17-355/17-655: Program Analysis
Mini-Assignment 1
Due 11:59pm on April 7, 2017

1. Write a 1-2 paragraph reaction to Nadia Polikarpova’s talk last week. Your reaction should discuss your impressions of the benefits, drawbacks, and applicability of one or more of the techniques she described. You may also ask a question about the technique if you have one (I’ll try to answer it, if I know the answer!)

If you had a conflict and were unable to attend Nadia’s talk, you can find a video (accessible to any CMU student) at:

https://www.cs.cmu.edu/~scsdean/Hiring/Manual_capture_for_GHC_6117_CATTURA_1_at_Mar_29,_2017_13_59_17_(Source).mp4

2. Write a 1-2 paragraph reaction to Michael Bond’s guest lecture this last week. Your reaction should discuss your impressions of the benefits, drawbacks, and applicability of the dynamic analysis he described. You may also ask a question about the technique if you have one (which again, I’ll try to answer if I can).

3. If you did not complete the SMT lab in recitation on Friday, March 31st, finish any parts that were left over and turn them in. The lab exercises are attached below. If you completed the lab in recitation section, you do not need to do anything for this question.

Put your answers to questions 1 and 2 into a text file turn it in on Blackboard. If you didn’t already show Prof. Aldrich your solutions to the lab last Friday, also include the Z3 formulas and (for reachable errors) an appropriate test case input for questions A and B below.
A. Can this error happen? Using ideas from symbolic execution, write a SMT formula that will be true if and only if the error is reachable for some input. Evaluate your formula on riseforfun.com/z3 and report the results. If the error can happen, get a model from z3 and use that to compute an input that can trigger the error.

Note: you (and the SMT solver) may assume that the function f (definition not shown) represents a mathematical function, and in particular executes deterministically.

```c
void bar(int x) {
    int y1 = x + 1;
    int y2 = y1 * 2;
    int y3 = y2 - 2;
    int y4 = y3 - x;
    if (x == f(y4)) {
        if (y4 != f(f(x)))
            ERROR
    }
}
```

B. Do the same for this function. For this example, you need not check the bounds of the array access (although a real concolic testing tool probably would).

```c
void foo(int i, int j, int x, int a[]) {
    int y = a[i];
    if (x != y)
        a[j] = x;
    if (a[i] != y)
        ERROR
}
```