A Data Set of Program Invariants and Error Paths

Dirk Beyer
LMU Munich, Germany
Witnesses from Software Verification

Programs available in public benchmark repository of the verification-research community [1]:
https://github.com/sosy-lab/sv-benchmarks

Witnesses available in the data set [2] and described in this paper [6].
Example: Witness with Invariants

What is a witness?
An automaton that contains invariants (or error paths).

```c
int main() {
    unsigned int x = nondet();
    unsigned int y = x;
    while (x < 1024) {
        x = x + 1;
        y = y + 1;
    }
    // Safety property
    assert(x == y);
    return 0;
}
```
Main Purpose of Witnesses: Result Validation

Software-verification community mostly interested in result validation \[4, 3, 5\].

- Validate untrusted results
- Easier than full verification
Possible Research Questions

What else can we do with these nice verification artifacts?

- Visualization of error paths
- Annotations of programs with invariants
- Classification of bugs
- Classification of program invariants
- Can violation witnesses improve understanding of bugs?
- Can correctness witnesses improve understanding the correctness proof?
- Is it possible to predict (and later check) program invariants?
Statistics about the Witnesses

<table>
<thead>
<tr>
<th>Witness Measure</th>
<th>All Witnesses</th>
<th>Correctness Witnesses</th>
<th>Violation Witnesses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Median</td>
<td>Mean</td>
<td>Max</td>
</tr>
<tr>
<td>Number of States</td>
<td>27</td>
<td>950</td>
<td>1.5 \cdot 10^6</td>
</tr>
<tr>
<td>Number of Transitions</td>
<td>27</td>
<td>1200</td>
<td>1.5 \cdot 10^6</td>
</tr>
<tr>
<td>Number of Invariants</td>
<td>3.0</td>
<td>380</td>
<td>0.70 \cdot 10^6</td>
</tr>
<tr>
<td>Length of All Invariants</td>
<td>270</td>
<td>35000</td>
<td>9.6 \cdot 10^6</td>
</tr>
</tbody>
</table>


Data set is result of 450 days of CPU time, distributed over 168 computers.
Purpose of a Data Set

- Analyze invariants and error paths
- Gain insights from data analysis
- Almost no analysis was done yet for witnesses

Remember the research questions:
- Can violation witnesses improve understanding bugs?
- Can correctness witnesses improve understanding the correctness?
- Is it possible to predict (and later check) program invariants?

Lots of papers need to be written!

Thanks! Questions?


