

Worksheet-7 solution


From lecture given on 2/4/2019

- Consider the program below with **two critical instructions, geti() and print()**. Write the most efficient code that this program can be transformed to, in terms of minimizing the number of operations performed. List all the analyses and optimizations that you have learned that can contribute to the overall transformation. (You need not to be specify an algorithm to perform the transformation, just the optimizations that could contribute to the transformation.)

```
z = 0;
x = geti();    // critical instruction
do {
    y = 2;
    w = x;
    if (y > 1) y = 3;
    if (y == 3 && w == x) break;
} while ( x > 0 );
print(z);      // critical instruction
```

Case 1 : last line = "print z"

- None of the **defs** inside the loop reach the **use** of z on line 9, so the whole loop is dead code.
- **Def** of z on line 1 is a constant definition, which can be propagated to line 9.
- **x = geti();**
print(0);



```
1. z = 0;
2. x = geti();      // critical instruction
3. do {
4.   y = 2;
5.   w = x;
6.   if (y > 1) y = 3;
7.   if (y == 3 && w == x) break;
8. } while (x > 0);
9. print(z);      // critical instruction
```

Case 2 : last line = "print y"

- **Def** of z on line 1 doesn't reach any **use**. (dead code)
 - Constant **def** of y on line 4 can be propagated to the if-condition of line 6.
 - **Use** of w on line 7 can be replaced with x (Copy propagation from line 5).
1. z = 0;
 2. x = geti(); // critical instruction
 3. do {
 4. y = 2;
 5. w = x;
 6. if (y > 1) y = 3;
 7. if (y == 3 && w == x) break;
 8. } while (x > 0);
 9. print(y); // critical instruction

Case 2 : last line = "print y"

- The if-condition on line 6 now always holds TRUE. We can remove the if-condition on line 6.
- Again, the constant **def** of y on line 6 can be propagated to line 7.
- **Def** of w on line 5 doesn't reach any use (dead code).
- **Def** of y on line 4 doesn't reach any **use** (dead code).

```
1. z = 0;
2. x = geti();      // critical instruction
3. do {
4.     y = 2;
5.     w = x;
6.     if (2 > 1) y = 3;
7.     if (y == 3 && x == x) break;
8. } while ( x > 0 );
9. print(y);        // critical instruction
```

Case 2 : last line = "print y"

- The if-condition on line 7 always holds TRUE. The loop body is executed only once. Since y is only used once in the last line (line 9), the constant def of y on line 6 can be propagated to line 9. The whole loop can then be deleted.

- **x = geti();
print(3);**

```
1. z = 0;
2. x = geti();           // critical instruction
3. do {
4.   y = 2;
5.   w = x;
6.   if (2 > 1) y = 3;
7.   if (3 == 3 && x == x) break;
8. } while ( x > 0 );
9. print(y);           // critical instruction
```