CSE 231 Spring 1998

Project, Part 2 Due Thurs May 14

This aim of this part of the project is to construct SSA form for a Simple SUIF program, using the algorithm discussed in class and in the paper we read.

- **Variables included in SSA construction**: In general, aliasing of variables can make SSA messy. Fortunately, in Simple SUIF, variables in pseudo-registers are guaranteed not to be aliased. Therefore, in our project, variables are pseudo-registers.

- **BB Def-Use Info**: For each procedure, your will need to find the variables defined and used in a basic block. One alterative is to construct a separate data structure to hold these, together with a link back to the Simple SUIF instruction in which the def or use occurs (this can be done when you construct your CFG). Another alternative is to find them on-the-fly in the Simple SUIF instruction stream. Note that, for a procedure call, there are no defs of variables within the procedure, because of call-by-value parameter passing and the fact that variables cannot be aliased. The only uses in a procedure call are the actual parameters.

- **Dominance Frontiers**: For each procedure, your project should find the Dominance Frontier sets of each node in the CFG for the procedure.

- **Placing the minimal number of Phi functions and Renaming**: Here again there are alternative representations. One alternative is a separate data structure where the phi functions are placed into the beginning of the appropriate basic blocks in the CFG and the variables renamed. The second alternative is to amend the Simple SUIF instruction stream. You can create new registers with `new_register`. This can take the place of renaming. You can insert calls to a dummy procedure `Phi` to stand for the Phi function you want to insert, or you can create a new instruction with `new_instr`. In making your design, you need to keep in mind that the final phase of the project will use the representation of SSA form created in this part to perform some optimization and then generate Simple SUIF instructions. There will be no references to Phi functions in the final code.

Specification of Output:

- As before, a pair and a triple are ordered lists. Otherwise, lists may be output in any given order (i.e., lists are sets).

- Important!: To simplify and standardize your output, we are providing output functions. You are required to use these. For example, they will automatically provide basic block names. Your program should print its output to “standard output” through this output library. (You may still print things to “standard error”, though they may be ignored, and you can print any amount of white space to “standard output”). You may find this library at /cse/class/sp98/cse231/software/, the files are names.c and names.h. For those of you worried about wasting time integrating this into your previous project, Nick has done much of it for you. You are free to make your own modifications, if you don’t trust ours.

Your program should output for each procedure

1. The Dominance Frontier (DF) sets. A DF set description should be output as a pair, whose first element is the basic block name, whose second element is a list of basic block names in the dominance frontier of the first element. You should output a list of DF set descriptions in the program.

2. A Variable-Phi function placement list. A Phi Function placement description for a variable is a pair. The first element is the variable name. The second element is a list of basic block names where a Phi function for the variable is inserted during your construction. You should output a list of all Phi function placement descriptions for the variables included in SSA construction.

Submission Procedure: Check the web page for instructions on submitting your project.