

SV-COMP 2025

14th Competition on Software Verification

Dirk Beyer and Jan Strejček
(Competition Chairs)

2025-10-28, Dagstuhl



Report in Proc. TACAS 2025, doi:10.1007/978-3-031-90660-2_9

Supported by:



HUAWEI



Motivation - Goals

1. Community suffers from unreproducible results
→ Establish set of benchmarks
2. Publicity for tools that are available
→ Provide state-of-the-art overview
3. Support the development of verification tools
→ Give credits and visibility to developers
4. Establish and develop standards
→ Specification language, Property definitions,
Benchmark definitions, Witness formats, Validation process

Schedule of Sessions at ETAPS

Session 1:

- ▶ Competition report by organizers
- ▶ System presentations
- ▶ Short discussion

Session 2:

- ▶ Open jury meeting, community discussion, moderated by organizers

Procedure – Time Line

Five Steps – Five Mile Stones:

- ▶ Benchmark submission
- ▶ System submission for training and qualification
- ▶ Benchmark freeze
- ▶ Final system submission
- ▶ Notification of results (approved by teams)

The mile stones are further supported by several deadlines.

Verification Problem

Input:

- ▶ C program (GNU/ANSI C standard) and property
 - Reachability safety
 - No overflow
 - Memory safety (valid-deref, valid-free, valid-memtrack)
 - Memory cleanup
 - Termination
 - No Data race
- ▶ or Java program and property
 - Assertion validity
 - No runtime exception

Output:

- ▶ TRUE + correctness witness (property holds)
- ▶ FALSE + violation witness (property does not hold)
- ▶ UNKNOWN (failed to compute result)

Validation Problem

Input:

- ▶ C program (GNU/ANSI C standard)
- ▶ property
- ▶ correctness or violation witness

Output:

- ▶ TRUE = correctness witness confirmed /
 violation witness refuted
- ▶ FALSE = correctness witness refuted /
 violation witness confirmed
- ▶ UNKNOWN = failed to decide

Environment

Machines (1000 \$ consumer machines):

- ▶ CPU: 3.4 GHz 64-bit Quad-Core CPU
- ▶ RAM: 33 GB
- ▶ OS: GNU/Linux (Ubuntu 24.04)
- ▶ < 2026 : execution on "bare metal",
 ≥ 2026 : OCI images + podman

Resource limits for **verification**:

- ▶ 15 GB memory
- ▶ 15 min CPU time on 4 processing units

Resource limits for **validation**:

- ▶ 7 GB memory
- ▶ 15 min CPU time on 2 processing units (correctness)
- ▶ 1.5 min CPU time on 2 processing units (violation)

Scoring Schema

Common principles: Ranking measure should be

- ▶ easy to understand
- ▶ reproducible
- ▶ computable in isolation for one tool for verification track

SV-COMP:

- ▶ Ranking measure reflects the quality of verification work
- ▶ Expressed by a community-agreed score
- ▶ Tie-breaker is CPU time

For the validation track, the verdicts of the witnesses are based on voting, because we cannot afford the manual effort necessary to establish the ground truth for thousands of generated witnesses.

Scoring Schema for Verification Track (2025, unchanged)

Reported result	Points	Description
FALSE correct	+1	Error found and confirmed
FALSE incorrect	-16	False alarm (imprecise analysis)
TRUE correct	+2	Proof found and confirmed
TRUE incorrect	-32	Missed bug (unsound analysis)
UNKNOWN	0	Failure, out of resources, . . .

Scoring Schema for Validation Track (2025, unchanged)

Reported result	Points	Description
on correctness witnesses		
FALSE correct	+1	Witness was correctly refuted
FALSE incorrect	-16	Witness was refuted but it is correct
TRUE correct	+2	Witness was correctly confirmed
TRUE incorrect	-32	Witness was confirmed but it is incorrect
on violation witnesses		
FALSE correct	+1	Witness was correctly confirmed
FALSE incorrect	-16	Witness was confirmed but it is incorrect
TRUE correct	+2	Witness was correctly refuted
TRUE incorrect	-32	Witness was refuted but it is correct

Fair and Transparent

Jury:

- ▶ Team: one member of each participating candidate
- ▶ Term: one year (until next participants are determined)

Systems:

- ▶ All systems are openly available at Zenodo
- ▶ Essential information available in FM-Tools repository
- ▶ FM-Tools to announce new versions via MR
<https://fm-tools.sosy-lab.org>
- ▶ Open submission and discussion
- ▶ Configurations and Setup in GitLab repository
→ Integrity and reproducibility guaranteed

80 Competition Candidates in 2025

Qualification:

- ▶ 62 in verification track (2026: 74 total, 38 active)
- ▶ 18 in validation track
- ▶ One person can participate with different tools
- ▶ One tool can participate with several configurations (frameworks, no tool-name inflation), all tools are conceptually different, no parameter testing
- ▶ Peer review

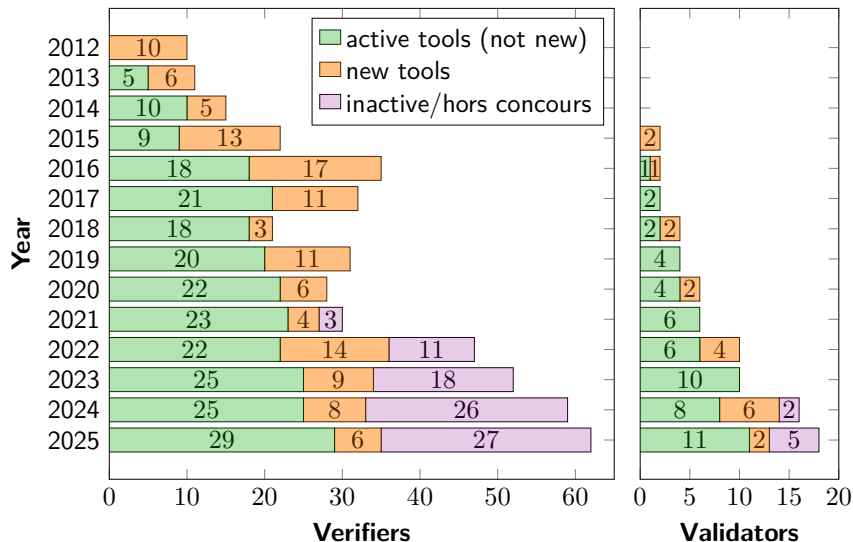
Benchmark quality:

- ▶ Community effort, documented on GitLab

Role of organizer:

- ▶ Just service: Advice, Technical Help, Executing Runs, Evaluation

Number of Participants (2025)



Benchmark Sets

- ▶ Everybody can submit benchmarks (conditions apply)
- ▶ Eight meta categories when closed:

Category	Number of tasks
ReachSafety	11 268
MemSafety	4 042
ConcurrencySafety	3 175
NoOverflows	8 211
Termination	2 328
SoftwareSystems	4 329
Overall	33 353
JavaOverall	673

- ▶ Scores are normalized: every category has same weight

Reproducibility

- ▶ SV-Benchmarks:
<https://gitlab.com/sosy-lab/benchmarking/sv-benchmarks>
- ▶ SV-COMP Setup:
<https://gitlab.com/sosy-lab/sv-comp/bench-defs>
- ▶ Resource Measurement and Process Control:
<https://github.com/sosy-lab/benchexec>
- ▶ Archives:
<https://gitlab.com/sosy-lab/benchmarking/fm-tools>
- ▶ Witnesses:
<https://doi.org/10.5281/zenodo.15012077>

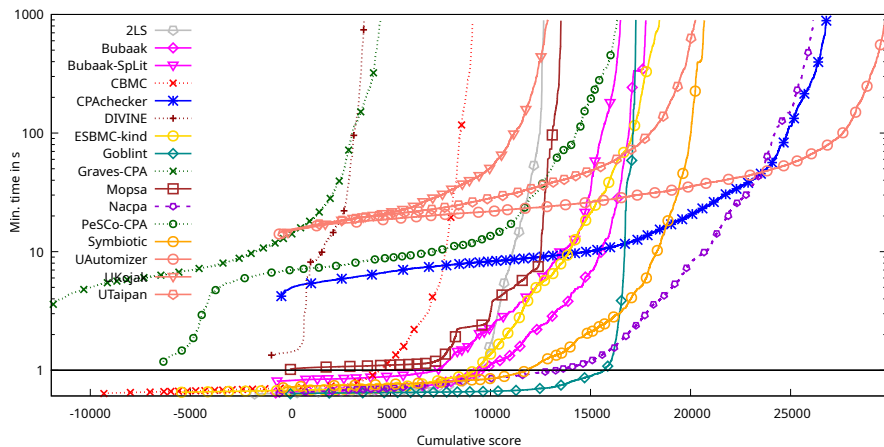
Computation Effort:

- ▶ 942 284 verification runs (6.3 years of CPU time),
pre-runs: 3.3 million verification runs (17 years of CPU time)
- ▶ 21.8 million validation runs (7.0 years of CPU time),
pre-runs: 88 million validation runs

Reproducibility

Content	DOI	Reference
Verification Tasks	10.5281/zenodo.15012096	[8]
Competition Results	10.5281/zenodo.15012085	[7]
FM-Tools (Verifiers and Validators)	10.5281/zenodo.15055359	[2]
Verification Witnesses	10.5281/zenodo.15012077	[9]
BENCHEXEC	10.5281/zenodo.15007216	[12]
CoVeriTeam	10.5281/zenodo.11193690	[10]

Results – Example: Overall



Impact / Achievements

- ▶ Large benchmark set of verification tasks
→ established and used in many papers
for experimental evaluation
- ▶ Good overview over state-of-the art
→ covers model checking and program analysis
- ▶ Participants have an archived track record
of their achievements
- ▶ Infrastructure and technology for
controlling the benchmark runs:
[BENCHEXEC](#) [5], [BENCHCLOUD](#) [3],
[FM-WECK](#) [11], [FM-TOOLS](#) [1], [COVERITEAM](#) [4]

Competition Report [6] and System Descriptions
are archived in Proceedings of TACAS 2025

New Development in 2025

- ▶ Organization committee
- ▶ More verification tasks (in each meta category)
- ▶ New Java property: no runtime exceptions (demo)
- ▶ Handcrafted witnesses in validation track
- ▶ New base categories (most prominently Intel-TDX-Module)
- ▶ Witnesses in format 2.0 also for violation witnesses
- ▶ Split hors concours into *inactive* and *meta verifiers*
- ▶ Void tasks and empty categories excluded from score computation
- ▶ Medals only for positive scores
- ▶ Sponsorship program with Huawei

Better Support of Witness Format 2.0 by Validators

Validator	Witness Format 1.0		Witness Format 2.0	
	Correctness	Violation	Correctness	Violation
CONCURRENTW2T		✓		
CPACHECKER	✓	✓	✓	✓
CPA-w2T [⊗]		✓		
CProver-w2T [⊗]		✓		
DARTAGNAN		✓		
GOBLINT			✓	
GWIT [⊗]		✓		
JCWIT [⊗]	✓			
LIV	✓		✓	
META Va	✓	✓	✓	✓
METAVAL++ ^{new}			✓	
MOPSA			✓	
NITWIT [⊗]		✓		
SYMBIOTIC-WITCH		✓		
UAUTOMIZER	✓	✓	✓	✓
UREFEE	✓		✓	
WIT4JAVA		✓		
WITCH				✓

Voting of Validators: Violation Witnesses 1.0

witnesses format	1.0	correct	34%
validators	11 (2 for Java)	wrong	6%
witnesses	125214	undecided	59%

refutations	6	1	0	0	0	0	0	0	0	0
	5	14	5	1	0	0	0	0	0	0
	4	164	72	19	63	2	0	0	0	0
	3	2940	1933	3463	235	132	13	0	0	0
	2	2839	7481	7394	1873	430	320	180	0	0
	1	6922	15420	20894	5640	5406	4079	677	164	0
	0	2198	7188	6866	5944	4131	7429	1496	847	339
		0	1	2	3	4	5	6	7	8
		witness confirmations								

Voting of Validators: Violation Witnesses 2.0

witnesses format	2.0	correct	28%
validators	4	wrong	1%
witnesses	29819	undecided	71%

refutations	4	21	0	0	0	0
	3	31	4	0	0	0
	2	248	2006	81	0	0
	1	2794	8740	1159	605	0
	0	2789	3686	1864	4997	794
		0	1	2	3	4
		witness confirmations				

Voting of Validators: Correctness Witnesses 1.0 and 2.0

refutations	witnesses format		1.0		correct		52%
	validators		6 (1 for Java)		wrong		0%
	witnesses		195918		undecided		48%
2	104	14	16	0	0	0	
1	864	1105	663	67	35	0	
0	31340	60818	64778	22927	11039	2148	
	0	1	2	3	4	5	
witness confirmations							

refutations	witnesses format 2.0						correct	71%	
	validators 8						wrong	0%	
	witnesses 87147						undecided	29%	
1	530	604	643	159	7	6	0	0	0
0	8497	15118	19540	18854	17546	4014	1261	284	84
	0	1	2	3	4	5	6	7	8
witness confirmations									

Planned Changes for 2026

- ▶ Simpler tool registration and qualification process
- ▶ Smoke tests via FM-Weck
- ▶ Benchmark-category renaming and property renaming
- ▶ *FalsificationOverall* will include also termination benchmarks
- ▶ New *True-Overall* category (counterpart of *FalsificationOverall*)
- ▶ Refinement of the termination property
 - ▶ no-cycle, bounded-recursion, no-blocking,...
- ▶ No assumption that memory allocation always succeeds
- ▶ Allow un-preprocessed C programs
- ▶ Reintroduce wall time track as a demo category
- ▶ New rules for AI-based tools
- ▶ Instant score results (during preruns)
- ▶ Instant (but incomplete) validation results (preruns)

Planned Changes for 2026 (cont.)

- ▶ Witness format 2.1:
 - ▶ termination and non-termination witnesses
 - ▶ concurrency support
 - ▶ function contracts
- ▶ No support of correctness witnesses in format 1.0 (except for Java)
- ▶ Lower time limit for validation of correctness witnesses
- ▶ No weighting between wrong/correct validation tasks in validation track

Sponsorship

New sponsorship agreement with Huawei!

- ▶ Travel support
- ▶ Demo category with awards
- ▶ Hardware support
- ▶ Student assistants

Huawei will contribute more industrial benchmark programs, will define a demo category on those, and assign prices.

Thanks to:

- ▶ TACAS (PC Chairs + TACAS SC, thanks!)
- ▶ Organization committee
- ▶ Competition jury/program committee
- ▶ Participants from community (111 people)
- ▶ Sponsors: Huawei and LMU Munich
- ▶ Next we celebrate the winners

Report:



References I

- [1] Beyer, D.: Find, use, and conserve tools for formal methods. In: Proc. Festschrift Podelski 65th Birthday. Springer (2024), available online: https://www.sosy-lab.org/research/pub/2024-Podelski65.Find_Use_and_Conserve_Tools_for_Formal_Methods.pdf
- [2] Beyer, D.: FM-Tools Release 2.2: Data set of metadata about tools for formal methods (SV-COMP 2025, Test-Comp 2025). Zenodo (2025). <https://doi.org/10.5281/zenodo.15055359>
- [3] Beyer, D., Chien, P.C., Jankola, M.: BENCHCLOUD: A platform for scalable performance benchmarking. In: Proc. ASE. pp. 2386–2389. ACM (2024). <https://doi.org/10.1145/3691620.3695358>
- [4] Beyer, D., Kanav, S.: COVERITEAM: On-demand composition of cooperative verification systems. In: Proc. TACAS. pp. 561–579. LNCS 13243, Springer (2022). https://doi.org/10.1007/978-3-030-99524-9_31
- [5] Beyer, D., Löwe, S., Wendler, P.: Reliable benchmarking: Requirements and solutions. Int. J. Softw. Tools Technol. Transfer **21**(1), 1–29 (2019). <https://doi.org/10.1007/s10009-017-0469-y>
- [6] Beyer, D., Strejček, J.: Improvements in software verification and witness validation: SV-COMP 2025. In: Proc. TACAS (3). pp. 151–186. LNCS 15698, Springer (2025). https://doi.org/10.1007/978-3-031-90660-2_9

References II

- [7] Beyer, D., Strejček, J.: Results of the 14th Intl. Competition on Software Verification (SV-COMP 2025). Zenodo (2025).
<https://doi.org/10.5281/zenodo.15012085>
- [8] Beyer, D., Strejček, J.: SV-Benchmarks: Benchmark set for software verification (SV-COMP 2025). Zenodo (2025).
<https://doi.org/10.5281/zenodo.15012096>
- [9] Beyer, D., Strejček, J.: Verification witnesses from verification tools (SV-COMP 2025). Zenodo (2025).
<https://doi.org/10.5281/zenodo.15012077>
- [10] Beyer, D., Wachowitz, H.: Coveriteam Release 1.2.1. Zenodo (2024).
<https://doi.org/10.5281/zenodo.11193690>
- [11] Beyer, D., Wachowitz, H.: FM-WECK: Containerized execution of formal-methods tools. In: Proc. FM. pp. 39–47. LNCS 14934, Springer (2024).
https://doi.org/10.1007/978-3-031-71177-0_3
- [12] Wendler, P., Beyer, D.: sosy-lab/benchexec: Release 3.29. Zenodo (2025).
<https://doi.org/10.5281/zenodo.15007216>