

PhD Course on Symbolic Execution, 2021 Lectures

- The roles of Symbolic Execution in Verification, Testing, Fuzzing, and Software Engineering
- Theory and techniques for Symbolic Execution
 - 2 lectures? (including KLEE?)
- SMT and incremental SMT for Symbolic Execution
 - 2 lectures
- Are there other topics to be covered?
 - Open to suggestions

Papers on General Topics (for first Lecture Part)

- James King: Symbolic execution and program testing
- A Survey of Symbolic Execution Techniques. ROBERTO BALDONI, EMILIO COPPA, DANIELE CONO D'ELIA, CAMIL DEMETRESCU, and IRENE FINOCCHI

Dynamic Symbolic Execution

- Godefroid, P., Levin, M. Y., Molnar, D. 2008. Automated whitebox fuzz testing <https://www.ndss-symposium.org/ndss2008/automated-whitebox-fuzz-testing/>
- Godefroid, P., Klarlund, N., Sen, K.: DART: Directed Automated Random Testing
-

Papers/Seminars by participants, organized by topics

Papers in boldface are recommended for reviewing/discussion

Loops

- **Avgerinos, T., Rebert, A., Cha, S.K., Brumley, D.: Enhancing Symbolic Execution with Veritesting** <https://dl.acm.org/doi/abs/10.1145/2568225.2568293>
- Fast Acceleration of Ultimately Periodic Relations, Marius Bozga, Radu Iosif, and Filip Konečný

Path Explosion

- **Efficient State Merging in Symbolic Execution. V Kuznetsov, J Kinder, S Bucur, G Candea**

Environment interaction

- **S2E: a platform for in-vivo multi-path analysis of software systems** <https://dl.acm.org/doi/abs/10.1145/1961296.1950396>

- Synthesizing framework models for symbolic execution
<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7886900>
- Android testing via synthetic symbolic execution
<https://www.comp.nus.edu.sg/~abhik/pdf/ase18.pdf>
- Kenneth L. McMillan: Lazy Annotation for Program Testing and Verification. [CAV 2010](#): 104-118

Memory models

- **Coppa et al: Memory models in symbolic execution?**
<https://onlinelibrary.wiley.com/doi/epdf/10.1002/stvr.1722>

Symbolic Execution Challenges

- Benchmarking the Capability of Symbolic Execution Tools with Logic Bombs
<https://ieeexplore.ieee.org/document/8443109>

SMT for symbolic execution

- **Blake Loring, Duncan Mitchell, Johannes Kinder: Sound regular expression semantics for dynamic symbolic execution of JavaScript**
- Roberto Amadini, Mak Andrlon, Graeme Gange, Peter Schachte, Harald Søndergaard, Peter J. Stuckey: Constraint Programming for Dynamic Symbolic Execution of JavaScript
- [Hristina Palikareva](#), [Cristian Cadar](#): Multi-solver Support in Symbolic Execution. CAV 2013. 53-68

Concurrency

- **Con2colic testing**
<https://dl.acm.org/doi/10.1145/2491411.2491453>
- Symbolic execution of multithreaded programs from arbitrary program contexts
<https://dl.acm.org/doi/10.1145/2660193.2660200>
- **Assertion guided symbolic execution of multithreaded programs**
<https://dl.acm.org/doi/10.1145/2786805.2786841>

Other

- Cristian Cadar, Vijay Ganesh, Peter M. Pawlowski, David L. Dill, and Dawson R. Engler. 2006. EXE: Automatically Generating Inputs of Death
- **JPF-SE: A Symbolic Execution Extension to Java PathFinder**
- concolic tool: CREST <https://github.com/jburnim/crest>
- **Symbolic execution with SymCC: Don't interpret, compile!**
<https://www.usenix.org/conference/usenixsecurity20/presentation/poeplau>
- Systematic comparison of symbolic execution systems: intermediate representation and its generation <https://dl.acm.org/doi/10.1145/3359789.3359796>
The paper compares the impacts of choosing the code or intermediate representation a symbolic engine executes. Though it is technical, they introduce symbolic execution engines and describe quite nicely when and why an Intermediate Representation plays a role in such context.
- [Sergey Mechtaev](#), [Alberto Griggio](#), Alessandro Cimatti, [Abhik Roychoudhury](#): Symbolic execution with existential second-order constraints. [ESEC/SIGSOFT FSE 2018](#): 389-399

- Suzette Person, [Matthew B. Dwyer](#), [Sebastian G. Elbaum](#), [Corina S. Pasareanu](#): Differential symbolic execution. [SIGSOFT FSE 2008](#): 226-237
- Corina S. Pasareanu, [Quoc-Sang Phan](#), [Pasquale Malacaria](#): Multi-run Side-Channel Analysis Using Symbolic Execution and Max-SMT. [CSF 2016](#): 387-400

Further Reading

Practicals

- Learning to use existing symbolic execution tools.
- Use of existing tools to analyse/test, ... software/protocols, ...
- Writing your own SymEx/Fuzzing tool
- JDart?
- ?

Tutorials/Videos Possible links that could be used

- Lecture by Naik <https://www.youtube.com/watch?v=QrtGOrSrVPQ>
- Tutorial on Symbolic Execution using Angr
<https://github.com/ercoppa/symbolic-execution-tutorial>
- Klee tutorials <https://klee.github.io/tutorials/>