

CS406: Compilers

Homework Assignment 3 - Local Optimizations, Control Flow Graphs, Dataflow Analysis Due: 29/3/2022

The objective of this assignment is to practice applying the concepts underlying:

1. Optimizations involving common subexpression elimination and loops
2. Creation of control flow graphs
3. Liveness analysis and constant propagation

1 Problem Set

1. (2 points) Consider the code shown below:

```
1   A = 7;  
2   B = A + 2;  
3   C = A + B;  
4   D = C + B;  
5   B = C + B;  
6   A = A + B;  
7   E = C + D;  
8   F = C + D;  
9   G = C + B;  
10  H = E + F;
```

- (a) (1 point) Show the result of performing Common Subexpression Elimination (CSE) on the code. Draw a table of two columns, the first column indicates the statement and the second column indicates the set of expressions that are available after executing the statement.
- (b) (0.5 points) Suppose E and C are aliased. How would that change the results of CSE?
- (c) (0.5 points) In the original code, if two variables were to be aliased, performing CSE would add redundancy to the code. Name the two variables. Explain how the redundancy is introduced.

2. (1.5 points) Consider the following code:

```
1   X := 2  
2   Label1:  
3   Y := X + 1  
4   if Z > 8 goto Label2  
5   X := 3  
6   X := X + 5  
7   Y := X + 5  
8   X := 2  
9   if Z > 10 goto Label1  
10  X := 3  
11  Label2:  
12  Y := X + 2  
13  X := 0  
14  goto Label3  
15  X := 10  
16  X := X + X  
17  Label3:  
18  Y := X + 1
```

- (a) (1.5 points) Draw a CFG for the code shown. Also, write the set of leaders.

3. (1.5 points) Perform constant propagation and constant folding optimizations on the code shown in Question 2.a

4. (2 points) Consider the following code:

```

1   x = a * a;
2   c = -1 * x;
3   L1:
4   if (x >= 100) goto L2;
5   z = 2 * c;
6   y = z * x;
7   x = x + 1;
8   goto L1;
9   L2:

```

(a) (0.5 points) Identify loop invariant expression. Can it be factored out of the loop body?

(b) (0.5) What are the basic and mutual induction variable(s)?

(c) (1.5 points) Perform code motion and strength reduction on the code. Show the resulting code.

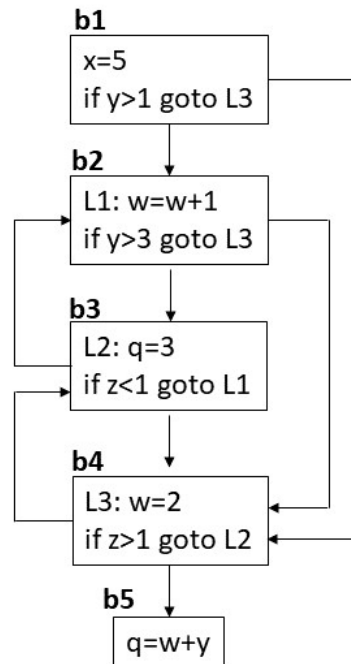


Figure 1: Code with a Single Loop

5. (4 points) Consider the CFG shown in Figure 1. Perform liveness analysis on the CFG. Indicate what variables are live at the entrance and exit of each basic block. Draw a table of three columns, first column is the basic block name, second column is the **LiveIn** set, third column is the **LiveOut** set.

2 What you need to submit?

- A single document named CS406HA3_response_<yourrollnumber>.pdf. E.g., a student with roll number 10002000 would submit the response file CS406HA3_response_10002000.pdf containing:
 - *handwritten* responses to all questions.

3 How to submit?

1. Click on the link shared with you on the discussion forum. This will create a repository in your GitHub account. Clone the repository into your local development environment.
2. Write your responses clearly on a sheet of paper and convert it to digital form (by taking a photo or scanning). Name the file (digitized) as mentioned previously.
3. Add the file to your GitHub local repository using `git add` command.
4. Save the changes using `git commit` command
5. Upload the changes to GitHub using `git push` command
6. Release your changes by first tagging your commit on the *local environment* using `git tag -a cs406ha3submission -m "<release description>"`.

Next, push the tag to *GitHub* with the help of the following command: `git push -tags`

If you want to make changes after you have submitted (repeat the above steps from 2 to 5 and apply modified commands shown below in place of step 6):

```
> git tag -a -f cs406ha3submission -m "<release description>"
> git push -f --tags
```

Make sure that the digital versions of hand-written documents are clearly visible. You will loose points for unclear responses.