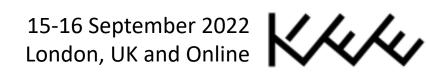
# Symbolic Execution Projects from the Software Reliability Group

Cristian Cadar, Julian Büning, Frank Busse, Alastair Donaldson, Pritam Gharat, Thom Hughes, Timotej Kapus, Martin Nowack, Jordy Ruiz, Daniel Schemmel, Arindam Sharma, Ahmed Zaki







## Symbolic Execution for Evolving Software

## **Automated Chopped Symbolic Execution**

- Automatically skip parts of the code that are irrelevant to a patch
- Talk by Martin tomorrow

# **Product Programs for Cross-Version Symbolic Execution**

- Leverages ideas from product program constructions (used to reason about non-interfence)
- Reason about multiple versions in the same symbolic execution instance

## Long-Running Deterministic Symbolic Execution

#### **Memoised Symbolic Execution**

- Allow "forever" runs of symbolic execution by:
  - saving the current run to disk
  - incrementally bringing it back into memory

#### **Deterministic Memory Allocation**

- Effectiveness of memoised symbolic execution and other techniques depends on determinism across runs
- KDAlloc is a memory allocator specifically designed for symex:
  - is cross-run and cross-path deterministic
  - maximises the probability of finding memory-safety bugs
  - keeps a low memory and performance overhead
- Talk by Daniel just before lunch

### **Exploration Heuristics**

#### **Pending Constraints**

- Aggressively prioritize paths whose constraints can be solved via caching or seeding
- Defer pending constraints until they really need to be solved
- Talk by Frank tomorrow

#### **Confirming Static Analysis Reports**

- Guide symbolic execution to follow the traces in SA reports
- If successful, the bug report is confirmed and an input produced
- Poster talk by Frank after lunch

#### **Approximating Floating Point via Fixed Point**

- SMT solvers for floating-point arithmetic are notoriously slow
- Would an approximation via fixed-point arithmetic be fast and precise enough?
- Our approach defers fixed-point reasoning as much as possible, with concrete operations staying in the floating-point domain

# Symbolic Execution Projects from the Software Reliability Group

- Automated Chopped Symbolic Execution
- Multi-Version Testing with Product Programs
- Confirming Static Analysis Bug Reports
- Memoised Symbolic Execution
- Deterministic Memory Allocation
- Pending Constraints
- Approximating Floating Point via Fixed Point