CS243 Midterm Examination

Winter 2006-2007

You have 1 hour 15 minutes to work on this exam. The examination has 70 points. If you spend about one minute for each point, you will have five minutes to spare. Please budget your time accordingly.

Write your answers in the space provided on the exam. If you use additional scratch paper, please turn that in as well.

Your Name: _____

In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

Signature: _____

Problem	Points	Score
1	13	_____
2	12	_____
3	20	_____
4	25	_____
Total	**70**	_____

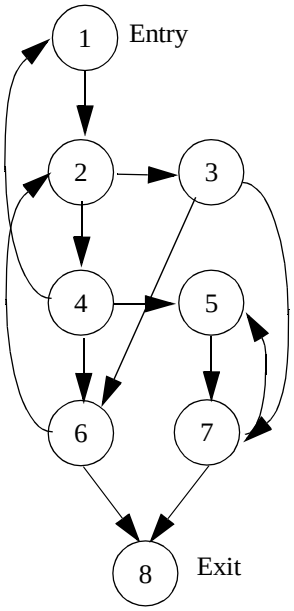
1. (13 points) True or false. Explain your answer.
   a. (3 points) If a node n_1 dominates n_2, n_1 is always visited before n_2 in a depth first search.

   b. (3 points) If a node n_1 dominates n_2, n_1 is always visited before n_2 in a reverse post ordering.

   c. (3 points) Given a monotone data flow framework, had all the interior points of the data flow solver been initialized with the "bottom" of the semi-lattice, the answer at every point in every program would have been "bottom".

   d. (4 points) Given a machine with 5 registers, Chaitin's register coloring algorithm (the one presented in class) will succeed in coloring any program that has no more than 4 active live ranges at any point in the program.

2. (12 points) Please show all work for partial credit. Consider the following flow graph:



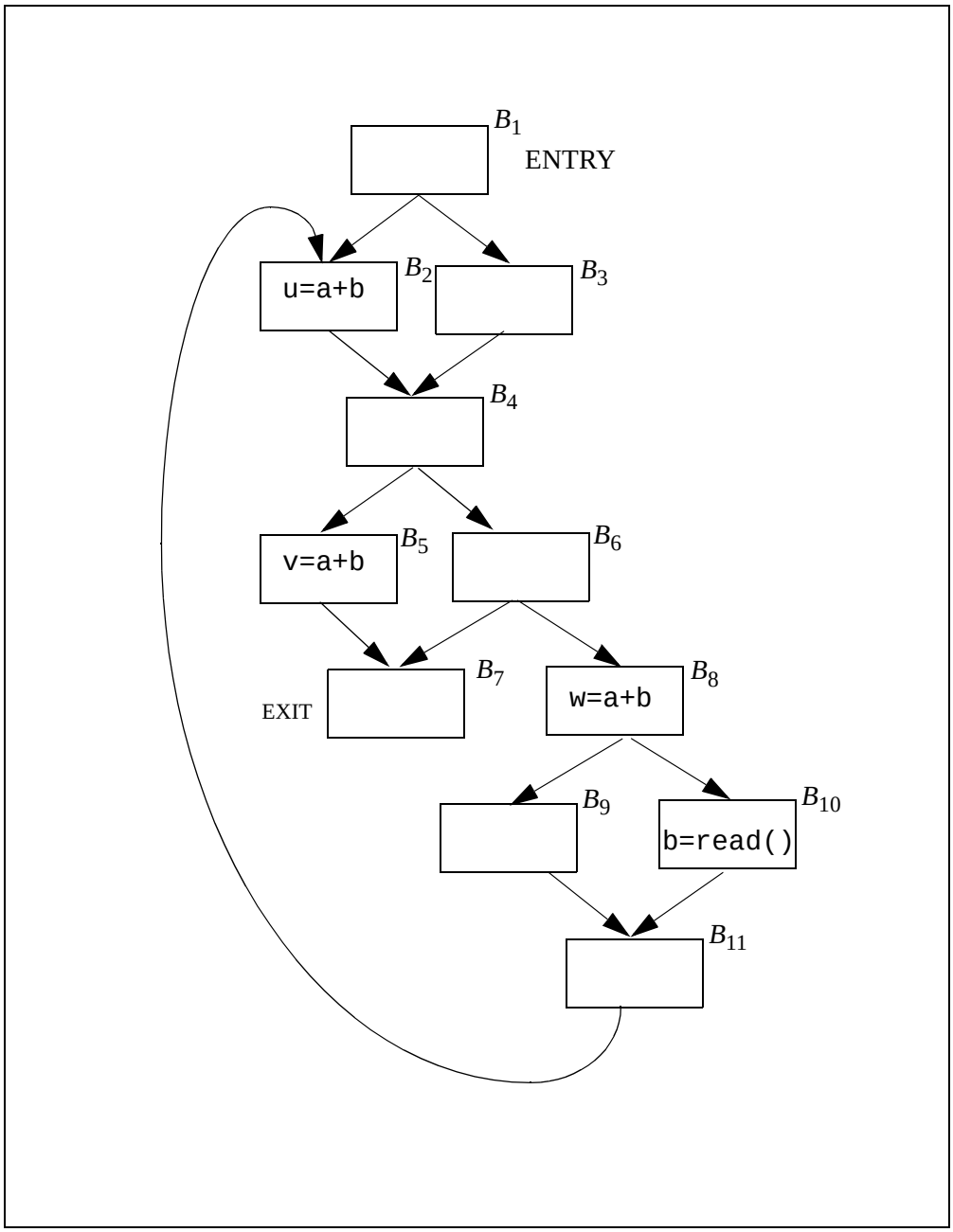
a. Show the dominator tree for this flow graph.

b. What are the back edges in this flow graph?

c. Find the natural loop associated with each of the back edges.

d. Is the flow graph reducible?

3. (20 points) Show the result of applying lazy code motion to the following program. Just show the result of the optimization -- it is not necessary to show any intermediate steps.



4. (25 points) Critical sections are a very fundamental concept in concurrent programming. A communicating process must issue a LOCK operation before entering a critical section and issue an UNLOCK operation on exit in order to guarantee mutual exclusion.

   For simplicity, assume that there is only one lock in the system, and therefore the LOCK and UNLOCK operations do not have any arguments. You can ignore all the other operations in the program. In a correct execution sequence, every LOCK operation must be immediately followed by an UNLOCK operation, and every UNLOCK operation must be preceded by a LOCK operation.

   Your task is to develop a static program analysis that will issue the following four types of warnings:

   - Warning I: LOCK $\rightarrow$ LOCK. Issue a warning on a LOCK operation if it can potentially follow another LOCK operation.
   - Warning II: UNLOCK $\rightarrow$ UNLOCK. Issue a warning on an UNLOCK operation if it can potentially follow another UNLOCK operation.
   - Warning III: LOCK $\rightarrow$ EXIT Issue a warning on a LOCK operation if the program may terminate without performing an UNLOCK.
   - Warning IV: ENTRY $\rightarrow$ UNLOCK. Issue a warning on an UNLOCK operation if it may be executed before any LOCK operations.

   You may assume that there are no procedure calls in the input programs. Describe your analysis. You may want to define one or more data flow algorithms to solve this problem. If you solution uses data flow analysis, specify the algorithm fully by answering the following questions:

   i. What is the direction of your data flow analysis?

   ii. What is the set of values in the semi-lattice?

   iii. Draw a diagram of the lattice, identifying the top and bottom elements clearly.

   iv. How would you initialize the iterative algorithm?

   v. Define the transfer function of a basic block.

   vi. How would you initialize the information at the entry/exit nodes?

   vii. Is your data flow framework monotone? (No explanation is necessary).

   viii. Is your data flow framework distributive? (No explanation is necessary).

   ix. Will your algorithm necessarily converge? If so, why?

   Make sure you specify how you identify the operations that have raised the warning.

*** Extra Space ***