CS 701

Final Exam

Wednesday, May 15, 1996

12:25 - 2:25

1325 CSST

Instructions

Answer question #1 and any three others. (If you answer more, only the first four will count.) Point values are as indicated. Please try to make your answers neat and coherent. Remember, if we can't read it, it's wrong. Partial credit will be given, so try to put something down for each question (a blank answer always gets 0 points!).

1. (1 point)
   A dominator tree encodes what relationship among basic blocks?

2. (33 points)
   A common compiler optimization is *copy propagation*. Assume we have an assignment of the form `x = y` where `x` and `y` are simple scalar variables (no subscripts, pointers or expressions are allowed). Consider all the uses of `x` that this assignment reaches. If it is the case that

   (a) If this assignment reaches a particular use of `x`, it is the *only* assignment to `x` that reaches that use.

   (b) On all paths from this assignment to a use of `x`, there are no assignments to `y`.

then the assignment can be removed and all uses of `x` that the assignment reached can be replaced with uses of `y`.

We can easily check for condition (a) using a standard reaching definitions analysis. Formulate a data flow problem that allows us to check whether condition (b) holds. Be use to specify the direction of the analysis (forward or backward), the lattice of values you use, the transfer function used in each basic block, the meet operator and initial values. Assume the program *has not* been put in SSA form and that all variables are initialized prior to use.

This data flow problem can be formulated in many different ways. *Any* reasonable formulation is acceptable.

3. (a) (23 points)

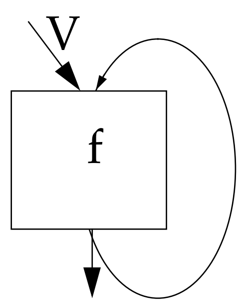
A key step in putting a program in SSA (static single assignment) form is placement of the ϕ functions. Assume that all assignments to variables have been rewritten so that each variable is now assigned to exactly once. Give an algorithm to determine, for a set of variables $x_0, \ldots, x_n$ derived from a single variable x, where to place ϕ functions and the appropriate arguments for each ϕ.

(b) (10 points)

Show that once a program is in SSA form, copy propagation, as defined in question 2, is very easy. In particular no analyses beyond those needed to put a program in SSA form are needed.

4. (33 points)

What properties must a data flow problem satisfy to be *rapid*? Show that if a particular data flow problem is rapid, then the following fragment of a control flow graph can be evaluated iteratively in only two passes (input value **V** is fixed, **f** is the block's monotone transfer function).



5. (33 points)

Assume we are doing partial redundancy elimination. What is the criterion used to determine when a calculation can be removed? Explain why removing computations using this criterion is *safe*. That is, why can we be sure that the computations we have inserted into predecessors have transformed partial redundancies into full redundancies whereever computations are removed?

6. (a) (18 points)

Give an algorithm that solves a forward data flow problem by iteratively visiting basic blocks in some fixed node order O.

(b) (15 points)

Assuming order O has no special properties, what is the worst-case number of passes that will be needed, assuming n nodes and a lattice of height h. How does your worst case bound change if order O is the depth-first ordering that is usually recommended?