NAME: **ROLL NO:**

CS738: Advanced Compiler Optimizations
Mid Semester Examination, 2018-19 I

Max Time: 2 Hours Max Marks: 95

NOTE:

- There are total **3** questions on **3 pages**.
- Write your name and roll number on the question paper and the answer book.
- Presenting your answers properly is your responsibility. You lose credit if you can not present your ideas clearly, and in proper form. Please DO NOT come back for re-evaluation saying, "What I actually meant was ...".
- Be precise and write clearly. Remember that somebody has to read it to evaluate!

Notations

- CFG stands for control flow graph.
- $\mathsf{IN}(S)$ denotes the program point before the statement S. $\mathsf{OUT}(S)$ denotes the program point after the statement of S.
- $\mathsf{PRED}(S)$ denotes the set of predecessors, and $\mathsf{SUCC}(S)$ denotes the set of successors of S.
- In a CFG, $x \xrightarrow{+} y$ denotes a path from node x to node y, having one or more edges. Both x and y are considered to be a part of the path.
- $\mathrm{DF}^+(\varphi)$ denotes the Iterated Dominance Frontier of the set of CFG nodes φ.

---

1. Prove the following statement: [15[5+10]]

   For any non-null path $p : X \xrightarrow{+} Z$ in a CFG, there exists a node $X' \in \{X\} \cup \mathrm{DF}^+(\{X\})$ on p that dominates Z. Moreover, unless X dominates every node on p, the node X' can be chosen in $\mathrm{DF}^+(\{X\})$.

   (a) X **dominates every node in** p. Clearly X dominates Z.

   (b) X **does not dominate every node in** p. Suppose the sequence of nodes in the path p is $n_0(= X), n_1, n_2, \ldots, n_k(= Z)$. Since X does not dominate all nodes in p, some of the nodes in p will be in $\mathrm{DF}^+(\{X\})$ **(WHY?)**. Let n_j be the node in $\mathrm{DF}^+(\{X\})$ such that it has the highest value of j. We claim that $X' = n_j$, i.e., n_j dominates Z. Suppose n_j does not dominate $Z = (n_k)$. Then, $\exists i, j < i \leq k$ such that n_j does not dominate n_i. Choose smallest such i. We have, parent of n_i dominated by n_j, but n_i is not (strictly) dominated by n_j. This gives us:

   $$\begin{aligned} & n_i \in \mathrm{DF}(\{n_j\}) \\ \Rightarrow\ & n_i \in \mathrm{DF}^+(\{n_j\}) \\ \Rightarrow\ & n_i \in \mathrm{DF}^+(\{X\}) \end{aligned}$$

   But this contradicts the fact that j is the largest index such that $n_j \in \mathrm{DF}^+(\{X\})$.

2. **Shortest Use Distance of A Definition.** [35[25+8+2]]

Let a definition d define a variable x at a program point π_d. Let program point π_u contain a use of x on some path from π_d to Exit, such that x is not redefined between π_d and π_u. The number of instructions between π_d and π_u is a **use distance** of d. If there is no use of x corresponding to d on some path from π_d to Exit, then the use distance on that path is ∞.

The **shortest use distance (SUD)** of d is defined as the minimum over all use distances of d.

Figure **??** shows an example program CFG, having variables A, B and C. SUDs for various definitions for this example are:

Stmt	Var	SUD
S1	A	∞
S2	A	2
S3	B	1
S6	C	1

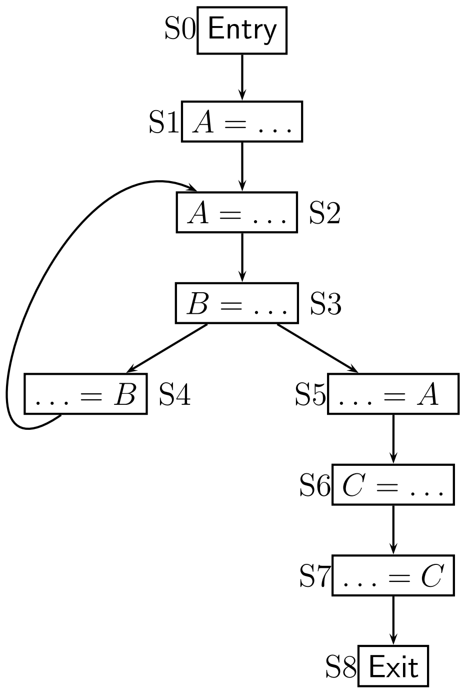


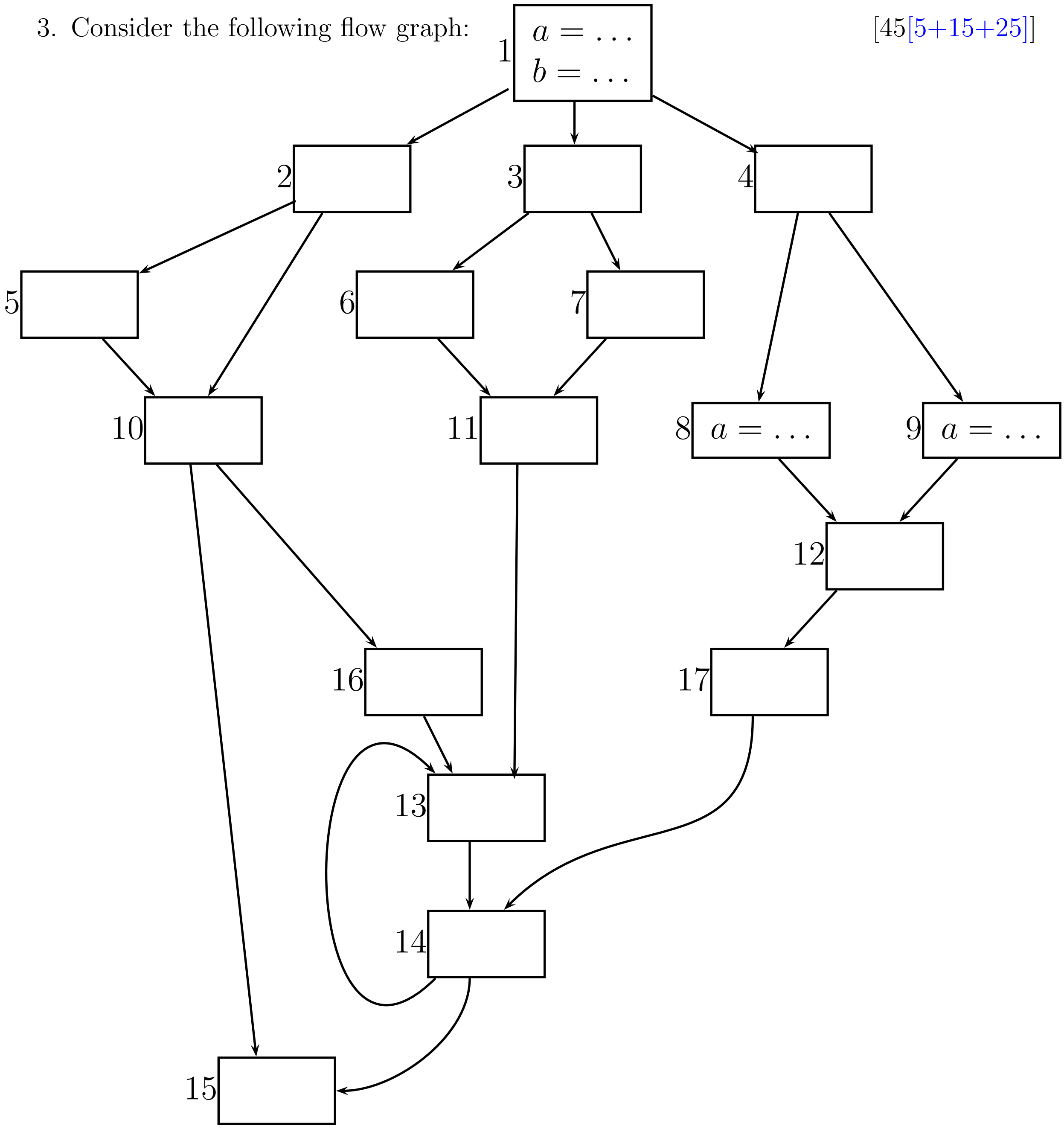
Figure 1: An Example Program

(a) Design a data-flow analysis to compute SUDs for definitions present in a program. Recall that you have to talk about the 4 components $\langle D, S, \wedge, F\rangle$. Describe the lattice $\langle S, \wedge\rangle$ in details, with the help of a lattice diagram. You also need to describe flow functions for various statements of interest.

(b) Is your analysis guaranteed to terminate? Justify.

(c) Give one application of this analysis.

3. Consider the following flow graph: [45[5+15+25]]



Block #1 is the Entry block.

(a) Draw the dominator tree for the graph.

(b) Calculate the dominance frontier for each block.

(c) Convert the flow graph to minimal SSA form. Show the important steps in conversion.