Symbolic State Debugging of C/LLVM-IR programs

TECHNISCHE UNIVERSITÄT DARMSTADT

Software Engineering Group

Master's Thesis in Software Engineering (FG Software Engineering, Prof. Dr. Reiner Hähnle)

Background

Symbolic execution is a versatile static program analysis technique. It is used for automatic test generation, fuzzing, debugging, deductive program verification and more.

In our group we developed the symbolic execution debugger (SED) which allows to analyse, inspect and visualize *all* feasible program execution paths and their intermediate states (see Figure 1).

The SED is realized as an Eclipse extension and hooks into the standard debugging framework provided by Eclipse. The SED uses the KeY program verifier for Java as its symbolic execution engine, but is designed in modular fashion allowing one to exchange the underlying symbolic execution engine.

This thesis is in the context of the LOEWE-Schwerpunkt "Software-Factory 4.0". The idea is to use the SED to perform regression analysis on C programs.

Approximate Work Distribution

Analysis
Programming
Literature

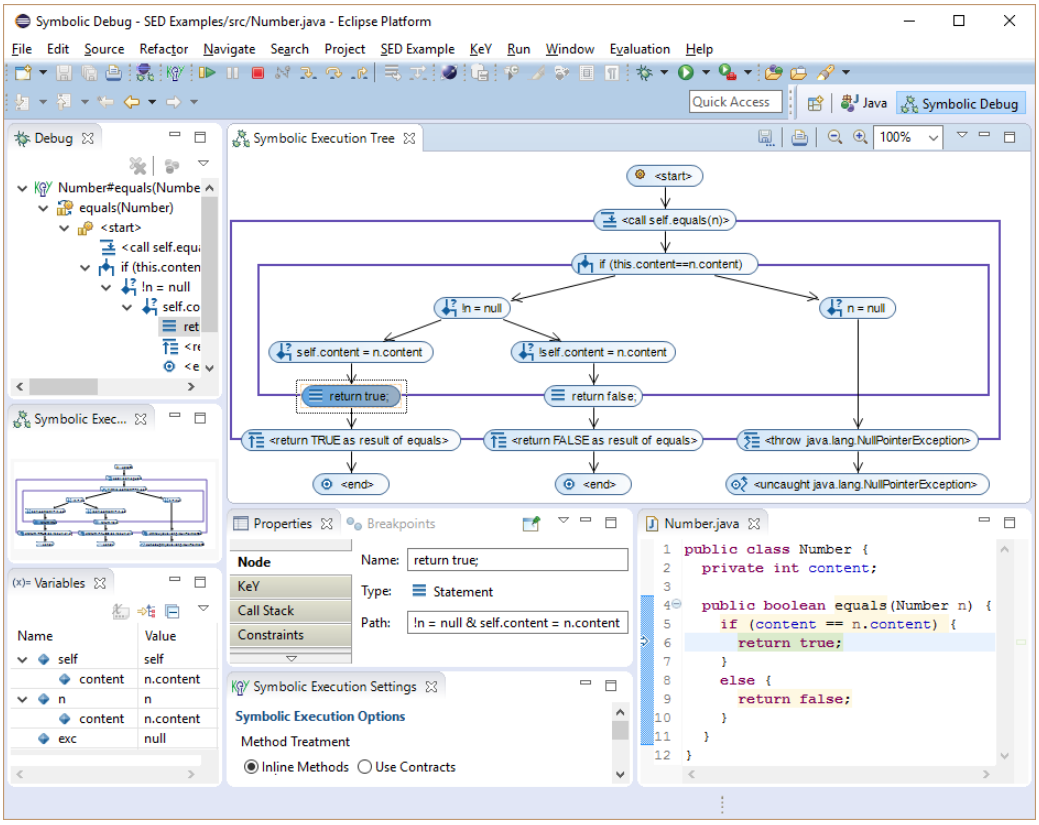

Figure 1: The Symbolic Execution Debugger

Task

The task is to adapt SED to be used for C programs using the KLEE symbolic interpreter instead of KeY. KLEE is a symbolic interpreter for LLVM-IR code which is an intermediate representation language used by LLVM compiler framework. C source code (and other) compiled by LLVM is first translated to LLVM-IR and then compiled to the target (machine) language.

In detail, the goals of this thesis consist of

- visualization and control of KLEE's symbolic execution using SED
- extraction and presentation of the symbolic states and the the symbolic program heap of the C program encountered during symbolic execution
- extraction of path conditions
- integration of SED in Eclipse CDT

Requirements

- Very good knowledge of Java, knowledge about Eclipse not required but of advantage
- Basic knowledge of C (or interest to learn C)
- Interest in formal methods and static program analysis

Literature

- M. Hentschel et al.: *The Symbolic Execution Debugger (SED): a platform for interactive symbolic execution, debugging, verification and more*, International Journal on Software Tools for Technology Transfer, Springer, 2018, available at: https://doi.org/10.1007/s10009-018-0490-9 (from within the TU Darmstadt network)
- *KLEE Homepage*, klee.github.io/getting-started/
- *KeY SED Homepage*, www.key-project.org/applications/debugging/
- *Software-Factory 4.0 Homepage*, www.software-factory-4-0.de

Contact

FG Software Engineering

- Nathan Wasser
  email: wasser@cs.tu-darmstadt.de
  room: A205 in S2|02
- Richard Bubel
  email: bubel@cs.tu-darmstadt.de
  room A225 in S2|02)