

CMPSC471 2nd Programming Project: Register Allocators

(due = December 3rd, 2018; 11:55PM)

In this programming project, you will implement two register allocators for a basic block (also called *local* register allocators). In the first one, *top-down*, you will adopt a metric beforehand (it is up to you what metric you employ) and use it in deciding which variables to assign to registers. In the second one, *bottom up*, you will allocate registers to variables as you process each instruction (in a greedy fashion). When you consume all your registers, you will select a victim variable/value and spill it into memory.

Inputs to both of your allocators will be the number of registers (K) and a set of instructions (that constitute the basic block). All instructions will be of type “Op vi, vj, vk”, where “op” is a generic operator and “vi”, “vj” and “vk” are the *variables* (values) to be assigned to either registers or memory. An example basic block would look like this:

Op v1, v2, v2

Op v3, v1, v4

Op v1, v5, v5

Op v7, v6, v3

Op v1, v1, v1

Op v4, v6, v4

The output of your allocator will be (i) the set of variables allocated to registers and (ii) the rewritten basic block after the register allocation. You can use any software to parse the input basic block (or write your own) and any programming language to implement your allocators.