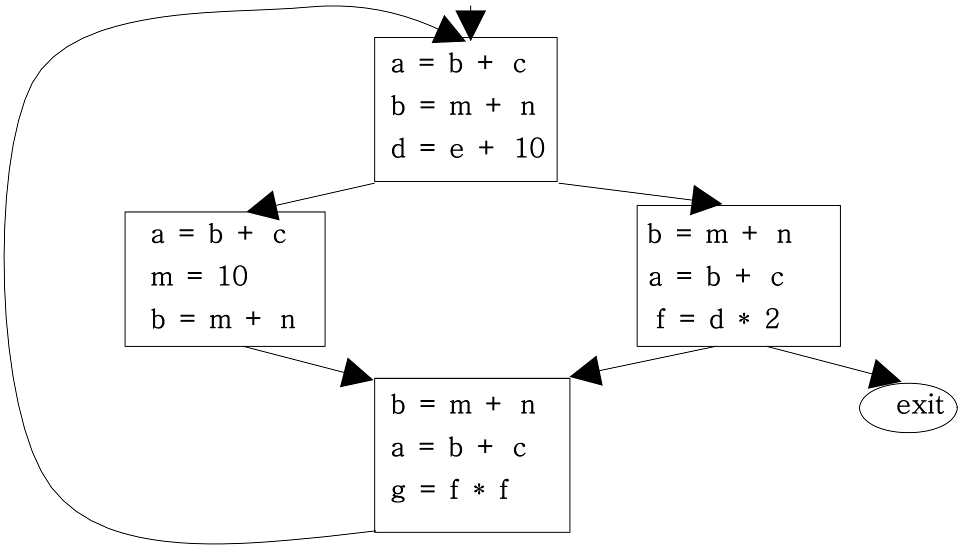
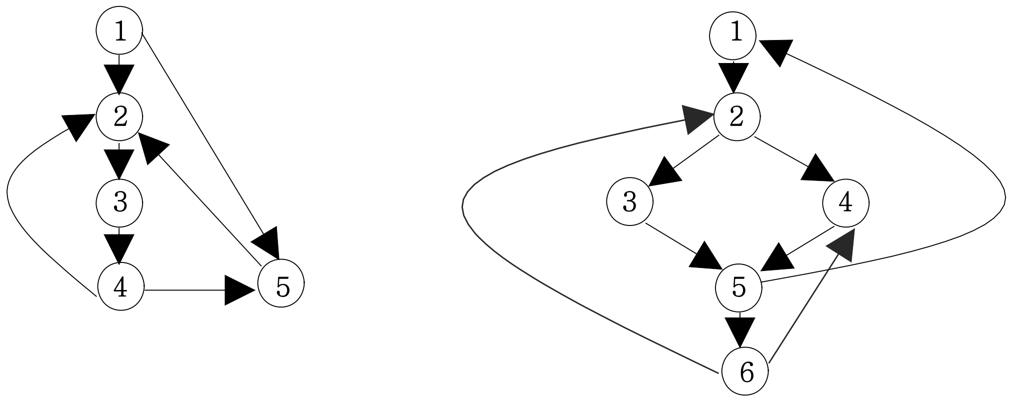
Midterm Exam of Advanced Compilers (2008)

(1) Show the result code after global CSE (common expression elimination), followed by LICM (loop-invariant-code-motion) for the following example.



(2) Find all natural loops from the following control flow graphs



(a) (b)

(3) Find induction variables in the following program. Show the result code after applying strength reduction and loop test replacement.

```
      i = 1
top:  i = i + 1
      d = i - 2
      a = *c
      if a > 0 goto L
      i = i + 3
      e = 2 * i
L:    k = i * 2
      if i > 10 goto top
```

(4) We consider a language that allows only boolean variables which can hold either TRUE or FALSE value, and allows only the following four statements:

a = TRUE, a = FALSE, a = b, a = NOT b

For each point of the program, we want to determine if a variable can possibly (not definitely) hold a TRUE value. Devise a data flow framework for this purpose.

(*a*) The domain of values for one variable is {}, {T}, {F}, {T, F}. Draw a diagram for the lattice for one variable with these values. Mark the top and the bottom.

(*b*) For which values can we say that the variable can possibly hold a TRUE value?

(*c*) What is your meet operator?

(*d*) Is this a forward problem or a backward problem?

(*e*) Specify the boundary condition and the initialization of the iterative algorithm.

(*f*) Define a transfer function for a basic block when we assume that the basic block has a single statement (we can easily compose these functions to get the transfer function of a whole basic block). For this, let us denote the value of a variable x at the beginning and at the end of a basic block B as IN(B, x) and OUT(B, x), respectively. Then, define OUT(B, x) for each of the following cases:

i) if B does not assign to x

ii) if B contains x = TRUE

iii) if B contains x = FALSE

iv) if B contains x = y

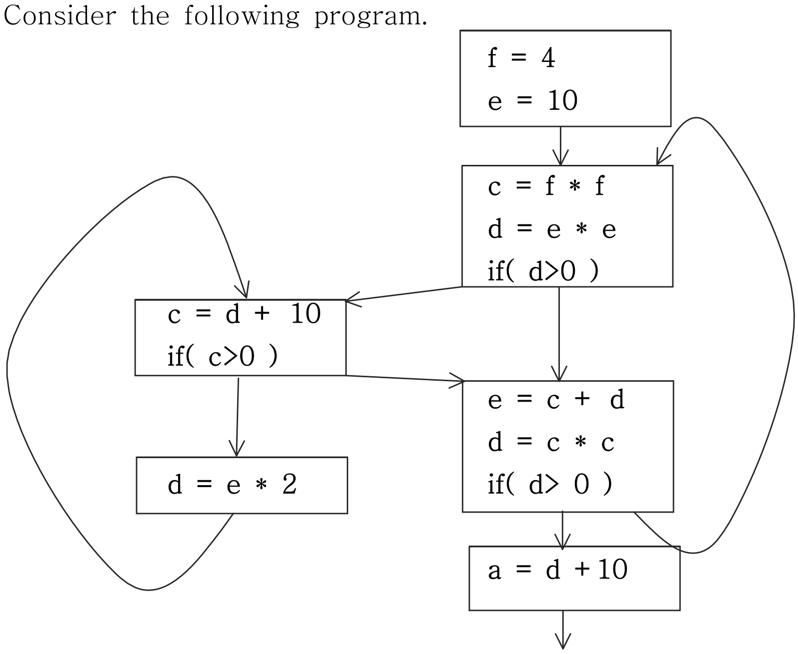
v) if B contains x = NOT y

(*g*) Is the data flow framework monotone? Explain using only one case when B has a single statement a = NOT b.

(*h*) Is the data flow framework distributive? Explain using only one case when B has a single statement a = NOT b.

(*i*) Will the iterative algorithm converge? Explain your answer.

(5) Consider the following program.



(a) Find the live ranges of each variable. Is there a variable that has multiple live ranges? If so, specify which one it is. Replace the operands of each instruction by live ranges such that if there is a single live range for a variable, use the variable name itself as previously, but if there are multiple live ranges for a variable x, use x1, x2, x3, etc, as new operands. Show the replaced code.
(b) Draw the interference graph for this replaced program with the live ranges.
(c) Is there any way to color all nodes in the graph with **three** registers by Chaitin's coloring algorithm? If so, show how it works. If not, state why not and state if the graph can be colored with Briggs's optimistic coloring algorithm. If so, show how it works. If not, state why not.

(6) Let us assume that a load takes **two** cycles while other instructions take a **single** cycle. And, there are no dependences between load and store. Draw a data dependence graph for the following basic block with delay cycles added to each edge. Then, show a schedule with infinite resources.

```
i1 : LD     r2 = 9(r4)
i2 : ST     r4,  0(r3)
i3 : LD     r4 = r2(r3)
i4 : ADD    r1 = r5, r4
i5 : SUB    r2 = r1, 1
i6 : ADD    r5 = r2, r3
i7 : ADD    r6 = r5, r4
```