SE465 Midterm W2016

1. A) The fault is on line 2: it should be i **<** list_length;

B) A test case that does not execute the fault does not exist. Line 2 of the program runs regardless of the input.

C) A test case that executes the fault and does not result in a failure is
list = [1]
num_to_find = 1
list_length = 1
The program iterates the first item in the list, sees that it is the num_to_find, and returns 1, which is the expected output.

2. A) MUST belief:

```
int* x = NULL;    // MUST belief: x is null
int y = *x/5;     // MUST belief: x is not null
```

We clearly have a bug here, since we have two contradicting MUST beliefs.

MAY belief:

```
A();              // MAY belief: A() must be followed by B()
B();
…
A();
B();
…
A();
B();
…
A();
```

A();..B(); implies the MAY belief that A() must be followed by B(). In our code, we have one instance where A() is not followed by B(), which could be a bug.

B) Valgrind. It is a dynamic tool, while all the others are static tools.

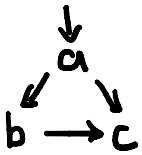
C) These functions write to a buffer without checking the size of the buffer, which could lead to buffer overflow.

D) A benign data race is one that does not affect the outcome of the program. In this program, if the program throws an exception in the while loop and enters the catch block, the lock is never released. This can be fixed by adding an unlock statement in the catch block.

3. Mutant 1: Apply a relational operator replacement to line 7: change the != to ==. A test case that kills the mutant is a=4, b=2. The expected output is 2. The mutant output is 4.

Mutant 2: Apply an arithmetic operator replacement to line 9: change the - to a /. A test case that kills the mutant is a=9, b=6. The expected output is 3. The mutant output is 1.

4. A) BC does not subsume STPC. Counterexample:

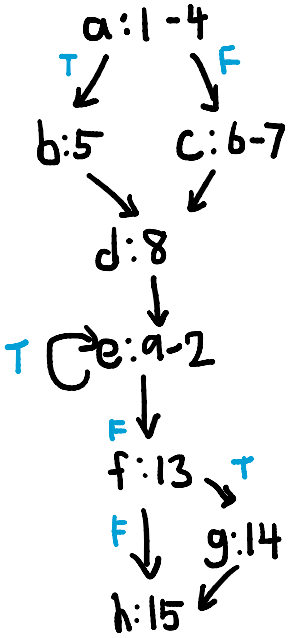


BC: ab

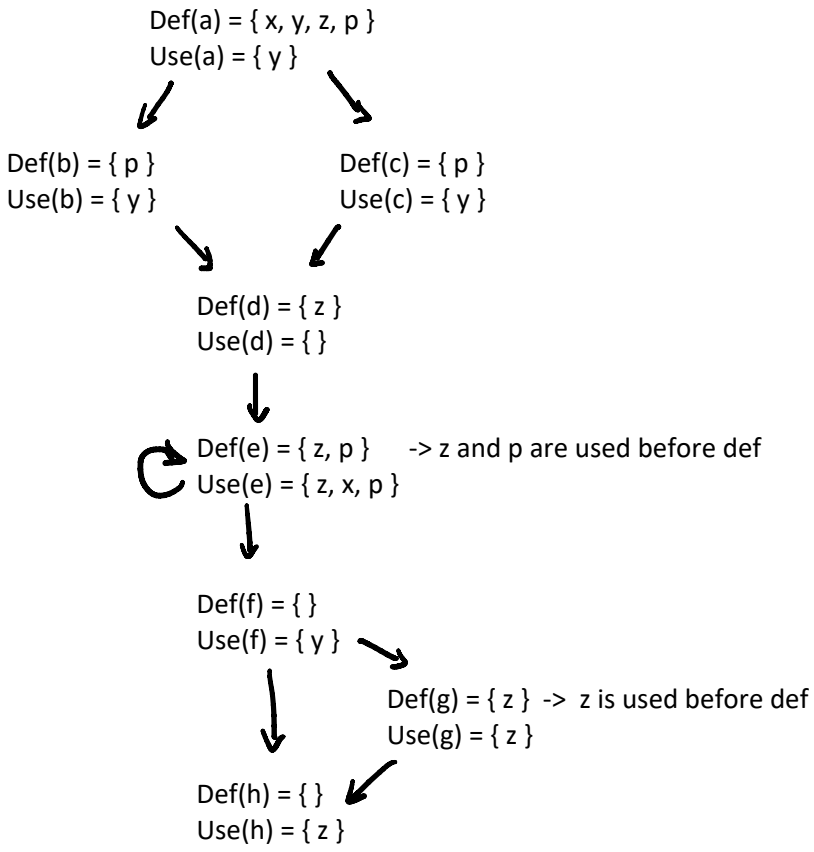
STPC: a, b, c, ab, ac, bc, abc

B) PPC subsumes ADUPC. Since a set of test cases satisfying PPC will cover every simple path, they will cover every du-path (since a du-path is a simple path). Hence every set of test cases satisfying PPC will also satisfy ADUPC.

5. a) [can't figure out how to make it 8 nodes]



b)



c) Infeasible: abdefh, acdefgh
- both ab and fg
- Both c and g

abdefgh, acdefh, ee

d) z is defined in nodes d, e, g.
z is used in nodes e, g, h.
du(d, e, z): de

du(d, g, z):
du(d, h, z):
du(e, e, z): ee
du(e, g, z): efg
du(e, h, z): efh
du(g, e, z):
du(g, g, z):
du(g, h, z): gh

AUC: de, ee, efg, efh, gh

e) The fault is that the do-while loop. To fix the fault, we just change the do-while loop to a while loop. A test case that executes the fault is x=2, y=0.