Assignment 1: Graph Coverage

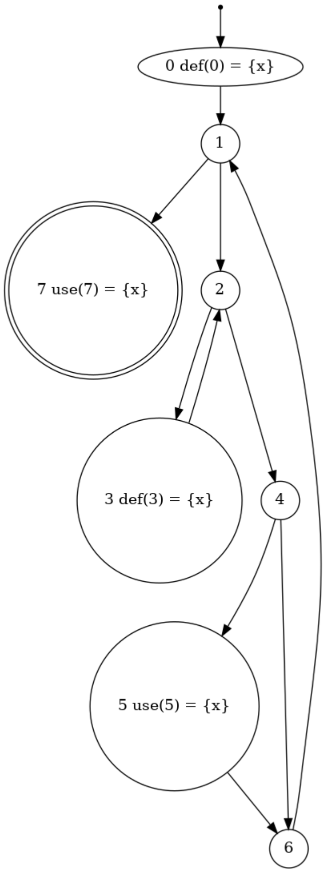
CS458 Dynamic Analysis of Software Source Code

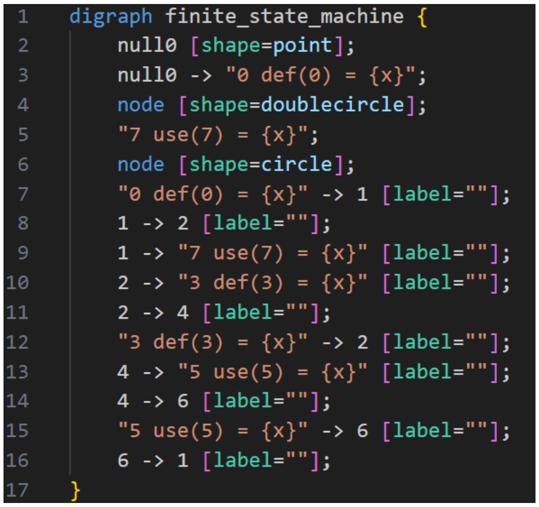
20234252, Yang Heechan

March 21, 2024

1 Problem #1 (40 points) Answer the following questions about graph I

1.1 (a) Draw the graph using DOT language





```
digraph finite_state_machine {
    null0 [shape=point];
    null0 -> "0 def(0) = {x}";
    node [shape=doublecircle];
    "7 use(7) = {x}";
    node [shape=circle];
    "0 def(0) = {x}" -> 1 [label=""];
    1 -> 2 [label=""];
    1 -> "7 use(7) = {x}" [label=""];
    2 -> "3 def(3) = {x}" [label=""];
    2 -> 4 [label=""];
    "3 def(3) = {x}" -> 2 [label=""];
    4 -> "5 use(5) = {x}" [label=""];
    4 -> 6 [label=""];
    "5 use(5) = {x}" -> 6 [label=""];
    6 -> 1 [label=""];
}
```

1.2 (b) List all of the du-paths with respect to x

- $b1$ = [0, 1, 7]

- $b2$ = [0, 1, 2, 4, 5]

- $b3$ = [3, 2, 4, 5]

- $b4$ = [3, 2, 4, 6, 1, 7]

- $b5$ = [3, 2, 4, 5, 6, 1, 7]

1.3 (c) For each test path, determine which du-paths that test path du-tours.

	$b1$	$b2$	$b3$	$b4$	$b5$
$t1$	O	X	X	X	X
$t2$	O	X	X	X	X
$t3$	O	O	X	X	X
$t4$	O	X	X	O	X
$t5$	O	O	O	X	O
$t6$	O	O	O	O	O

1.4 (d) List a minimal test set that satisfies all defs coverage with respect to x. (Direct tours only)

- $t1$ = [0, 1, 7]

- $t5$ = [0, 1, 2, 3, 2, 3, 2, 4, 5, 6, 1, 7]

1.5 (e) List a minimal test set that satisfies all uses coverage with respect to x. (Direct tours only)

- $t1$ = [0, 1, 7]

- $t3$ = [0, 1, 2, 4, 5, 6, 1, 7]

- $t5$ = [0, 1, 2, 3, 2, 3, 2, 4, 5, 6, 1, 7]

1.6 (f) List a minimal test set that satisfies all du-paths coverage with respect to x. (Direct yours only)

- $t1$ = [0, 1, 7]

- $t3$ = [0, 1, 2, 4, 5, 6, 1, 7]

- $t4$ = [0, 1, 2, 3, 2, 4, 6, 1, 7]

- $t5$ = [0, 1, 2, 3, 2, 3, 2, 4, 5, 6, 1, 7]

2 Problem #2 (40 points) Use the following method printPrimes() which prints n small prime number with a given input n for question a-e below. Answer the question based on the given control flow graph.

2.1 (a) 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).

- Change code number that increments `curPrime` by 1 (`curPrime++;`) so that `curPrime` increments by `numPrimes` (`curPrime += numPrimes;`).

2.2 (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.

- test case: $(n = 1)$

2.3 (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.

- Node Coverage
  - $TR_{\text{NC}} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
- Edge Coverage
  - $TR_{\text{NC}} = \{(0, 1), (1, 2), (1, 10), (2, 3), (3, 4), (3, 7), (4, 5), (4, 6), (5, 7), (6, 3), (7, 8), (7, 9), (8, 9), (9, 1), (10, 11), (11, 12), (11, 14), (12, 13), (13, 11)\}$
- Prime Path Coverage
  - $TR_{\text{NC}} = \{[0, 1, 2], [0, 1, 10], [1, 2, 3], [1, 10, 11], [2, 3, 4], [2, 3, 7], [3, 4, 5], [3, 4, 6], [3, 7, 8], [3, 7, 9], [4, 5, 7], [4, 6, 3], [5, 7, 8], [5, 7, 9], [6, 3, 4], [6, 3, 7], [7, 8, 9], [7, 9, 1], [8, 9, 1], [9, 1, 2], [9, 1, 10], [10, 11, 12], [10, 11, 14], [11, 12, 13], [12, 13, 11], [13, 11, 12], [13, 11, 14]\}$

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

- $t1 = [0, 1, 2, 3, 4, 6, 3, 7, 8, 9, 1, 10, 11, 12, 13, 11, 14]$

- $t2 = [0, 1, 2, 3, 4, 6, 3, 4, 5, 7, 8, 9, 1, 10, 11, 12, 13, 11, 14]$ (is an infeasible.)

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

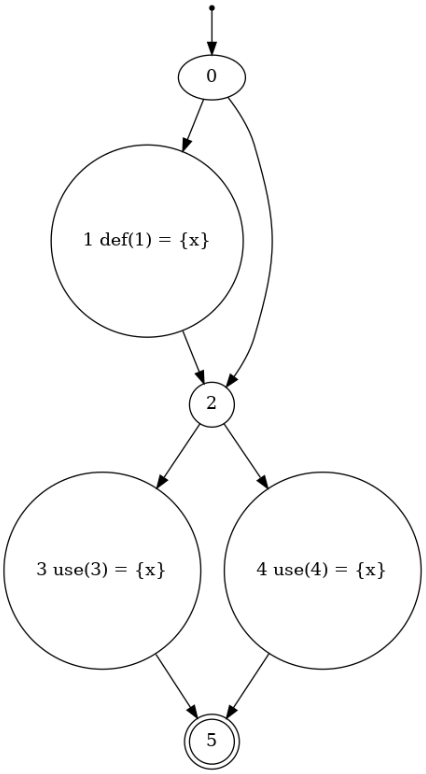
- $t1 = [0, 1, 2, 3, 4, 6, 3, 7, 8, 9, 1, 10, 11, 12, 13, 11, 14]$
- $t2 = [0, 1, 2, 3, 4, 6, 3, 7, 8, 9, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 11, 14]$

3 problem #3 (20 points) Show that prime-path coverage subsumes All-DU-paths coverage, but not vice versa.

3.1 (a) Explain why prime-path coverage subsumes All-DU-paths coverage.

- All-DU-path coverage requires all paths from a definition to its use to be covered.
- Prime-path coverage requires all prime paths, which are paths that no node appears more than once (except possibly the first and last node) and do not appear a subpath of any simple paths, to be covered.
- Therefore, prime paths cover all possible unique combinations of branches, which includes All-DU-paths.

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



- $t1 = [0, 1, 2, 3, 5]$
- $t2 = [0, 1, 2, 4, 5]$

3.3 (c) Show that T does not satify prime-path coverage on G

- T which is the test suite [$t1$, $t2$] does not satisfy prime-path coverage because this test suite does not cover a unique path [0, 2, 4, 5] which is a unique path from prime-path.