Assignment 4: Using a Software Checker

CS 6962 – Software Verification

November 15, 2012

Deadline: **Monday, Dec 3, 2012 at 11:59pm.**

The goal of this assignment is for you to gain a little bit of practical experience using a software checker to find bugs in programs. I intentionally left the assignment somewhat unstructured and open-ended — you will learn much more if exploring on your own instead of blindly following my prescribed steps.

Your Task. Your main task is to check a real-life application of your choice using a software checker of your choice. In the end, you will write a short report about the experience. I suggest you use Java Path Finder (JPF) as your checker [http://babelfish.arc.nasa.gov/trac/jpf]. JPF is a software model checker for concurrent Java programs. It is being actively developed and is easy to install on both Linux and MacOS. Alternatively, you can try KLEE, which is a concolic execution engine for C programs [http://klee.llvm.org]. I have a little bit less experience with KLEE than with JPF, but I believe it is also well-maintained and should be easy to install (on Linux). We have used it a lot within our research group and our experience is very positive. Installing these tools on Windows might be tricky, so I would advise against going down that route. Alternatively, you can use some other available software checker, just consult me quickly before you do that.

Note that the main power of JPF is to analyze concurrent programs, so your real-life application should ideally have some threading. On the other hand, KLEE primarily targets sequential C programs. You need to apply the checker on at least one real-life application (it can be your application), but you are free to do more than that if you want and have time. I would advise that you start with something small, like tutorials, examples that come with the tool, etc. And then gradually move to your chosen real-life application. A good place to find those is in experimental results sections of published tool papers. Try not to pick an application for which your checker doesn't work at all (this might require several iterations).

Assignment Deliverables. A short write-up (PDF format) about your experience using the software checker of your choice. Here are some of the questions you should try to answer in the write-up. How easy/hard was it to use the checker? On which real-life application(s) did you use it? How large is your real-life application (lines of code)? Is there anything particularly interesting about it? Did the checker report any bugs in the application? Are those real bugs or false positives? What about checker running times (make sure you report those)? Did you have any scalability issues? How did you work around those?

One page for this write-up is totally fine. Page limit is 3 pages. Email me your write-up as a PDF document. Do not email me source code, binaries, or anything like that.