Exam 4 - 628 Cover Sheet

I affirm that I have completed this exam without unauthorized assistance from any person, materials, or device. I also affirm that I have completed the exam according to the restrictions listed in the exam document.

Signed: ______________________________________

Print name: ___________________________________

Date: ____

Please submit a signed and dated copy of this cover sheet as the first page of your exam submission.

You will not receive credit for the exam unless your submission includes this (signed and dated) cover sheet.

601.628 Compilers and Interpreters Fall 2020, Exam 4 - 628

Due: Friday, December 18 by 5:00pm Baltimore time (UTC-05:00)

The permitted resources for this exam are:

- The textbook(s)
- Materials posted directly on the course website (e.g., slides)

Do *not* use any resources other than the ones explicitly noted above.

You may *not* write program(s) or use automated calculation devices or programs. You will need to do required calculations by hand. In other words, this is a “pencil and paper” exam, but you can (and should) type your answers. (Neatly handwritten answers are also fine.)

Do *not* discuss the exam with anyone else: your answers must be your own answers.

You will submit your answers to Gradescope as “Exam 4 - 628” in PDF format. When you upload to Gradescope, you will need to select the page of your submitted document corresponding to your answer to each question. You may use software (word processor, LaTeX, etc.) to prepare your answers.

Important: Make sure the first page of your submitted document is the signed and dated cover sheet (which is the first page of this exam document.) You will not receive credit for the exam if your submission does not include the (signed and dated) cover sheet.

Important: Show your work, and justify your answers. “Bare” answers (without supporting work or justification) may not receive full credit.

[Questions begin on next page.]

Question 1. [36 points] Consider the following basic block consisting of high-level instructions:

```
localaddr vr6, $160
muli vr7, vr0, $8
addi vr8, vr6, vr7
localaddr vr9, $0
muli vr10, vr0, $8
addi vr11, vr9, vr10
ldci vr12, $10
ldi vr13, (vr11)
muli vr14, vr13, vr12
sti (vr8), vr14
ldci vr2, $1
addi vr3, vr0, vr2
mov vr0, vr3
```

Assume that

- The leftmost operand is always the destination
- Integer values, pointers, and virtual registers are 8 bytes
- `localaddr` loads the address of a local variable at a specified storage offset into a virtual register
- `muli` and `addi` are integer multiplication and addition, respectively
- `ldci` loads an integer constant into a virtual register
- A parenthesized register (e.g., "`(vr11)`") indicates a memory reference
- `ldi` loads an integer value from memory into a virtual register
- `sti` stores the integer value of a virtual register in memory
- `mov` copies a value from one virtual register to another

(a) [27 points] Perform local value number analysis on this basic block. At the point immediately after each instruction show:

- The mapping of virtual registers to value numbers
- The mapping of value numbers known to have a constant value to their constant value
- If the instruction is a def (assignment to a virtual register), the value number being assigned to the destination virtual register

Value numbers should be integers. Introduce new value numbers as necessary to represent unknown values. As an example of how to write the information known after each instruction, $\boxed{\{\text{vr12} \rightarrow \text{VN6}\}, \{\text{VN6} \rightarrow 10\}}$ means that vr12 contains value number 6, and value number 6 represents the constant value 10. Note that these mappings will generally contain multiple entries, i.e., if two virtual registers have been assigned a value, then mapping of virtual registers to value numbers should contain two entries.

(b) [9 points] Show which def instructions in the basic block could be replaced with an `ldci` or `mov` instruction to store a value currently available as a constant or in a virtual register.

Question 2. [36 points] Consider the following basic block of high-level instructions (with the meaning of the different kinds of high-level instructions being the same as in Question 1):

```
localaddr vr19, $4000000
muli vr20, vr0, $4000
addi vr21, vr19, vr20
muli vr22, vr1, $8
addi vr23, vr21, vr22
ldi vr4, (vr23)
localaddr vr24, $2000000
muli vr25, vr2, $4000
addi vr26, vr24, vr25
muli vr27, vr1, $8
addi vr28, vr26, vr27
ldi vr29, (vr28)
muli vr30, vr3, vr29
addi vr31, vr4, vr30
mov vr4, vr31
localaddr vr32, $4000000
muli vr33, vr0, $4000
addi vr34, vr32, vr33
muli vr35, vr1, $8
addi vr36, vr34, vr35
sti (vr36), vr4
ldci vr5, $1
addi vr6, vr1, vr5
mov vr1, vr6
```

Perform local register allocation for all virtual registers *except* for vr0, vr1, vr2, and vr3. For each of the instructions, indicate the current assignments of machine registers for each virtual register that has been allocated a machine register. Assume that 3 machine registers are available, called MR0, MR1, and MR2. Show where register spills and restores are needed. For each spill and restore, indicate

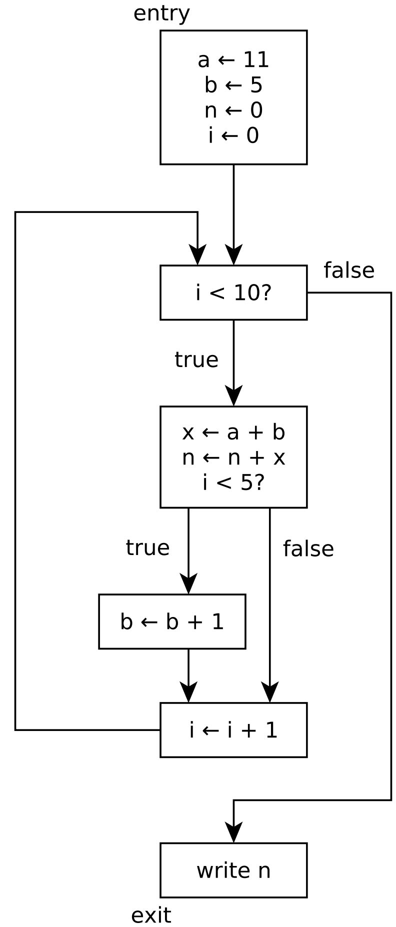
- The virtual register
- The machine register
- The spill location

You can assume that spill locations are numbered by integers, and that as many spill locations as necessary may be used.

[Exam continues on next page.]

Question 3. [18 points] Let's say that you want to implement a dataflow analysis to determine which variables contain a constant value (and if so, which constant value) at each point in a control-flow graph. This will be a forward analysis.

Consider the following control flow graph:



Assume that the dataflow values are sets of facts where each fact is a pair of the form (*variable*, *constant*), such that a set will contain at most one pair for a specific variable. For example, the dataflow fact after the $a \leftarrow 11$ statement at the beginning of the entry block would be $\boxed{\{(\text{a}, 11)\}}$.

(a) [9 points] Describe how sets of dataflow facts are combined. Specifically:

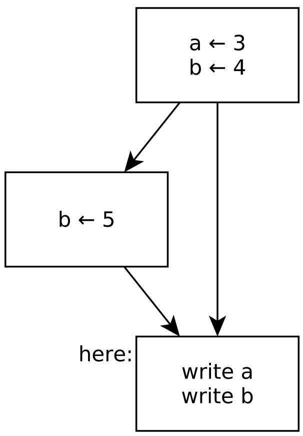
(a.1) As an example, what set of facts would result from combining $\boxed{\{(\text{p}, 2), (\text{q}, 3), (\text{r}, 5)\}}$ and $\boxed{\{(\text{p}, 2), (\text{q}, 4)\}}$?

(a.2) Describe a general procedure for combining sets of dataflow facts

(b) [9 points] Show the sets of facts that are true at the beginning and end of each basic block in the control flow graph. If any block needs to be modeled more than once (e.g., because its beginning fact changed as a result of a loop back edge), indicate which block (or blocks) were affected, and how the data values changed.

Question 4. [10 points] The dataflow analysis you described in Question 3 will not, in general, discover any information about variables with values that aren't guaranteed to be constant at a particular location.

For example, consider the following control-flow graph:



The variable **b** can be known to be positive at the beginning of the block labeled "here".

Describe how to extend the analysis so that it can determine (at least in some cases) which values are positive and negative. The analysis should still be able to determine which variables are constant (and what their values are), but should "downgrade" constant values to positive or negative when necessary. Give some examples of how dataflow values would be combined.