Static Analysis for Software Engineering (ECE750-T05): Final*

July 31, 2013

This final exam has 5 questions of equal weight. Please do not collaborate; if you have a question, you may contact me (only) and I will get back to you. You may talk to other people about which questions you have completed and how long it took you to do each question.

Question 1:

This question is about the Dillig, Dillig, Aiken and Sagiv paper "Precise and Compact Modular Procedure Summaries for Heap Manipulating Programs". The three parts of this question have equal weight.

(1a) Symbolic Heap. Produce a symbolic heap for function f in Example 6.

(1b, 1c) Precondition, summary. Consider the following code:

```
define g(a1 : ptr(ptr(int)), a2 : ptr(ptr(int))) =
  let t1 : ptr(int) in t1 <- *a1; assert (*t1 == 2) end;
  choose (a2 <- a1, *a2 <- alloc(ptr(int)));
```

For part (1b), produce a precondition for function g, and for part (1c), produce a summary for function g, following the algorithm in the paper.

Question 2:

This question is about the Pradel/Gross paper "Automatic Testing of Sequential and Concurrent Substitutability". Consider the following two classes:

```
1 class A {
2   public A() { }
3   public int size() { return 0; }
4   public void add(T x) {
5     throw new UnsupportedOperationException("read-only!");
6   }
7 }
```

*version 0, July 31

```
 8
 9 class B extends A {
10   private T[] data; private int size;
11   public void B(int n) { data = new T[n]; size = 0; }
12   public int size() { return size; }
13   public void add(T x) { data[size++] = x; }
14 }
```

Show me how the techniques described in the paper would work on classes `A` and `B`. I'm looking for a set of calls, tests, and how an oracle identifies the difference between the classes. Be explicit.

Question 3:

Pick one of the papers we talked about in class (not one you presented) and write a critique of the paper. Start by summarizing the contributions of the paper. Identify two strengths and two weaknesses of the paper, providing reasons for your judgments. Finish with a one-paragraph summary of your discussion, which recommends that the paper either be accepted or rejected from a top-tier conference. (This exercise is basically asking you to write a review of the paper.)

Question 4:

Apply invariant generation, as in the Nguyen, Kapur, Weimer, and Forrest paper ("Using Dynamic Analysis to Discover Polynomial and Array Invariants"), to the following gcd code:

```
int gcd(int a, int b) {
  int c;
      while(b > 0)
      {
              c = a % b;
              a = b;
              b = c;
      }
      return a;
}
```

Figure out where there are interesting invariants to exhibit, produce them, and hand them in. As with the other questions, you may use resources already on the Internet, but don't ask for help from people.

Question 5:

In this question, you will formulate and implement a simple intraprocedural static analysis for the Clang framework[1] (`http://clang.llvm.org/`). Your task is to detect array accesses `a[i]` where, on some path, array index `i` has not been compared to a constant. For instance, this is OK:

```
if (i < 5) {
    printf("%d\n", a[i]);
}
```

while this should trigger a warning:

```
void foo(int * a, int i) {
    printf("%d\n", a[i]);
}
```

Hand in a textual description of your analysis as well as an implementation of a Clang pass which implements it.

[1] See `http://clang-analyzer.llvm.org/checker_dev_manual.html`, and the Clang `ArrayBoundsChecker`, for more information.