UNIVERSITY OF CALIFORNIA
Department of Electrical Engineering
and Computer Sciences
Computer Science Division

CS 264 **S. L. Graham**
Fall 1995 **A. B. Downey**

Assignment 6: Compiler Support for Debugging

Due: Friday, 8 December 1995

The purpose of this assignment is to evaluate the approaches to debugger support that we have been discussing. Please give a hardcopy of your solution to Allen.

1 Investigate Debugging for Your Cadaver

There are two aspects to this question — to look at the debugging support that exists for your cadaver, and to look at the needs.

- What kinds of optimization are done by your cadaver (e.g. basic block optimization, optimized register allocation, inlining, general 'global' optimization, pipeline reordering, etc.)? Explain how you determined that information.
- Which of the optimizations done by your cadaver affects the debugger's ability to provide data values? Manage code location?
- Are there other optimizations that are not done in this particular compiler but that make sense for the source language? Which ones would create problems for the debugger? (Note that byte-code compilers can benefit from many optimizations that are independent of hardware.)
- What debugger is available for your compiler? How powerful is it; e.g. can programs be modified or only inspected? What kinds of breakpoints can be set? Are the capabilities different in the case of a breakpoint than in the case of an interrupt?
- What are the performance penalties for debugging? Must the program be recompiled? Does it run significantly more slowly? Are there particular debugger features (maybe conditional breakpoints) that suffer significant performance hits?
- How is information shown to the user? Can the user have a continuing dialogue about the program (e.g. set subsequent breakpoints, roll back to earlier stages of execution, inspect call stacks and parameter values)?
- How does the debugger actually cause a breakpoint to occur?

- To what extent does the debugger support any of the capabilities discussed in class, with respect to optimized code or with respect to breakpoints? Explain how you determined your answer.

Note from Allen: depending on your knowledge of optimization and the complexity of your cadaver, this assignment could turn out to be substantial. It is up to you to find a reasonable compromise between diligence and attention to other obligations.

2 Formulate your opinions

The goal of research in debugging is to help the user to understand what's happening in the execution of the program, usually in order to fix something that's flawed. The two major directions we've looked at are providing information about optimized programs, and interrupting programs at particular execution states. Based on your experience as a programmer, and on the material we've covered concerning the state-of-the-art, what do you see as feasible improvements that might be made to debuggers for these purposes. (You need not restrict your consideration to your cadaver and its debugger). What additional research needs to be done? Be sure to relate your answer to concrete examples from the papers and from debuggers you have used.