Case Study on Verification-Witness Validators

Where We Are and Where We Go

Dirk Beyer and Jan Strejček
LMU Munich, Germany and Masaryk University, Czechia

December 5, 2022, at SAS 2022

Proc. SAS 2022, doi:10.1007/978-3-031-22308-2_8

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

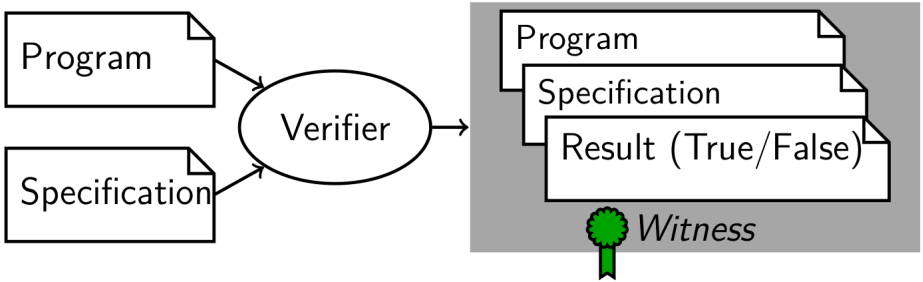
SoSy-Lab Software Systems

Static Analysis

- Goal: Verify correctness of computer programs
- Problem: Programming bugs in analysis tools lead to wrong results
- Solution: Witness-based result validation [3, 1, 2]
- State of the art: Yearly evaluation by SV-COMP gives overview of
  - 76 tools for static analysis (accumulated)
  - 10 validators for verification witnesses

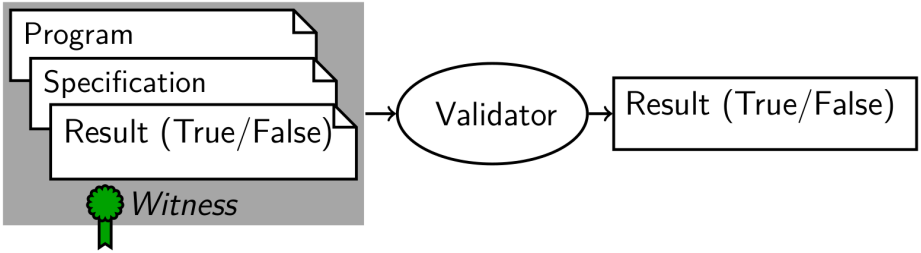
Software Verification with Witnesses

The verification witness explains and justifies the verification result.



[3, Proc. FSE 2015] [1, Proc. FSE 2016]

Witness Validation



- Validate untrusted results
- Easier than full verification

(Some) Validators are Buggy — Violation Witnesses

Category	Witnesses	CPAchecker	CPA-w2t	CProver-w2t	Dartagnan	MetaVal	nitwit	Symbiotic -Witch	uAutomizer
ReachSafety	5177	28	12	2	-	0	10	0	0
MemSafety	2804	0	0	26	-	2	-	0	0
ConcurrencySafety	1293	40	-	-	0	-	-	-	-
NoOverflows	167	0	0	0	-	0	-	0	0
Termination	56	21	-	-	-	0	-	-	0
SoftwareSystems	5903	5	0	27	-	0	0	51	4

Numbers of invalid violation witnesses
(resulting from incorrect verification results)
validated by witness validators

(Some) Validators are Buggy — Correctness Witnesses

Category	Witnesses	CPAchecker	MetaVal	UAutomizer
ReachSafety	894	0	315	3
MemSafety	326	-	0	0
NoOverflows	300	0	36	0
SoftwareSystems	888	0	403	0

Numbers of invalid correctness witnesses
(resulting from incorrect verification results)
validated by witness validators

Current Interpretation of Validator Output in a Compatition

Output for a violation witness

- `false` $\longrightarrow$ witness is confirmed $\longrightarrow$ verifier receives 1 point
- `true` or `unknown` $\longrightarrow$ witness is not confirmed $\longrightarrow$ verifier receives 0 points

Current Interpretation of Validator Output in a Compatition

Output for a violation witness

- `false` $\longrightarrow$ witness is confirmed $\longrightarrow$ verifier receives 1 point
- `true` or `unknown` $\longrightarrow$ witness is not confirmed $\longrightarrow$ verifier receives 0 points

Output for a correctness witness

- `true` $\longrightarrow$ witness is confirmed $\longrightarrow$ verifier receives 2 points
- `false` or `unknown` $\longrightarrow$ witness is not confirmed $\longrightarrow$ verifier receives 0 points

Current Interpretation of Validator Output in a Compatition

Output for a violation witness

- `false` $\longrightarrow$ witness is confirmed $\longrightarrow$ verifier receives 1 point
- `true` or `unknown` $\longrightarrow$ witness is not confirmed $\longrightarrow$ verifier receives 0 points

Output for a correctness witness

- `true` $\longrightarrow$ witness is confirmed $\longrightarrow$ verifier receives 2 points
- `false` or `unknown` $\longrightarrow$ witness is not confirmed $\longrightarrow$ verifier receives 0 points

Validators not used to refute a witness!

New Interpretation of Validator Output

Output for a violation witness

- `false` $\longrightarrow$ witness is confirmed
- `true` $\longrightarrow$ witness is refuted
- `unknown` $\longrightarrow$ witness is not confirmed/refuted

A violation witness should be refuted if it represents no program execution violating the considered property.

New Interpretation of Validator Output

Output for a violation witness

- `false` $\longrightarrow$ witness is confirmed
- `true` $\longrightarrow$ witness is refuted
- `unknown` $\longrightarrow$ witness is not confirmed/refuted

A violation witness should be refuted if it represents no program execution violating the considered property.

Output for a correctness witness

- `true` $\longrightarrow$ witness is confirmed
- `false` $\longrightarrow$ witness is refuted
- `unknown` $\longrightarrow$ witness is not confirmed

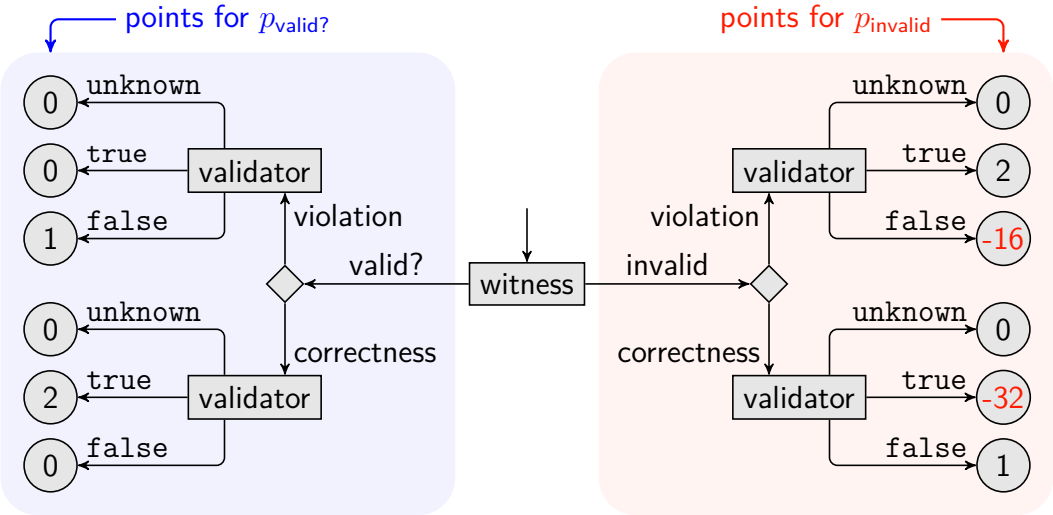
A correctness witness should be refuted if it contains an invariant that does not hold or if the program violates the considered property.

Competition Track for Witness Validators in SV-COMP 2023

Benchmark set:

- Witnesses from SV-COMP 2023 itself
- Invalid witnesses = witnesses of incorrect verification results
- Valid? witnesses = witnesses of correct verification results (may be incorrect)

Scoring Schema for One Category



Competition Track for Witness Validators in SV-COMP 2023

Competition track:

- Same deadlines and schedule as SV-COMP 2023
- Pre-runs of verifiers produce benchmarks for preruns of validators
- Officially only one category **Overall**
- Overall score computed by the same procedure as in SV-COMP from scores in individual categories

Conclusion

- Validators are an important part of the verification eco system
- They include bugs, just like verifiers
- We proposed a competition track on validators
  $\rightarrow$ community accepted the proposal, and SAS reviewers accepted the paper
- From SV-COMP 2023, there will be a yearly evaluation of validators
- Paper [4] in Proc. SAS 2022: doi:10.1007/978-3-031-22308-2_8

References I

[1] 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

[2] Beyer, D., Dangl, M., Dietsch, D., Heizmann, M., Lemberger, T., Tautschnig, M.: Verification witnesses. ACM Trans. Softw. Eng. Methodol. **31**(4), 57:1–57:69 (2022). https://doi.org/10.1145/3477579

[3] 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

[4] Beyer, D., Strejček, J.: Case study on verification-witness validators: Where we are and where we go. In: Proc. SAS. pp. 1–15. LNCS 13790, Springer (2022). https://doi.org/10.1007/978-3-031-22308-2_8