SE465 Midterm practice W2010

True/False

1. False. A du-path can contain any number > 1 of uses of the variable.

2. False. A prime path is a simple path of maximal length. Concatenating two prime paths results in a path that is no longer simple.

3. False. Prime path coverage can be satisfied with test paths that have the prime paths as subpaths.

4. True. For example, complete path coverage subsumes all-du-paths coverage.

$$\begin{tikzcd}
 & \mathrm{CPC} \arrow[dl] \arrow[d] \arrow[dr] & \\
\mathrm{ADUPC} \arrow[d] & \mathrm{PPC} \arrow[l] \arrow[r] \arrow[d] & \mathrm{EPC} \arrow[dl] \\
\mathrm{AUC} \arrow[r] \arrow[d] & \mathrm{EC} \arrow[d] & \\
\mathrm{ADC} & \mathrm{NC} &
\end{tikzcd}$$

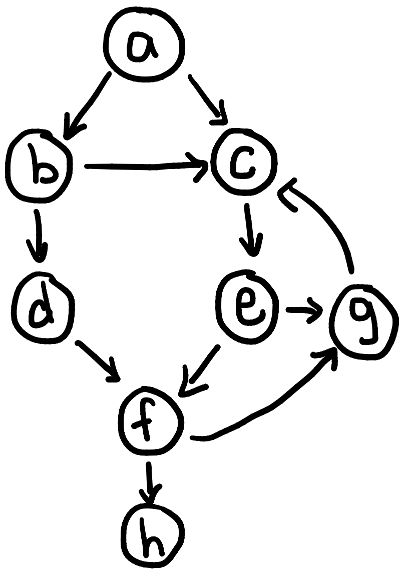
* If there are no self-loops

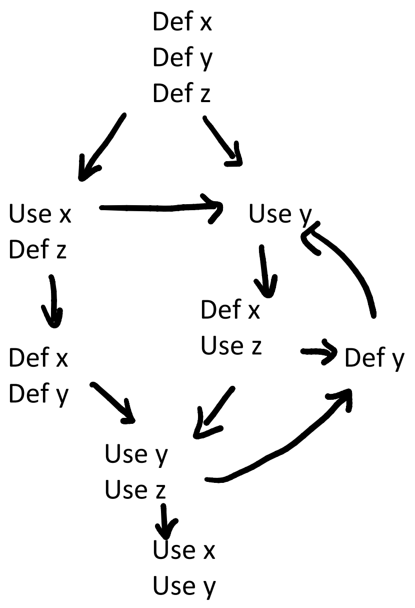
5. False ?

1. [Not covered]

2. [Not covered]

3.





Structural coverage

NC: a, b, c, d, e, f, g, h

EC: (a, b), (a, c), (b, c), (b, d), (c, e), (d, f), (e, f), (e, g), (f, g), (g, c), (f, h)

EPC: (a, b, c), (a, b, d), (a, c, e), (b, c, e), (b, d, f), (c, e, f), (c, e, g), (d, f, h), (d, f, g), (e, g, c), (e, f, g), (e, f, h), (f, g, c)

PPC:
(a, b, c, e, f, h), (a, b, c, e, f, g), (a, b, c, e, g),
(a, c, e, f, h), (a, c, e, f, g), (a, b, e, g),
(a, b, d, f, h), (a, b, d, f, g, c, e),
(c, e, g), (e, g, c), (g, c, e),
(c, e, f, g), (e, f, g, c), (f, g, c, e), (g, c, e, f),
(e, f, g), (f, g, e), (g, e, f)

Data flow coverage

x is defined in nodes a, d, e.
x is used in nodes b, h.
y is defined in nodes a, d, g.
y is used in nodes c, f, h.
z is defined in nodes a, b.
z is used in nodes e, f.

ADC:
  x: (a, b), (d, f, h), (e, f, h),
  y: (a, c), (d, f), (g, c)
  z: (a, c, e), (b, d, f)

AUC:
  x: (a, b), (d, f, h), (e, f, h)
  y: (a, c) or (a, b, c),
    (a, c, e, f) or (a, b, c, e, f),
    (a, c, e, f, h) or (a, b, c, e, f, h),
    (d, f),
    (d, f, h),
    (g, c),
    (g, c, e, f),
    (g, c, e, f, h)
  z: (a, c, e),
    (a, c, e, f),
    (b, c, e) or (b, d, f, g, c, e),
    (b, d, f) or (b, c, e, f)

ADUPC:
  x: (a, b), (d, f, h), (e, f, h)
  y: (a, c), (a, b, c), (a, c, e, f), (a, b, c, e, f), (a, c, e, f, h), (a, b, c, e, f, h), (d, f),
    (d, f, h), (g, c), (g, c, e, f), (g, c, d, f, h)
  z: (a, c, e), (a, c, e, f), (b, c, e), (b, d, f, g, c, e), (b, d, f), (b, c, e, f)

4.

```
// If x < 10, return x * y. Otherwise return x + y
function doMath(int x, int y) {
        if (x < 10) {
                return x * x
        }
        return x + y
}
```

$$\begin{tikzcd} & \text{if } x < 10 \arrow[dl, "\textit{true}"'] \arrow[dr, "\textit{false}"] & \\ \text{return } x * x & & \text{return } x + y \end{tikzcd}$$

The fault is on the line *return x * x* - by our specification it should be *return x * y*.

Test suite satisfying CPC:
Test 1: x = 5, y = 5
Test 2: x = 100, y = 100

Our test suite satisfies complete path coverage: we traverse all 2 possible paths. However, we do not find the fault, since it returns the correct solution for x=5, y=5, since x=y.