Milestone #6: Optimizations

Overview

For this milestone, you will add two optimization (tranformation) passes to your compiler. The first will propagate constants and fold constant calculations. The second will remove useless code.

Constant Propagation

Implement either Sparse Simple Constant Propagation or Sparse Conditional Constant Propagation. These were presented in lecture and are discussed in the recommended text book. Links to pseudocode and the relevant research paper are included on the course website.

Useless Code Elimination

Implement one of the following forms of Useless Code Elimination. (Note that each of these is possible without SSA, but the SSA representation simplifies the implementation.)

- SSA-based Unused Result: Assuming an SSA register "graph" (i.e., each register tracks its uses), if a register is defined but never used, then the defining instruction may be useless. Use this information to remove all such instructions as long as they are truly useless (i.e., it is not strictly safe to remove a call instruction if the result is not used since the function being called may have a side-effect).

  The removal of one instruction may cause another instruction to be rendered useless. Continue the removal process until there are no such useless instructions (this is similar to removing trivial ϕ instructions).

- (Relaxed) Critical Instruction Removal: Implement the Critical Instruction ("mark-and-sweep") useless code removal algorithm discussed in lecture.

  Recall that this algorithm initially marks, within the cfg, only those instructions that are considered inherently "critical". Continue marking every instruction on which a marked instruction depends (i.e., mark instructions that define registers used by a mark instruction). After all "reachable" instructions are marked, remove all remaining instructions.

Command-line Options

You should provide command-line options to enable/disable your optimizations (you can choose either as the default). This will simplify comparisons of the optimized and unoptimized versions.