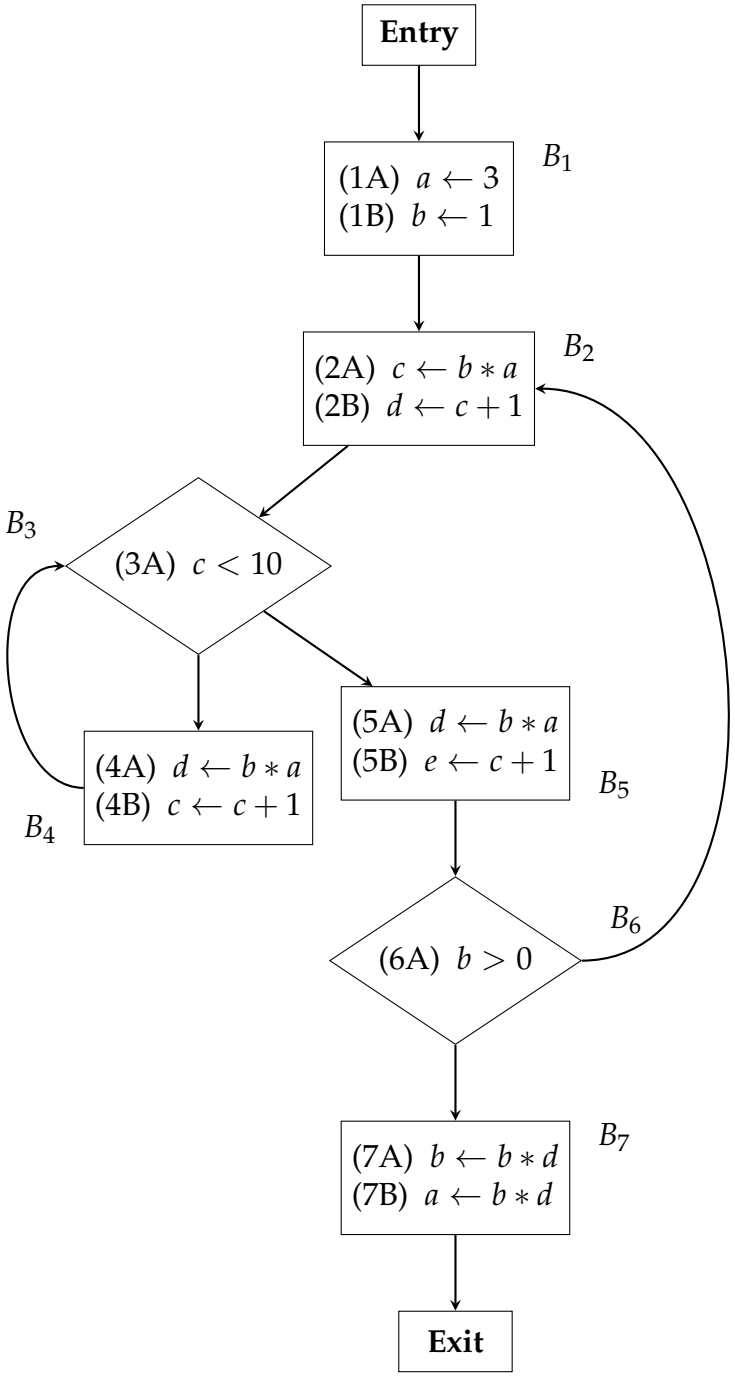
1 Program Analysis

This miniquiz will cover performing each of the three dataflow analyses discussed in lecture. Consider the following control graph for a single procedure.



1.1 Reaching Definitions

In this problem, you will perform a reaching definition analysis. Recall that a definition $x \leftarrow y \oplus z$ *reaches* a use U of x if U could read x as defined by $x \leftarrow y \oplus z$.

For each basic block B, the worklist algorithm updates its IN and OUT sets using these equations:

$$\mathsf{IN}[B] = \forall B_i \in \mathsf{pred}[B] \,.\, \bigcup \mathsf{OUT}[B_i].$$
$$\mathsf{OUT}[B] = (\mathsf{IN}[B] \setminus \mathsf{KILL}[B]) \cup \mathsf{GEN}[B].$$

Fill in the following table with the IN, OUT, KILL, and GEN sets for each basic block. As a starting point, the first row was filled in.

Basic Block B	GEN[B]	KILL[B]	IN[B]	OUT[B]
B_1	{1A, 1B}	{7A,7B}	$\emptyset$	{1A,1B}
B_2				
B_3				
B_4				
B_5				
B_6				
B_7				

1.2 Available Expressions

In this problem, you will perform an available expression analysis. Recall that an expression $x \oplus y$ is *available* at program point P if:

- All paths from the entry block to P evaluate $x \oplus y$ before reaching P.
- There are no re-definitions for x or y after the evaluation $x \oplus y$, but before P.

For each basic block B, the worklist algorithm updates its IN and OUT sets using these equations:

$$\mathsf{IN}[B] = \forall B_i \in \mathsf{pred}[B] \,.\, \bigcap \mathsf{OUT}[B_i].$$
$$\mathsf{OUT}[B] = (\mathsf{IN}[B] \setminus \mathsf{KILL}[B]) \cup \mathsf{GEN}[B].$$

Fill in the following table with the IN, OUT, KILL, and GEN sets for each basic block. As a starting point, the first row was filled in for you. Consider only the following expressions, numbered in the order provided below:

1. $b * a$
2. $c + 1$
3. $b * d$

Basic Block B	GEN[B]	KILL[B]	IN[B]	OUT[B]
B_1	$\emptyset$	{1,3}	$\emptyset$	$\emptyset$
B_2				
B_3				
B_4				
B_5				
B_6				
B_7				

1.3 Live Variables

In this problem, you will perform a liveness analysis. Recall that a variable x is said to be *live* at program point P if:

- Some path from P to the exit block contains a use U of x.
- There are no re-definitions for x along that path until U.

For each basic block B, the worklist algorithm updates its IN and OUT sets using these equations:

$$\mathsf{IN}[B] = \mathsf{USE}[B] \cup (\mathsf{OUT}[B] \setminus \mathsf{DEF}[B])\,.$$
$$\mathsf{OUT}[B] = \forall B_i \in \mathsf{succ}[B] \,.\, \bigcup \mathsf{IN}[B_i].$$

Fill in the following table with the IN, OUT, USE, and DEF sets for each basic block. As a start, the last row was filled in for you. Assume all variables are local, that is no variables are live upon exiting the procedure.

Basic Block B	USE[B]	DEF[B]	IN[B]	OUT[B]
B_1				
B_2				
B_3				
B_4				
B_5				
B_6				
B_7	$\{b, d\}$	$\{a, b\}$	$\{b, d\}$	$\emptyset$