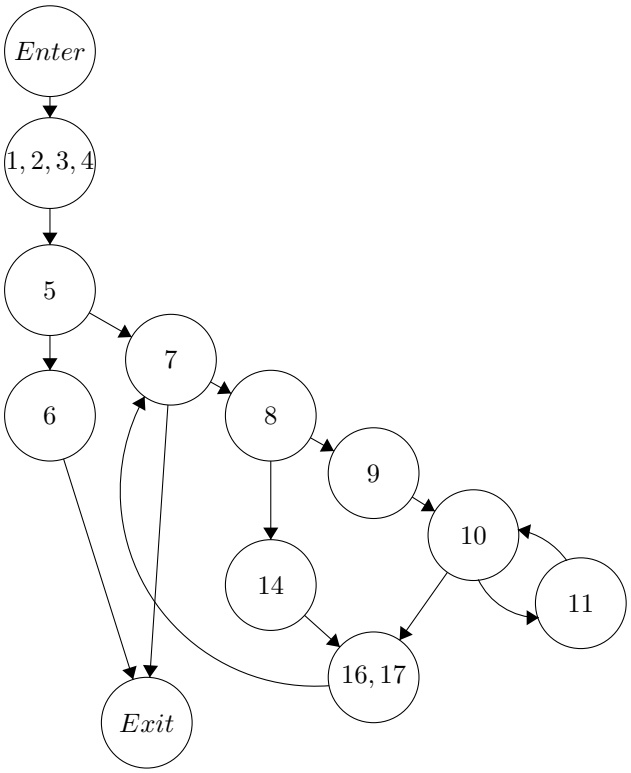
CS510 Homework 1

Sample Solution

February 12, 2015

1 Question 1

1.1 CFG



1.2 Immediate dominators

$dom(5) = \{1, 2, 3, 4, 5\}$
$dom(7) = \{1, 2, 3, 4, 5, 7\}$
$dom(8) = \{1, 2, 3, 4, 5, 7, 8\}$
$dom(9) = \{1, 2, 3, 4, 5, 7, 8, 9\}$
$dom(10) = \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$
$dom(11) = \{1, 2, 3, 4, 5, 7, 8, 9, 10, 11\}$
$dom(14) = \{1, 2, 3, 4, 5, 7, 8, 14\}$
$ipdom(5) = \{Exit\}$

$ipdom(7) = \{Exit\}$
$ipdom(8) = \{16\}$
$ipdom(9) = \{10\}$
$ipdom(10) = \{16\}$
$ipdom(11) = \{10\}$
$ipdom(14) = \{16\}$

1.3 3

Claim. Each node in CFG except entry has a unique immediate post dominator.

Note: many students proved that there cannot be two immediate post dominators, which is not a proof that there is exactly one. I only removed one point for this. By the same logic, I did not accept "there can only be one first" as a complete proof.

If the student showed that there could not be two or zero immediate post dominators, I excepted that answer, assuming it applies to 3 or more with out loss of generality.

For example, a good answer would be:

Supppose a statement s has two IPDOM a and b. Then there are too paths:

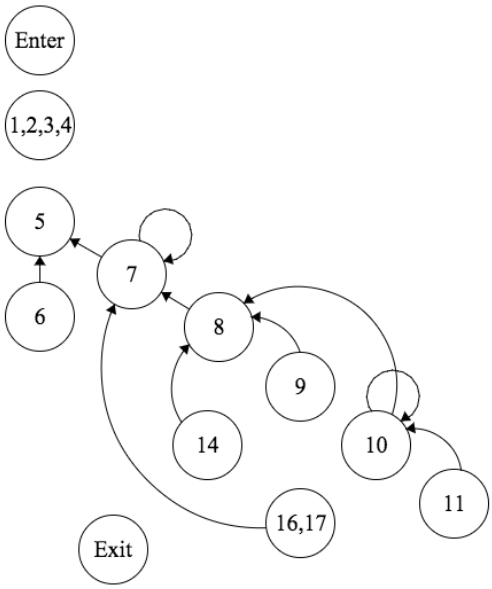
1. $s \to a \to ...b \to ... \to Exit$
2. $s \to b \to ...a \to ... \to Exit$

Then there must be a path $a \to ... \to Exit$. Which implies there is a path $s \to a \to ... \to Exit$. Which implies b is not IPDOM, which is a contradiciton. Without loss of generality this proof shows there cannot be $n > 1$ IPOM. What if the path to Exit is length 1? I.e. $s \to Exit$. Then the above proof doesn't work. In that case Exit is the IPDOM.

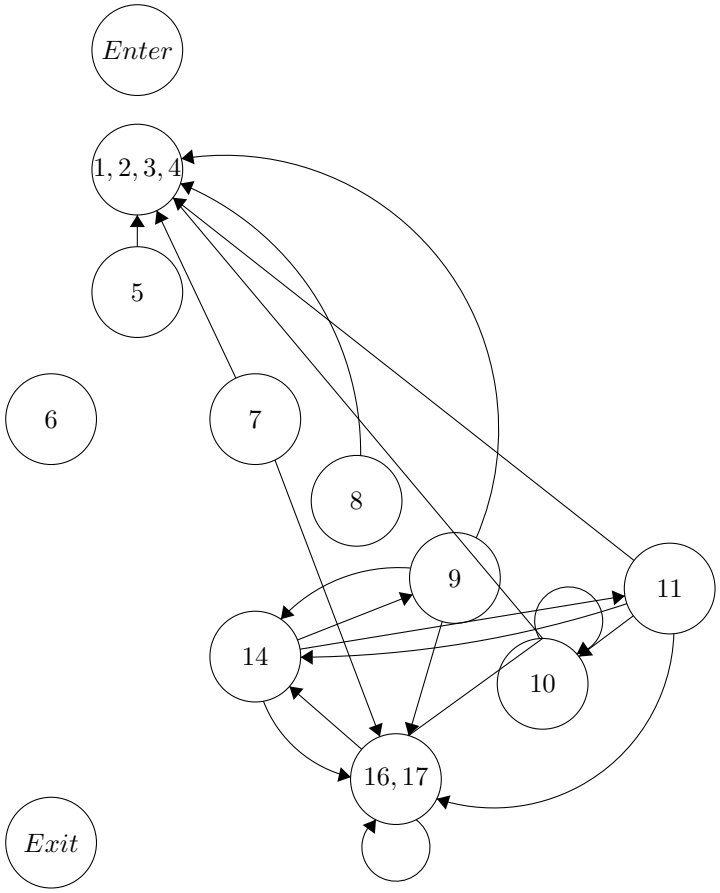
2 Question 2

Program dependence = control dependence + data dependence

2.1 Control dependence



2.2 Data dependence



3 Question 3

3.1 1

There are 5 unique path:

$Entry- > 1, 2, 3, 4- > 5- > 6- > Exit$

$Entry- > 1, 2, 3, 4- > 5- > 7- > Exit$

$Entry- > 1, 2, 3, 4- > 5- > 7- > 8- > 14- > 16, 17- > 7- > Exit$

$Entry- > 1, 2, 3, 4- > 5- > 7- > 8- > 9- > 10- > 16, 17- > 7- > Exit$

$Entry- > 1, 2, 3, 4- > 5- > 7- > 8- > 9- > 10- > 11- > 10- > 16, 17- > 7- > Exit$

3.2 2

There are 6 unique paths:

ACDF

ACDEF

ABCDF

ABCDEF
ABDF
ABDEF

4 Question 4

Let plain text string be : a b c d a a b c d e a a a b c d.

The initial lookup table is:

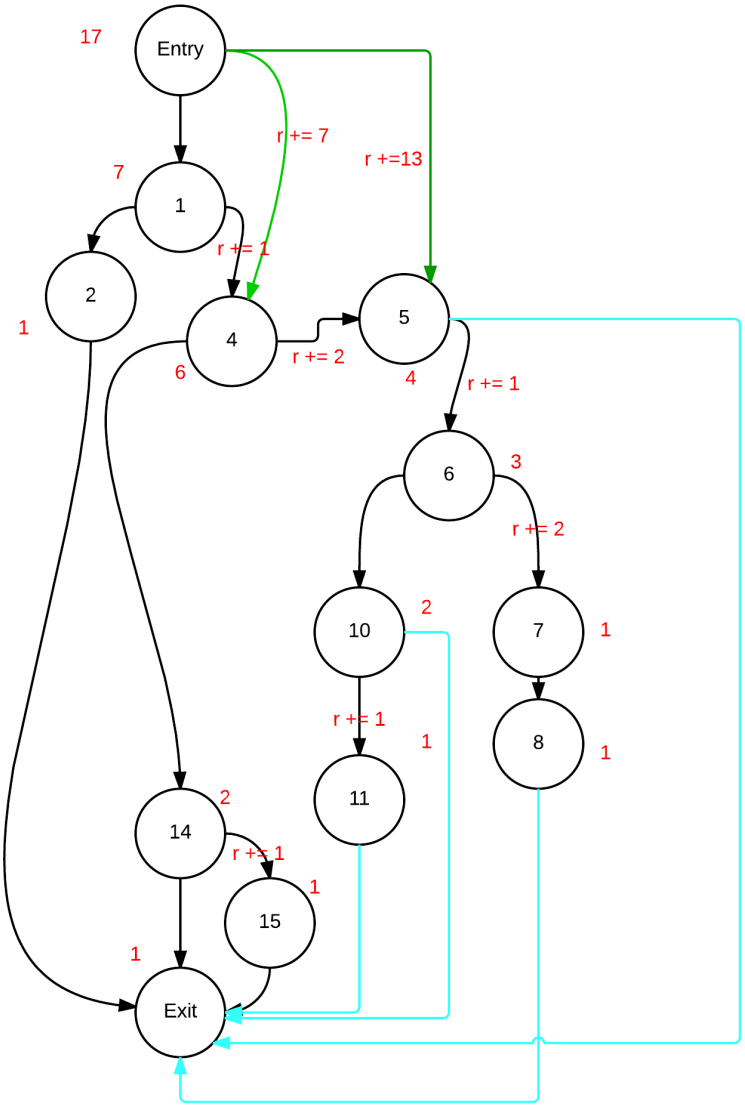
aa	a
bd	a
bc	d

Final compressed string:
m a m b m c 0 m a m a m b 0 0 m e m a m a m a m b 0 0

Final lookup table is:

aa	b
bd	a
bc	d
cd	e
da	a
ab	c
de	a
ea	a

5 Question 5


17
Entry
r += 7
r +=13
7
1
r += 1
2
1
4
5
6
r += 2
4
r += 1
6
3
r += 2
10
2
7
1
r += 1
1
8
1
11
2
14
r += 1
1
15
1
Exit

Path Encoding:

Enter, 1, 4, 14, End	0
Enter, 1, 4, 14, 15, End	1
Enter, 1, 4, 5, End	2
Enter, 1, 4, 5, 6, 10, End	3
Enter, 1, 4, 5, 6, 10, 11, End	4
Enter, 1, 4, 5, 6, 7, 8, End	5
Enter, 1, 2, End	6
Enter, 4, 14, End	7
Enter, 4, 14, 15, End	8
Enter, 4, 5, End	9
Enter, 4, 5, 6, 10, End	10
Enter, 4, 5, 6, 10, 11, End	11
Enter, 4, 5, 6, 7, 8, End	12
Enter, 5, End	13
Enter, 5, 6, 10, End	14
Enter, 5, 6, 10, 11, End	15
Enter, 5, 6, 7, 8, End	16

Instrumentation:

```
r = 0;
if (p1) {
        r+=6;
        s1;
} else {
        while (p2) {
        r+=2;
        while (p3) {
                r+=1;
                if (p4) {
                r+=2;
                s2;
                counter[r]++;
                r = 13;
                continue;
            }
            if (p5) {
                r+=1;
                break;
                counter[r]++;
                r = 7;
            }
            counter[r]++;
            r=13;
        }
        counter[r]++;
        r=7;
    }
    if (p6){
```

```
            r+=1;
            s3;
        }
        counter[r]++;
}
```

6 Question 6

Solution: Assume C language.

1. Multiple function call sites. The trace does not directly tell you where to return. The solution includes recording the program point that is returned to. Or the decoding algorithm needs to maintain a call stack.
2. When a method is called through a function pointer, the target needs to be traced.
3. For switch-case statements, the algorithm needs to trace which case is taken.
4. For long jumps and set jumps, the correspondence needs to be traced.

Assume Java language.

1. Due to class inheritance, the dynamic method that is being called needs to be traced.
2. The places that an exception is thrown and handled need to be traced.
3. The above (1).
4. The above (3).

Note: students didn't have to give all of these and are expected to give fuller explanations.