

ECE540S Optimizing Compilers

University of Toronto, Faculty of Applied Science and Engineering
Department of Electrical and Computer Engineering

Final Exam: April 29, 2003

35% of the final grade

150 Minutes

No Calculators, No Books, No Notes

Student Name: _____

Student Number: _____

Email Address: _____

Score: Q1 = / 20
 Q2 = / 20
 Q3 = / 15
 Q4 = / 20
 Q5 = / 15
 Q6 = / 10

TOTAL = / 100

Induction-Variable Optimizations

1) Use the code below to answer parts A and B.

```
      L = 4
      I = 0
L0:   T1 = I > 99
      IF (T1) GOTO L1
      T2 = addr X
      L = L + T2
      J = 4 * I
      K = J - 4
      T3 = T2 + K
      [T3] = L
      I = I + 3
      GOTO L0
L1:
```

A) On the code above, identify all basic induction variables (BIV) and derived induction variables (DIV). To the write of the definition of each DIV, provide its descriptive triplet (f, m, a), where f is the family/class of the variable, m is the multiplicative coefficient and is the additive component.

B) Perform loop invariant code motion, strength reduction (of induction variables), deadcode elimination, copy propagation and induction variable elimination on the code above. Write the updated code below.

Student Name: _____

(C) Draw the dominator tree for the CFG from part (B) (5 marks)

(D) Identify the back-edges in the CFG in part (B). You can list them in the format $x \rightarrow y$, which means an edge starting from x going to y . (5 marks)

Control Flow Analysis (20 marks)

2) Answer parts (A) – (C) using the data flow framework given below. The framework determines which files may be open at each program point. (25 marks).

$$\mathbf{Fin}(b) = \bigcup_{p \in \text{Pred}(b)} \mathbf{Fout}(p)$$

$$\mathbf{Fout}(b) = \mathbf{Open}(b) \cup (\mathbf{Fin}(b) - \mathbf{Close}(b))$$

where the local sets are defined as:

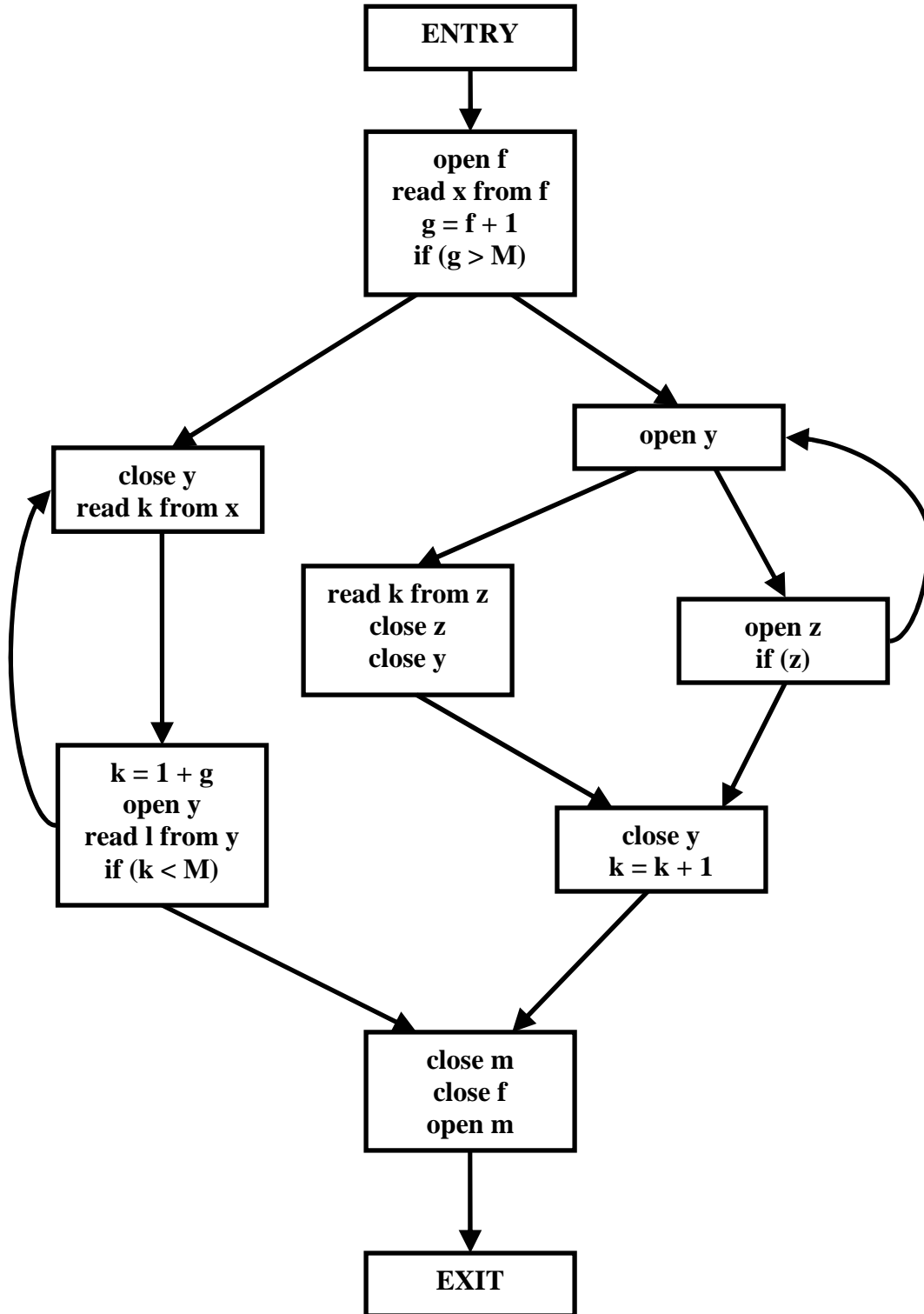
Open(b) is the set of files opened in basic block *b* using an open statement and are not closed by a close statement before the end of the same basic block.

Close(b) is the set of files closed in basic block *b* by a close statement.

(A) What type of data flow problem is this, a (forward/backward any-path/all-path)? (2 marks)

(B) To start solving the above data flow problem, initial conditions must be selected for either the *Fin* or *Fout* sets. First, state below which of these sets need initial conditions for this problem, the *Fin* or *Fout* sets. Next, give two safe choices for the initial condition for this set for any basic block *b* (do not consider the Entry and Exit blocks). (3 marks)

(C) Apply the data flow framework given above to the following control flow graph. Show the final Fin and Fout sets for each basic block. (15 marks)



Optimizations (50 marks)

3) You have been hired to develop a portable compiler (i.e. one that runs on a family of computer systems) and that compiles several languages. You can assume that the following is known about all of the systems/languages:

- 32-bit integers are supported. Underflows and overflows of integers can be ignored.
- integer add, subtract, left shift and right shift operations take 5 cycles, integer multiply operations take 2 cycles, and integer divide operations take 11 cycles.
- floating-point multiply, add, subtract and divide is supported by all systems. All floating-point operations take 20 cycles.

For parts (A) – (E) below, optimize each statement if possible. If you apply a compiler optimization, name the optimization that you have applied. If you do not apply an optimization state your reason. Assume that variables I, J and K are integers and R is a floating-point variable. (15 marks)

A) $R = 1.74596 + 0.11101$

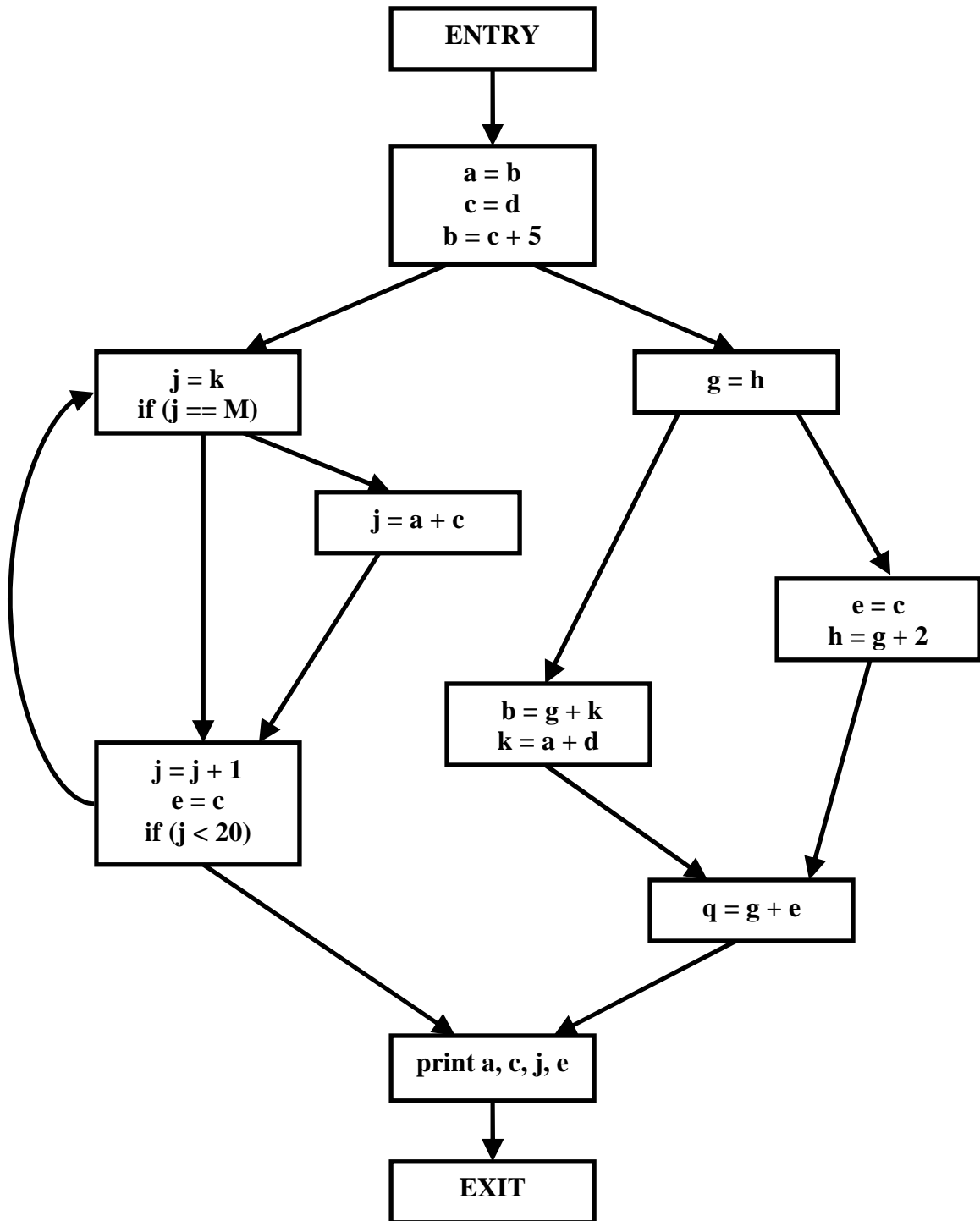
B) $I = J + J$

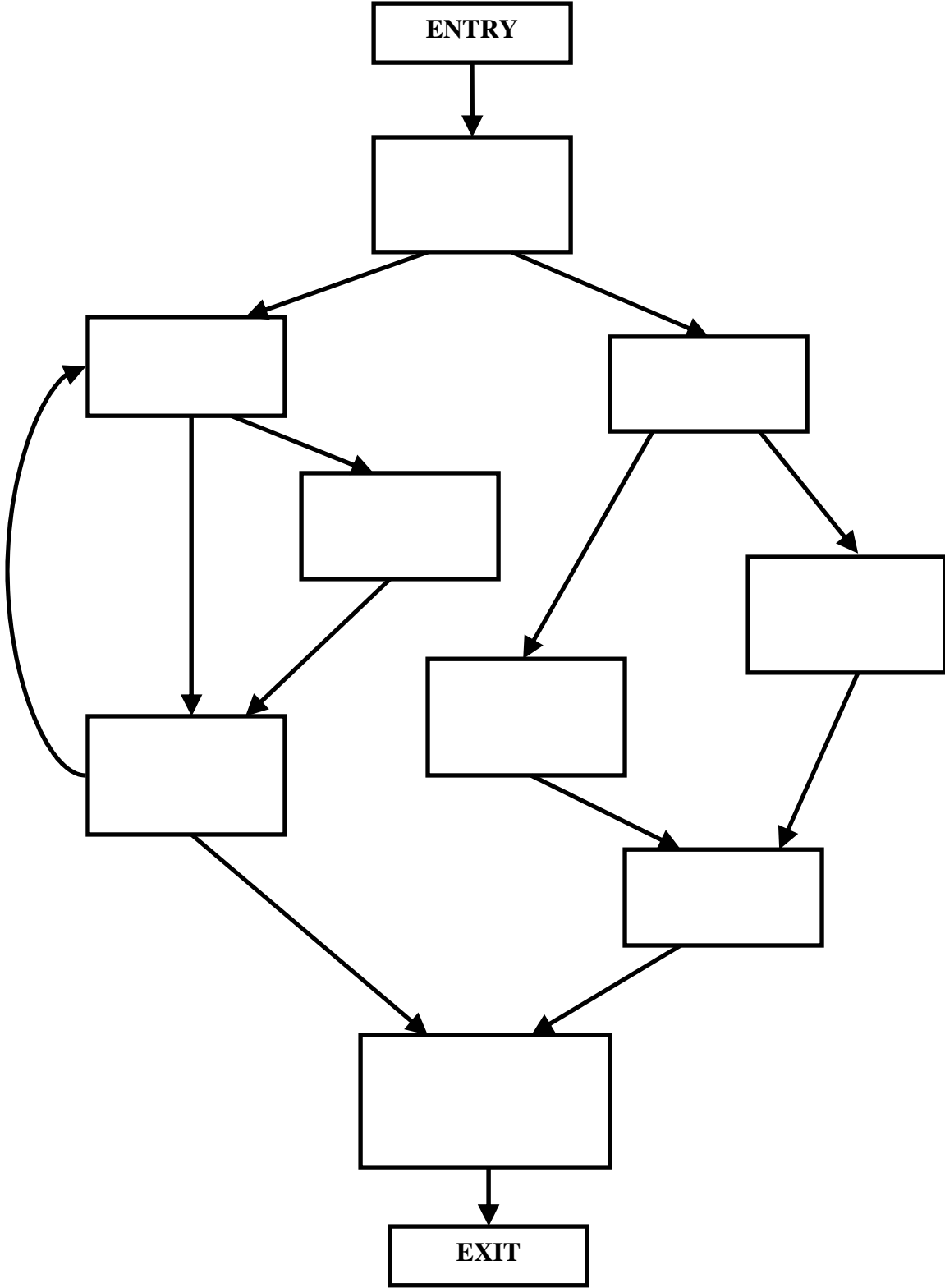
C) $K = I * 3$

D) $J = 10 + 5$

E) $K = (10 + J) / 2$

4) Perform Copy Propagation (CP) on the code below. For the original graph, identify the set of Available Copy Expressions at the input to each basic block. Write these sets on the graph below. Place the code resulting from CP in the graph on the next page (if a basic block is unchanged, you may write SAME in the node). Perform no other optimizations, only CP. (20 marks)





5) Show the following code after performing (A) local value numbering and (B) local common sub-expression elimination. Do not include the changes due to local value numbering when doing CSE, i.e. start from the original code for both part A and part B. You may assume that addition is commutative. (15 marks)

A = J + K
G = J
H = J + K
Q = G + K
A = Q + R
H = R + Q
T = G
K = G + K
H = J + K

A) After local value numbering (perform no other optimizations):

B) After local common subexpression elimination (perform no other optimizations):

6) Provide short answers to the following questions. (10 marks)

A) Your manager/supervisor tells you to implement code hoisting in the group's compiler. After evaluating your work, he complains that his testing shows that programs run no faster with your optimization applied. What do you tell him?

B) Your manager/supervisor tells you to implement both a common sub-expression elimination pass and a copy propagation pass. After you're done, he prepares to add these passes to the group's compiler. He asks you if the compiler should call the common sub-expression elimination pass first, or the copy propagation pass first. What do you tell him?