TracerX: Dynamic Symbolic Execution with Interpolation

Joxan Jaffar*, Rasool Maghareh†, Sangharatna Godboley‡

***National University of Singapore**
joxan@comp.nus.edu.sg

†Heterogeneous Compiler Lab, Huawei, Canada
rasool.maghareh@huawei.com

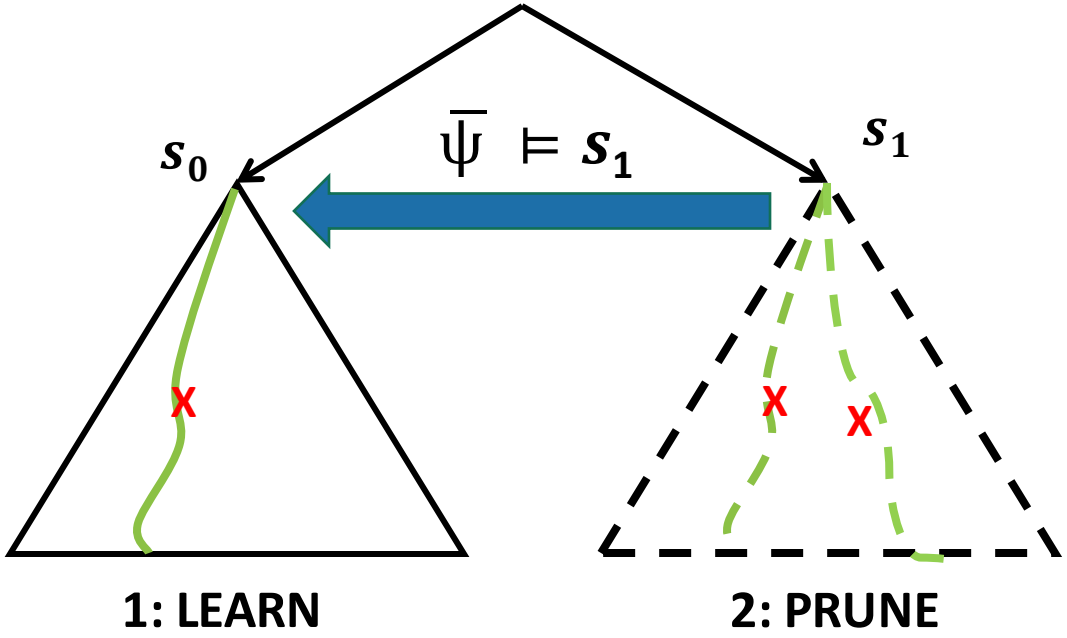
‡National Institute of Technology Warangal, India‡
sanghu@nitw.ac.in

Test-Comp 2021
April 2021

TracerX: Dynamic Symbolic Execution with Interpolation

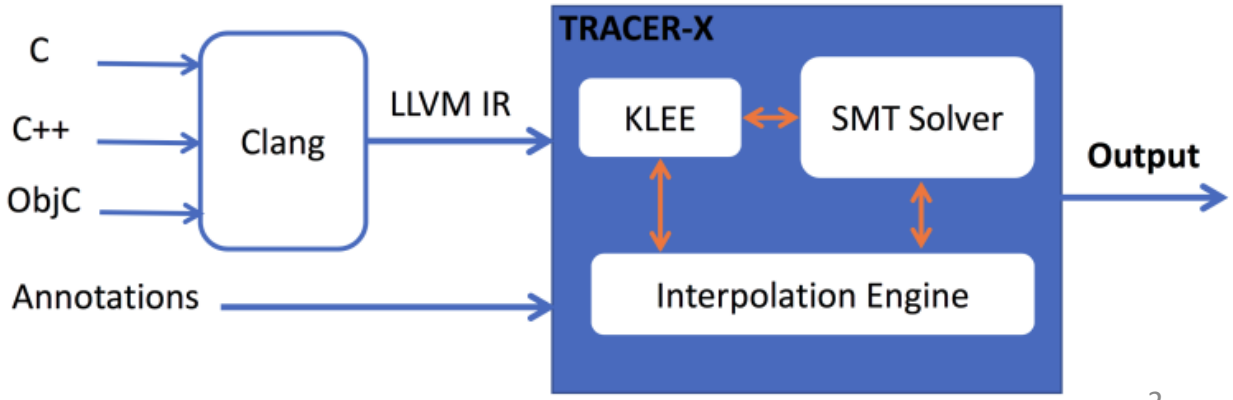
- **Website:** https://tracer-x.github.io/
- **Github:** https://github.com/tracer-x/

Interpolation ($\bar{\psi}$): Preserve the discovered infeasible paths and prune subsets
Subsumption: Prune similar subtrees if interpolant S_1 Implies interpolant



Architecture:

- **Forward run** similar to KLEE 1.0 (**DFS strategy**)
- **Backward run** generates interpolants
- **Subsumption** check using SMT calls



Test-Comp'21 Results (6th rank)

<table>
<tr><th></th><th>Coverage-Error</th><th>Cover-Branches</th></tr>
<tr><td>Arrays</td><td rowspan="10">KLEE > TracerX

• KLEE runs with Random Strategy
• TracerX runs with DFS</td><td rowspan="2">TracerX > KLEE</td></tr>
<tr><td>BitVectors</td></tr>
<tr><td>Control Flow</td><td>TracerX == KLEE</td></tr>
<tr><td>ECA</td><td rowspan="6">TracerX > KLEE</td></tr>
<tr><td>Floats</td></tr>
<tr><td>Heap</td></tr>
<tr><td>Loops</td></tr>
<tr><td>Recursive</td></tr>
<tr><td>Sequentialized</td></tr>
<tr><td>XCSP</td><td rowspan="2">KLEE > TracerX</td></tr>
<tr><td>BusyBox-MemSafety</td><td>All tools 0</td></tr>
<tr><td>DeviceDriversLinux64</td><td>All tools 0</td><td rowspan="2">TracerX > KLEE</td></tr>
<tr><td>Combinations</td><td>-</td></tr>
<tr><td>SQLite-MemSafety</td><td>-</td><td>All tools 0</td></tr>
<tr><td>MainHeap</td><td>-</td><td>TracerX > KLEE</td></tr>
</table>