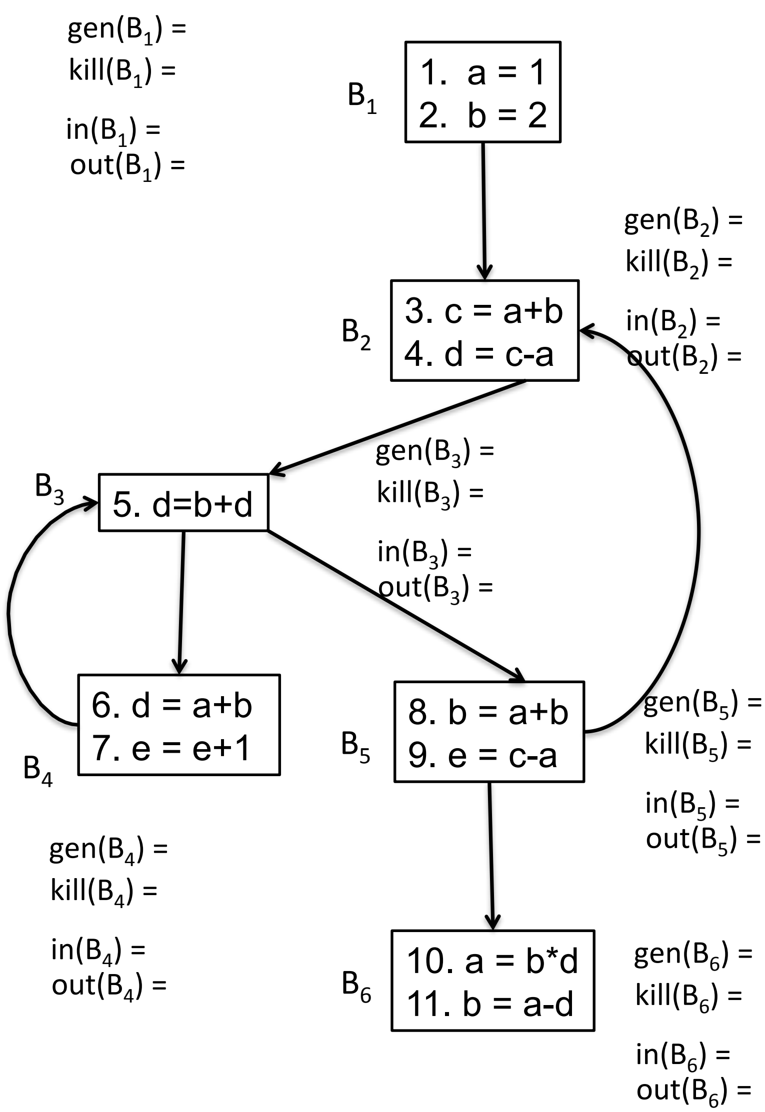
Homework 1

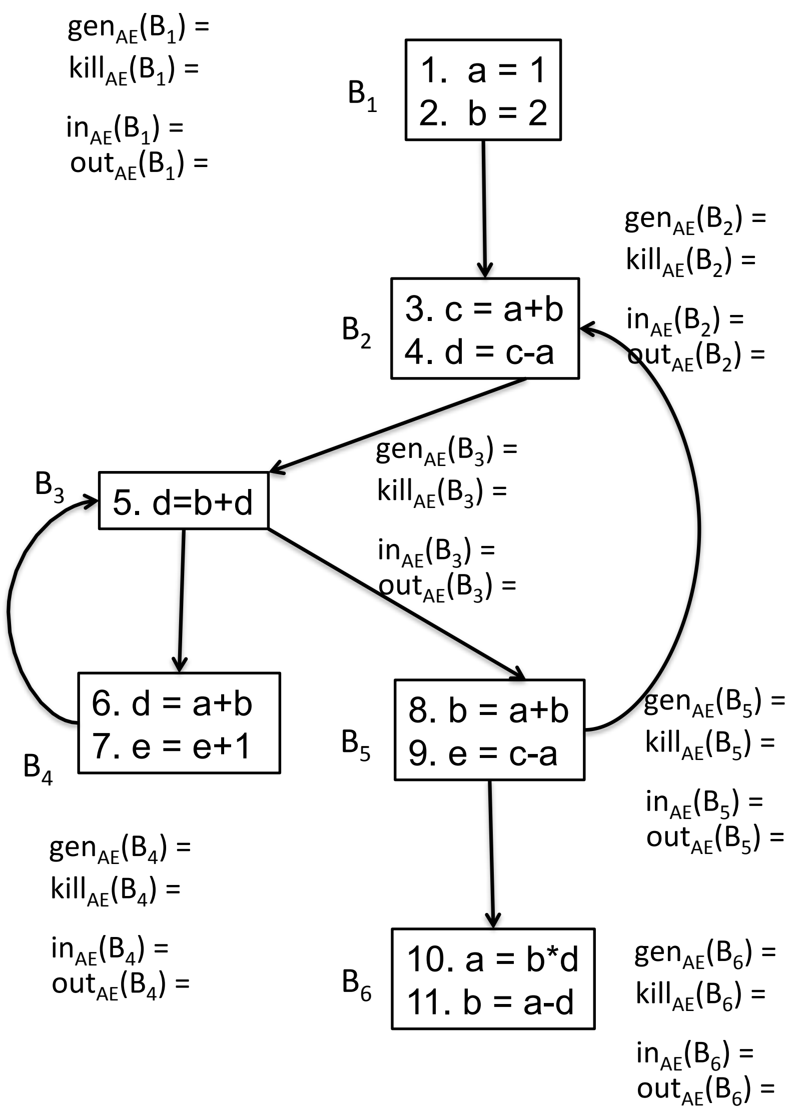
Posted Monday January 14, Due Monday January 28

50 points

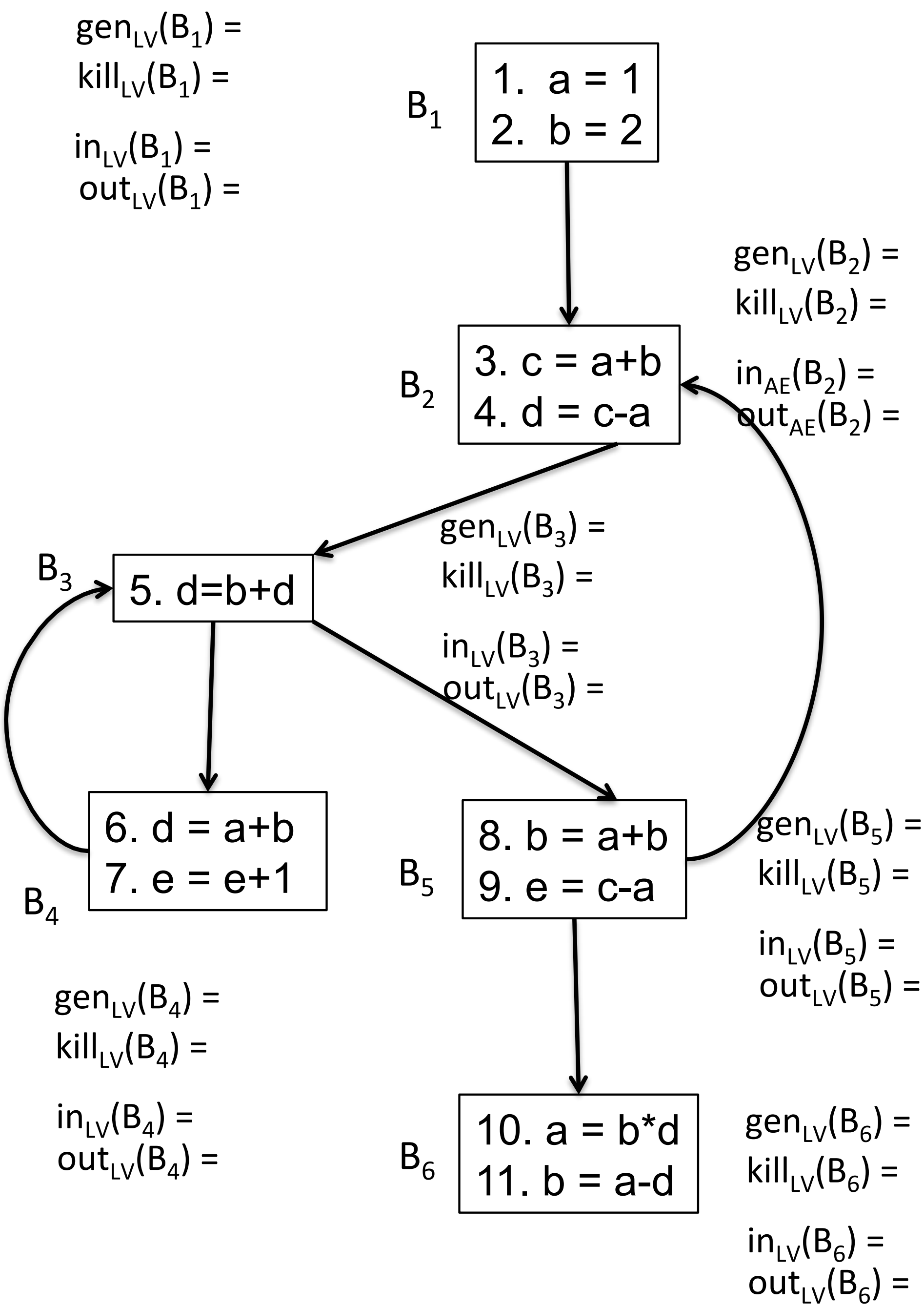
Problem 1 (6 pts). (From Aho, Lam, Sethi, Ullman.) For the CFG below fill in *Reaching Definitions* gen and kill sets for each block, and in and out sets for each block. (The in and out sets must show the final solution, not an intermediate value.)



Problem 2 (7 pts). (From Aho, Lam, Sethi, Ullman.) For the same CFG fill in *Available Expressions* $\mathsf{gen}_{\mathsf{AE}}$ and $\mathsf{kill}_{\mathsf{AE}}$ sets for each block, and $\mathsf{in}_{\mathsf{AE}}$ and $\mathsf{out}_{\mathsf{AE}}$ sets for each block.



Problem 3 (7 pts). (From Aho, Lam, Sethi, Ullman.) Now fill in *Live Variables* gen_{LV} and kill_{LV} sets for each block, and in_{LV} and out_{LV} sets for each block.



Problem 4 (10 pts, 2.5 pts each). (From Aho, Lam, Sethi, Ullman.) Let V be the set of complex numbers. Which of the following operations can serve as the meet operation for a lattice over V? For each of the choices below, if your answer is NO, explain why not. If your answer is YES, leave it at that.

a) Addition: $(a + ib) \wedge (c + id) = (a + c) + i(b + d)$

b) Multiplication: $(a + ib) \wedge (c + id) = (ac - bd) + i(ad + bc)$

c) Component-wise minimum: $(a + ib) \wedge (c + id) = min(a, c) + i\ min(b, d)$

d) Component-wise maximum: $(a + ib) \wedge (c + id) = max(a, c) + i\ max(b, d)$

Problem 5 (10 pts). The intraprocedural *Must-be-modified* problem is a backward dataflow problem solvable by fixpoint iteration. A variable is in the must-be-modified set on exit of CFG node n, if it is modified on *all paths* from n to exit. The problem statement is as follows: for each node n compute the set of variables that are in the must-be-modified set on exit from n.

a) (5 pts) Define the analysis as an instance of the dataflow framework. Specify

Lattice $L, \leq$:

Merge operator:

Transfer functions:

b) (5 pts) Are the functions for this problem distributive or monotone? Show your proof.

Problem 6 (10 pts). (Modified from Nielson, Nielson and Hankin) A *bit vector* dataflow analysis is a special case of a monotone dataflow analysis where

I. The property space L is the lattice of the subsets over some finite set D, and $\leq$ is either $\subseteq$ or $\supseteq$ and

II. The transfer function space is $F = \{f : \mathcal{P}(D) \to \mathcal{P}(D) \mid f(Y) = (Y \cap Y_f^1) \cup Y_f^2$ where $Y_f^1 \subseteq D$ and $Y_f^2 \subseteq D$ are constants$\}$

Note: $\mathcal{P}(D)$ denotes the powerset of D (the powerset is also frequently denoted by 2^D). The above condition states that every transfer function f can be written as $f(Y) = (Y \cap Y_f^1) \cup Y_f^2$ where Y is the argument of the function (the in(j) set in a forward problem), and Y_f^1 and Y_f^2 are *constants* that do not depend on Y.

a) (5 pts) Briefly argue that the four classical dataflow analyses are bit vector dataflow analyses.

b) (5 pts) Devise a distributive analysis that is *not* a bit vector analysis. Hint: Consider one of the non-distributive examples we studied in class, and drop one of the statements from the syntax.