100

# CS458 Dynamic Analysis of Software Source Code: Homework #1

Due on September 22, 2022 at 23:59

*Professor Moonzoo Kim*

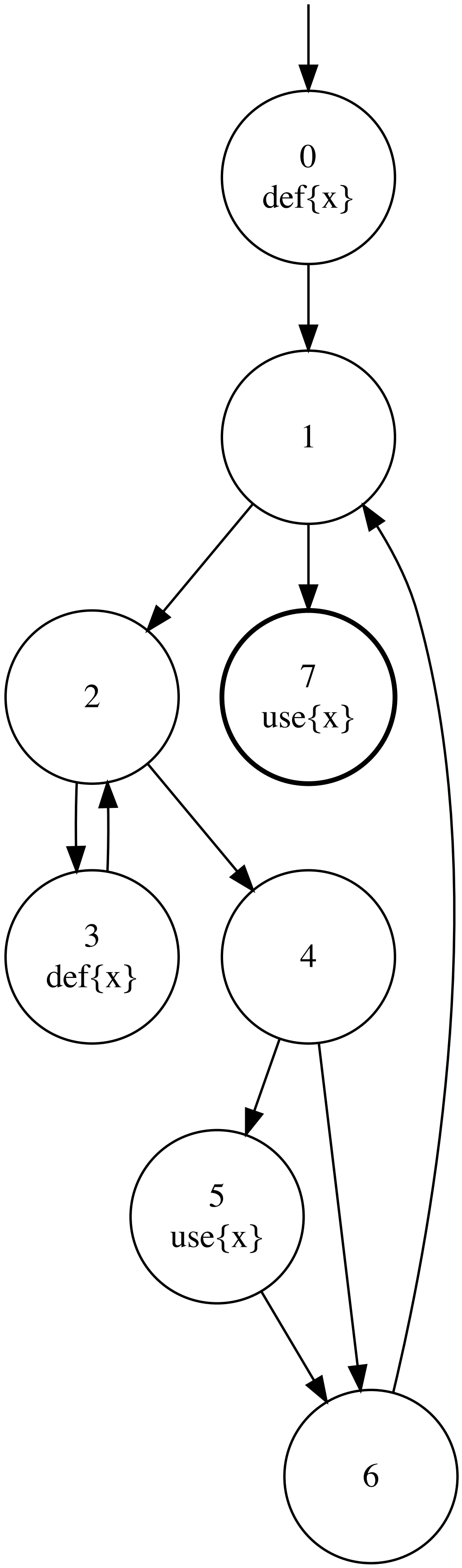
**20170231 SeungGi Min**

# 1. (a)

Draw the graph using DOT language.

```
digraph figure {
    rankdir=TB;
    null [shape=point width=0];
    node [shape=circle width=1];
    0 1 2 3 4 5 6;
    7 [shape=circle width=1 style=bold];

    null -> 0;
    0 -> 1;
    1 -> 2, 7;
    2 -> 3, 4;
    3 -> 2;
    4 -> 5, 6;
    5 -> 6;
    6 -> 1;
}
```


0
def{x}
1
2
7
use{x}
3
def{x}
4
5
use{x}
6

# 1. (b)

List all of the du-paths with respect to x. (Note: Include all du-paths, even those that are subpaths of some other du-path).

du1 = $[0, 1, 2, 4, 5]$
du2 = $[0, 1, 7]$
du3 = $[3, 2, 4, 5]$
du4 = $[3, 2, 4, 5, 6, 1, 7]$
du5 = $[3, 2, 4, 6, 1, 7]$

# 1. (c)

For each test path, determine which du-paths that test path du-tours (i.e., satisfying the def-clear requirement). For this part of the exercise, you should consider both direct touring and sidetrips. Hint: A table is a convenient format for describing this relationship.

| | du1 | du2 | du3 | du4 | du5 |
|---|---|---|---|---|---|
| t1 | | ⊙ | | | |
| t2 | | ◯ | | | |
| t3 | ⊙ | ◯ | | | |
| t4 | | ◯ | | | ⊙ |
| t5 | ◯ | ◯ | ⊙ | ⊙ | |
| t6 | ◯ | ◯ | ◯ | ◯ | ◯ |

⊙ means that the test path tours the du-path directly, and ◯ means that the test path tours the du-path with sidetrips.

# 1. (d)

List a minimal test set that satisfies all defs coverage with respect to x. (Direct tours only.) Use the given test paths.

$\text{du}(0, x) = \{\text{du1}, \text{du2}\}$
$\text{du}(3, x) = \{\text{du3}, \text{du4}, \text{du5}\}$

Since t1 tours du2 ∈ du(0, x) and t4 tours du5 ∈ du(3, x) directly, {t1, t4} is a minimal test set that satisfies all defs coverage using the given test paths.
*Note:* {[0, 1, 2, 4, 5, 6, 1, 2, 3, 2, 4, 5, 6, 1, 7]} is a minimal test set that satisfies all defs coverage, without using the given test paths.

# 1. (e)

List a minimal test set that satisfies all uses coverage with respect to x. (Direct tours only.) Use the given test paths.

$\text{du}(0, 5, x) = \{\text{du1}\}$

$\text{du}(0, 7, x) = \{\text{du2}\}$

$\text{du}(3, 5, x) = \{\text{du3}\}$

$\text{du}(3, 7, x) = \{\text{du4}, \text{du5}\}$

t3 tours du1 ∈ du(0, 5, x), t1 tours du2 ∈ du(0, 7, x), t5 tours du3 ∈ du(3, 5, x) and du4 ∈ du(3, 7, x) directly. Hence {t1, t3, t5} is a minimal test set that satisfies all uses coverage using the given test paths.
*Note:* {[0, 1, 7], [0, 1, 2, 4, 5, 6, 1, 2, 3, 2, 4, 5, 6, 1, 7]} is a minimal test set that satisfies all uses coverage, without using the given test paths.

# 1. (f)

List a minimal test set that satisfies all du-paths coverage with respect to x. (Direct tours only.) Use the given test paths.

t3 tours du1, t1 tours du2, t5 tours du3 and du4, t4 tours du5 directly. Hence {t1, t3, t4, t5} is a minimal test set that satisfies all du-paths coverage using the given test paths.
*Note:* {[0, 1, 7], [0, 1, 2, 3, 2, 4, 6, 1, 7], [0, 1, 2, 4, 5, 6, 1, 2, 3, 2, 4, 5, 6, 1, 7]} is a minimal test set that satisfies all du-paths coverage, without using the given test paths.

# 2. (a)

Consider test cases t1:(n = 3) and t2:(n = 5). Although these tour the same prime paths in printPrimes(), they do not necessarily find the same faults. Design a simple fault that t2 would be more likely to discover than t1 would (note that the fault should not change the control flow graph).

Let `MAXPRIMES` be 3. Then t1:(n = 3) would not find the fault, while t2:(n = 5) would find the out of bound access of `int [] primes`.

# 2. (b)

For printPrimes(), find a test case such that the corresponding test path visits the edge that connects the beginning of the `while` statement to the second `for` statement without going through the body of the while loop.

Let `n` be 1. Then the test path satisfies that `numPrimes >= n` without going through the body of the `while` loop, and satisfies that `0 <= numPrimes - 1` which visits the body of the second `for` loop.

# 2. (c)

Enumerate the test requirements for Node Coverage, Edge Coverage, and Prime Path Coverage for the graph for printPrimes(). Please write down the test requirements for prime path in an increasing order of a size of test requirements.

1. Node Coverage

   TR = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}

2. Edge Coverage

   TR = {(0, 1), (1, 2), (1, 10), (2, 3), (3, 4), (3, 7), (4, 5), (4, 6), (5, 7), (6, 3), (7, 8), (7, 9), (8, 9), (9, 1), (10, 11), (11, 12), (11, 14), (12, 13), (13, 11)}

3. Prime Path Coverage

   TR = {

   $[3, 4, 6, 3]$,
   $[4, 6, 3, 4]$,
   $[6, 3, 4, 6]$,
   $[11, 12, 13, 11]$,
   $[12, 13, 11, 12]$,
   $[12, 13, 11, 14]$,
   $[13, 11, 12, 13]$,

   $[0, 1, 10, 11, 14]$,

   $[0, 1, 2, 3, 4, 6]$,
   $[0, 1, 2, 3, 7, 9]$,
   $[0, 1, 10, 11, 12, 13]$,
   $[1, 2, 3, 7, 9, 1]$,
   $[2, 3, 7, 9, 1, 2]$,
   $[3, 7, 9, 1, 2, 3]$,
   $[7, 9, 1, 2, 3, 7]$,
   $[9, 1, 2, 3, 7, 9]$,

   $[0, 1, 2, 3, 7, 8, 9]$,
   $[1, 2, 3, 7, 8, 9, 1]$,
   $[2, 3, 7, 8, 9, 1, 2]$,
   $[3, 7, 8, 9, 1, 2, 3]$,
   $[4, 6, 3, 7, 9, 1, 2]$,
   $[7, 8, 9, 1, 2, 3, 7]$,
   $[8, 9, 1, 2, 3, 7, 8]$,
   $[9, 1, 2, 3, 7, 8, 9]$,

[0, 1, 2, 3, 4, 5, 7, 9],
[1, 2, 3, 4, 5, 7, 9, 1],
[2, 3, 4, 5, 7, 9, 1, 2],
[2, 3, 7, 9, 1, 10, 11, 14],
[3, 4, 5, 7, 9, 1, 2, 3],
[4, 5, 7, 9, 1, 2, 3, 4],
[4, 6, 3, 7, 8, 9, 1, 2],
[5, 7, 9, 1, 2, 3, 4, 5],
[5, 7, 9, 1, 2, 3, 4, 6],
[6, 3, 4, 5, 7, 9, 1, 2],
[7, 9, 1, 2, 3, 4, 5, 7],
[9, 1, 2, 3, 4, 5, 7, 9],

[0, 1, 2, 3, 4, 5, 7, 8, 9],
[1, 2, 3, 4, 5, 7, 8, 9, 1],
[2, 3, 4, 5, 7, 8, 9, 1, 2],
[2, 3, 7, 8, 9, 1, 10, 11, 14],
[2, 3, 7, 9, 1, 10, 11, 12, 13],
[3, 4, 5, 7, 8, 9, 1, 2, 3],
[4, 5, 7, 8, 9, 1, 2, 3, 4],
[5, 7, 8, 9, 1, 2, 3, 4, 5],
[5, 7, 8, 9, 1, 2, 3, 4, 6],
[6, 3, 4, 5, 7, 8, 9, 1, 2],
[4, 6, 3, 7, 9, 1, 10, 11, 14],
[7, 8, 9, 1, 2, 3, 4, 5, 7],
[8, 9, 1, 2, 3, 4, 5, 7, 8],
[9, 1, 2, 3, 4, 5, 7, 8, 9],

[2, 3, 4, 5, 7, 9, 1, 10, 11, 14],
[2, 3, 7, 8, 9, 1, 10, 11, 12, 13],
[4, 6, 3, 7, 8, 9, 1, 10, 11, 14],
[4, 6, 3, 7, 9, 1, 10, 11, 12, 13],
[6, 3, 4, 5, 7, 9, 1, 10, 11, 14],

[2, 3, 4, 5, 7, 8, 9, 1, 10, 11, 14],
[2, 3, 4, 5, 7, 9, 1, 10, 11, 12, 13],
[4, 6, 3, 7, 8, 9, 1, 10, 11, 12, 13],
[6, 3, 4, 5, 7, 8, 9, 1, 10, 11, 14],
[6, 3, 4, 5, 7, 9, 1, 10, 11, 12, 13],

$[2, 3, 4, 5, 7, 8, 9, 1, 10, 11, 12, 13]$,
$[6, 3, 4, 5, 7, 8, 9, 1, 10, 11, 12, 13]$

}

## 2. (d)

List a set of test paths that achieve Node Coverage but not Edge Coverage on the graph.

Consider the set of test paths is { [0, 1, 2, 3, 4, 6, 3, 4, 5, 7, 8, 9, 1, 10, 11, 12, 13, 11, 14] }. This achieves Node Coverage, but not Edge Coverage because of the unreached edges (3, 7) and (7, 9).
*Note:* This set of test paths cannot be executed by *any* tests. To visit node 5, `n` should be greater than 2 and a singleton set of test such that satisfies `n > 2` also achieves Edge Coverage.

## 2. (e)

List a set of test paths that achieve Edge Coverage but not Prime Path Coverage on the graph.

The set of test paths is { [0, 1, 2, 3, 4, 6, 3, 7, 8, 9, 1, 2, 3, 4, 5, 7, 9, 1, 2, 3, 4, 6, 3, 4, 6, 3, 7, 8, 9, 1, 10, 11, 12, 13, 11, 12, 13, 11, 12, 13, 11, 14] }. This achieves Edge Coverage, but not Prime Path Coverage because it doesn't contain some prime paths such that [0, 1, 10, 11, 14], [0, 1, 2, 3, 7, 9], etc.
*Note:* The given test path can be executed by t1:(n = 3).

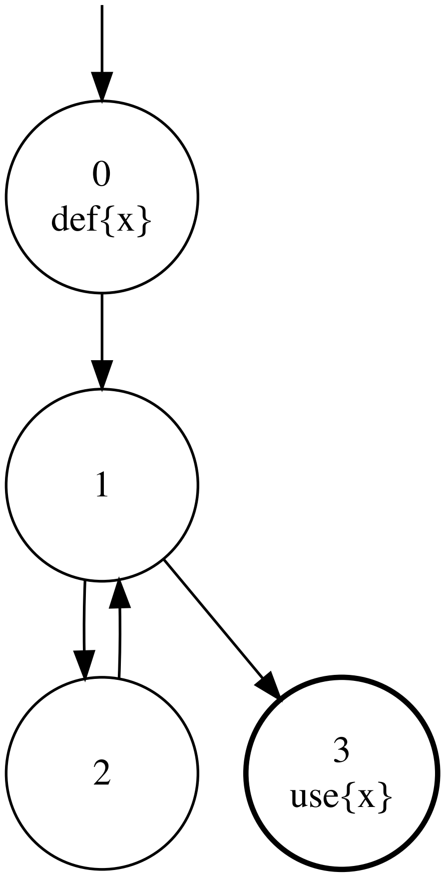
## 3. (a)

Explain why prime-path coverage subsumes All-DU-paths coverage.

Assume that there exists a set of test paths such that satisfies prime path coverage, but not all-du-paths coverage. Then there exists a du-path $p$ such that every prime path doesn't tour $p$. This is a contradiction because du-path $p$ is a simple path by the definition of du-path, and there should be at least one prime path that tours $p$ directly since $p$ is a simple path.
By contradiction, there is no such set of test paths that satisfies prime path coverage but not all-du-paths coverage, which implies that prime path coverage subsumes all-du-paths coverage.

## 3. (b)

Make an example graph G and a set of test paths T that satisfy All-DU-paths coverage.




Let the set of test paths T = { [0, 1, 3] }. The test requirement of all du-paths coverage contains all du-paths and there is only one du-path, [0, 1, 3]. Hence T satisfies all-du-paths coverage.

## 3. (c)

Show that T does not satisfy prime-path coverage on G.

The test requirement of prime path coverage contains all prime paths, {[0, 1, 3], [1, 2, 1], [2, 1, 2], [2, 1, 3]}. Since T doesn't contain prime paths except [0, 1, 3], T doesn't satisfy prime path coverage.