COL728 Major Exam
Compiler Design
Sem II, 2017-18

Answer all 4 questions Max. Marks: 30

1. Operational semantics: Consider the following statement in a C-like language:

```
//y == 1 and x == 2 at this point
z = (y = x - y) + (x = x + y);
```

This is a valid C statement

a. If before this statement is executed, x == 2 and y == 1, then what according to you is the final value of z? [1]

b. Write the formal operational semantics rules that you applied to arrive at your answer. Your rules should be correct (sound) and complete for evaluating this statement, and should agree with your answer in part a. With each rule, write a few sentences to indicate which rule gets invoked for evaluating which part(s) of the statement above. [7]

2. Register allocation through graph coloring : Explain what is optimistic coloring using an example program. Write an example program (different from the one that was used in class). Draw its register interference graph. And colour it using the optimistic coloring algorithm. Your example should involve at least one case where the “optimistic” nature of the coloring algorithm yields an advantage. Clearly mention at which step did the optimistic nature of the coloring algorithm help obtain a more efficient solution. [6]

3. Dataflow analysis : copy propagation

Recall that copy propagation refers to a transformation pass where occurrences of targets of copy-assignments are replaced with their RHS values and the copy-assignment is eliminated. E.g.,

```
x = y
z = x + 3
w = y + z
```

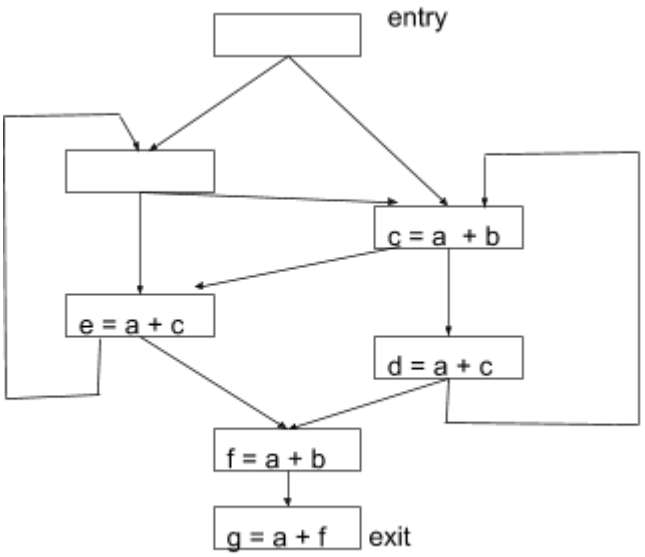
Will be replaced to:

```
z = y + 3
w = y + z
```

In this example: x has been replaced with y through copy-propagation.

Specify an algorithm that implements global copy-propagation. Your algorithm should involve a dataflow analysis followed by a transformation that makes use of the results of the dataflow analysis. Clearly specify the values that you compute using copy propagation. Specify the direction, meet operator, and the transfer functions. If your transfer functions need to know the Use/Kill sets for an instruction, please specify them clearly with examples. Also, mention the boundary conditions. Finally, describe the transformation pass that will use the results of this analysis. [6]

4. Lazy code motion



a. Apply the lazy code motion algorithm described in class to the program above. Show the result of the optimization --- it is not necessary to show any intermediate steps. (Introduce new basic blocks as necessary). [6]

b. Are there redundant operations remaining after the optimization? If not, explain why not. If yes, explain why redundant operations were left even after applying the transformation. [4]