

CSCI 565: Compiler Design and Implementation Spring 2006

Second Midterm Exam (a.k.a. Final Exam)

April 25th, 2006

INSTRUCTIONS

1. This exam lasts 2.5 hours for a total of 105 points covering 5 problems.
2. The exam is closed-book but you may have two pages of memory-enhancing notes.
3. Clearly identify your work by writing your NAME and ID on ALL pages you turn in.
4. Staple all your pages together and write how many pages are there on the front page.
5. You must use a blue or black pen. Pages written using pencil will be considered as draft notes and will not be graded.

STUDENT NAME: _____

STUDENT ID: _____

Total Number of Pages: _____

Grades:

Problem 1 out of 25: _____

Problem 2 out of 20: _____

Problem 3 out of 40: _____

Problem 4 out of 10: _____

Problem 5 out of 10: _____

Total out of 105: _____

STUDENT NAME: _____

STUDENT ID: _____

Problem 1: Loops and Loop Optimizations [25 points]

Consider the code depicted below for a function with integer local variables i , a , b , c , d and an integer input parameters p and n .

```
int i, a, b, c, d;

1:   i = 0;
2:   a = 1;
3:   b = i * 4;
4:   c = a + b;
5: L1: if ( i > n ) goto L2
6:   c = a + 1;
7:   i = i + 1;
8:   b = i * 4;
9:   if (c <= p) goto L3
10:  e = 1;
11:  c = e - b;
12:  a = e;
13:  goto L4
14: L3: d = 2;
15:  c = d + b;
16:  a = d;
17: L4: goto L1
18: L2: return;
```

For this code fragment determine:

1. [5 points] The Control Flow Graph (CFG).
2. [10 points] The dominator tree and the loops (identify the back edges).
3. [5 points] Use constant propagation to the statements in the body of the loop identified in (2).
4. [5 points] Are the statements in line 8 and 16 loop invariants? Explain and describe where can they be moved to if they can in fact be moved outside any of the loops.

STUDENT NAME: _____

STUDENT ID: _____

Problem 2: Register Allocation [20 points]

```
1:   i = 0;
2:   a = 1;
3:   b = i * 4;
4:   c = a + b;
5: L1: if (i > n) goto L2
6:   c = a + 1;
7:   i = i + 1;
8:   b = i * 4;
9:   if (c <= p) goto L3
10:  e = 1;
11:  c = e - b;
12:  a = e;
13:  goto L4
14: L3: d = 2;
15:  c = d + b;
16:  a = d;
17: L4: if goto L1
18: L2: return;
```

For the code (before any transformation) in Problem 1, replicated above, determine the following:

1. [5 points] The live ranges for the local variables i , a , b , c , d , e using the live variable analysis as the source for the information about the live ranges. You do not need to solve the iterative data-flow analysis problem so just derive a solution by inspecting the code.
2. [5 points] The interference graph using the simplest definition of interference where the webs include the live range in terms of the line numbers where an access to a variable (either a read or a write operation) occurs.
3. [5 points] Determine the minimum number of registers needed (without spilling, of course).
4. [5 points] Assuming that you were short of one register, which register(s) would you spill and at which points in the program would you insert the spill code? Explain.

STUDENT NAME: _____

STUDENT ID: _____

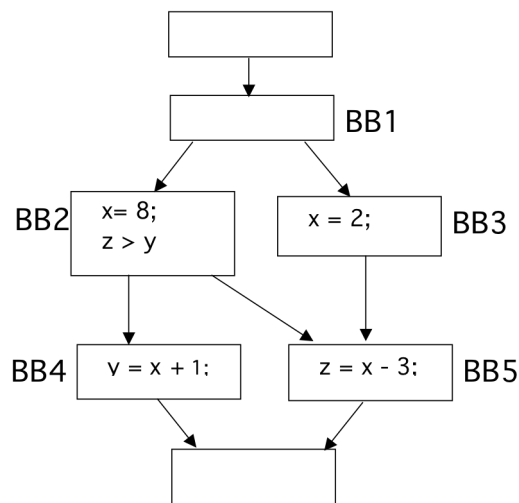
Problem 3: Iterative Data-Flow Analysis [40 points]

Your goal is to solve the Upwards Exposed Reads data-flow analysis problem.

Definition: *We say a variable is upwards exposed at a particular program point p if there is a definition at program point q that reaches that use at p .*

This is essentially the dual problem of the reaching definitions problem studied in class and aims at finding what uses of a variable are reached by a particular definition of that variable.

For the example in the CFG below the definition of x in basic block BB2 reaches the uses in BB4 and BB5 whereas the use in BB5 is reached by the definitions in BB2 and BB3.



Describe your approach to the Upwards Expose Use data-flow analysis by answering the following questions:

- [5 points] What is the set of values in the formulation and the initial values?
- [10 points] What is the direction of the problem, backwards or forwards and why?
- [10 points] What is the join function for this data-flow problem, *i.e.*, how to gather the information from multiple control flow paths?
- [10 points] How do you construct the transfer function of a basic block, based on the GEN and KILL typical formulations?
- [5 points] Why does an iterative approach works for this formulation, *i.e.* what property of the meet and join functions do you rely on to guarantee termination? Please explain.

STUDENT NAME: _____

STUDENT ID: _____

Problem 4: Single-Static Assignment Form [10 points]

1. [8 points] In a short paragraph, provide an intuitive explanation of how to rewrite a program into minimal SSA form. In particular, describe the motivation for Φ -node placement. Why are they placed where they are? We would like an answer that shows you understand the concepts, and not just a list of steps repeated from the lecture notes. Use examples when useful to illustrate your points.
2. [2 points] Describe a use of SSA (in some detail) in program optimization.

Problem 5: Instruction Scheduling and Data Dependence [10 points]

1. [3 points] What is the ultimate goal of the instructions scheduler?
2. [3 points] Other than data dependences what are other restrictions a scheduler has to face with when changing the order in which the instructions are dispatched?
2. [4 points] Briefly describe the types of data dependences and how they can be eliminated, if at all, to improve the ability of a scheduler to improve its results.