Budapest University of Technology and Economics
Fault Tolerant Systems Research Group

Improving the Klee automatic test generator

Kubriczky Ádám, (BSc)
Consulent: dr. Micskei Zoltán, MIT
System Engineering Specialization
Project Laboratory Summary
2017/18. II.

During my project laboratory, I have dived into the internals of the Klee. The Klee is an automatic test generator for programs written in the C language. The main goal of it is to create tests that cover most of the codebase and to identify not logical errors (like buffer overflow). To achieve this, it uses symbolic execution, however it does not work on the source code, but on an intermediate language, the LLVM bytecode. Symbolic execution means it does not try to find concrete values on the fly, but it executes the IL with symbolic values and only requests an evaluation for concrete values with the gained constraints when a branch terminates. There are other tools besides Klee that generate tests automatically and work with C codes, but Klee is free and open-source so I could look under the hood easily.

The tool works pretty well (until we do not use floating point numbers in our program), but the process of reaching the tests from a source code requires some preparations and effort. During the semester I have identified several usability problems of the Klee and I managed to choose one from them which seemed to be well-sized for the project laboratory.

The problem I have worked on is to make the test outputs more readable. So imagoine, we have a little program with a few variables we would like to test it on. After all the work done to get the tests, we would like to see the exact values of the variables of each test outputs. However, Klee only supports readable ouput on 4 bytes width integers (and it needs an extra option) and every other types are represented as an array of bytes. This is very frustrating, especially when dealing with complex types like compound structures.

I have managed to work out a working (but limited) solution to this problem and made a pull request to the Klee's GitHub repository. Currently, I am cooperating with the developers to refine the solution to let it be able to merged into the master branch.