

Multi-solver Support in Symbolic Execution

Hristina Palikareva and Cristian Cadar

Department of Computing, Imperial College London
London, United Kingdom
{h.palikareva, c.cadar}@imperial.ac.uk

Abstract

In this talk, we will present the results reported in our CAV 2013 paper [6] on integrating support for multiple SMT solvers in the dynamic symbolic execution engine **KLEE** [2]. In particular, we will outline the key characteristics of the SMT queries generated during symbolic execution, introduce an extension of **KLEE** that uses a number of state-of-the-art SMT solvers (**Boolector** [1], **STP** [4] and **Z3** [3]) through the **metaSMT** [5] solver framework, and compare the solvers' performance when run on large sets of **QF_ABV** queries obtained during the symbolic execution of real-world software. In addition, we will discuss several options for designing a parallel portfolio solver for symbolic execution tools.

References

- [1] Robert Brummayer and Armin Biere. Boolector: An efficient SMT solver for bit-vectors and arrays. In *TACAS'09*.
- [2] Cristian Cadar, Daniel Dunbar, and Dawson Engler. KLEE: Unassisted and automatic generation of high-coverage tests for complex systems programs. In *OSDI'08*.
- [3] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *TACAS'08*.
- [4] Vijay Ganesh and David L. Dill. A decision procedure for bit-vectors and arrays. In *CAV'07*.
- [5] Finn Haedicke, Stefan Frehse, Görschwin Fey, Daniel Große, and Rolf Drechsler. metaSMT: Focus on your application not on solver integration. In *DFTS'12*.
- [6] Hristina Palikareva and Cristian Cadar. Multi-solver support in symbolic execution. In *CAV'13*. <http://srg.doc.ic.ac.uk/files/papers/klee-multisolver-cav-13.pdf>.