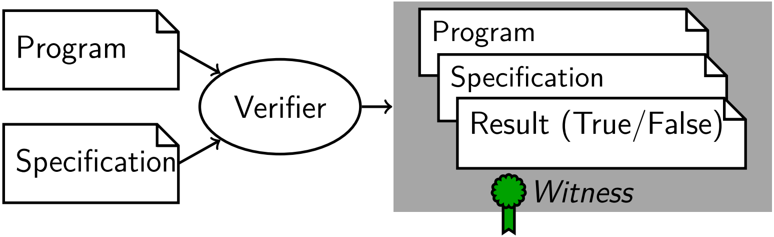
A Data Set of Program Invariants and Error Paths

Dirk Beyer

LMU Munich, Germany

Witnesses from Software Verification



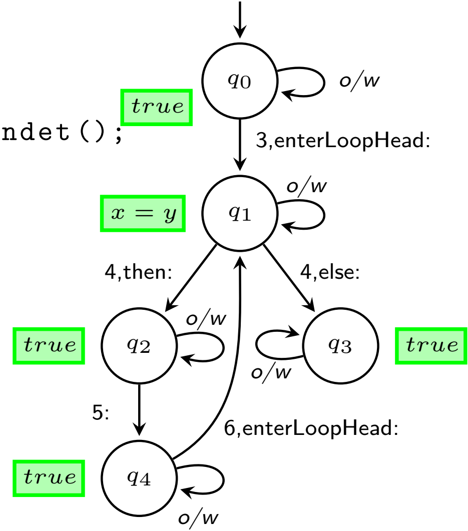
Programs available in public benchmark repository of the verification-research community [1]:
https://github.com/sosy-lab/sv-benchmarks

Witnesses available in the data set [2] and described in this paper [6].

Example: Witness with Invariants

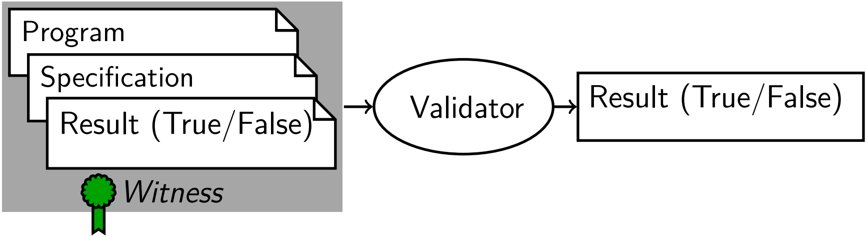
What is a witness?
An automaton that contains invariants (or error paths).

```
1 int main() {
2   unsigned int x = nondet();
3   unsigned int y = x;
4   while (x < 1024) {
5     x = x + 1;
6     y = y + 1;
7   }
8   // Safety property
9   assert(x == y);
10  return 0;
11 }
```



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

Witness Measure	All Witnesses				Correctness Witnesses				Violation Witnesses			
	Median	Mean	Max	Sum	Median	Mean	Max	Sum	Median	Mean	Max	Sum
Number of States	27	950	$1.5 \cdot 10^6$	$58 \cdot 10^6$	23	1 100	$1.0 \cdot 10^6$	$39 \cdot 10^6$	31	750	$1.5 \cdot 10^6$	$19 \cdot 10^6$
Number of Transitions	27	1 200	$1.5 \cdot 10^6$	$74 \cdot 10^6$	24	1 400	$0.90 \cdot 10^6$	$52 \cdot 10^6$	31	860	$1.5 \cdot 10^6$	$22 \cdot 10^6$
Number of Invariants					3.0	380	$0.70 \cdot 10^6$	$3.1 \cdot 10^6$				
Length of All Invariants					270	35 000	$9.6 \cdot 10^6$	$290 \cdot 10^6$				

The paper [6] provides more statistics,
and a detailed description of the structure of the data set.

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?

References I

- Beyer, D.: SV-Benchmarks: Benchmark set of 8th Intl. Competition on Software Verification (SV-COMP 2019). Zenodo (2019). https://doi.org/10.5281/zenodo.2598729

- Beyer, D.: Verification witnesses from SV-COMP 2019 verification tools. Zenodo (2019). https://doi.org/10.5281/zenodo.2559175

- Beyer, D., Dangl, M., Dietsch, D., Heizmann, M.: Correctness witnesses: Exchanging verification results between verifiers. In: Proc. FSE. pp. 326–337. ACM (2016). https://doi.org/10.1145/2950290.2950351

- Beyer, D., Dangl, M., Dietsch, D., Heizmann, M., Stahlbauer, A.: Witness validation and stepwise testification across software verifiers. In: Proc. FSE. pp. 721–733. ACM (2015). https://doi.org/10.1145/2786805.2786867

- Beyer, D., Dangl, M., Lemberger, T., Tautschnig, M.: Tests from witnesses: Execution-based validation of verification results. In: Proc. TAP. pp. 3–23. LNCS 10889, Springer (2018). https://doi.org/10.1007/978-3-319-92994-1_1

- Beyer, D.: A data set of program invariants and error paths. In: Proc. MSR. pp. 111–115. IEEE (2019). https://doi.org/10.1109/MSR.2019.00026