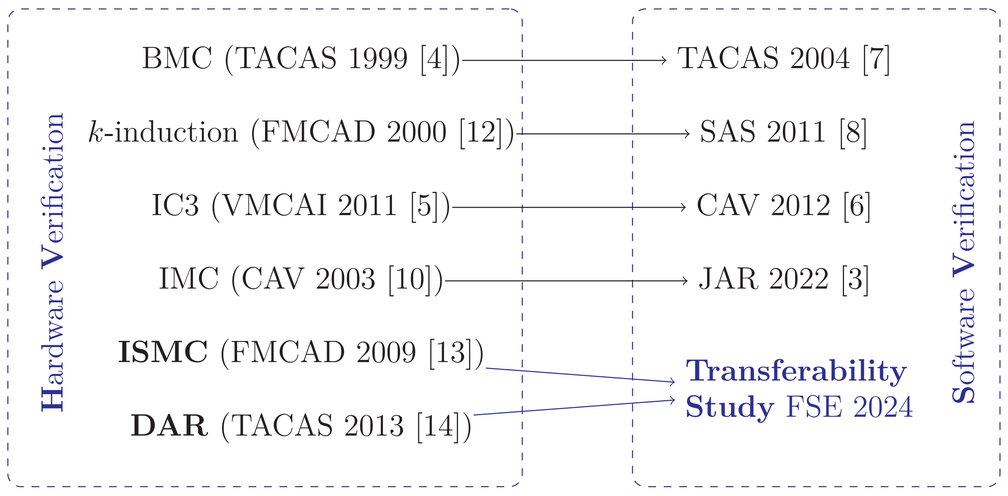
A Transferability Study of Interpolation-Based Hardware Model Checking for Software Verification

Dirk Beyer, Po-Chun Chien, Marek Jankola, and Nian-Ze Lee

{dirk.beyer,po-chun.chien,marek.jankola,nian-ze.lee}@sosy.ifi.lmu.de

Presentation at FSE 2024: 11:00, Wednesday, July 17, Room: Pitanga (Baobá 2)

Adoption of Algorithms from HV to SV



Our Contributions

- Systematic transferability study
- Confirming important claims from the original publications [13, 14]
- Comparing ISMC and DAR against Impact [11] and predicate abstraction [9], two interpolation-based approaches from SV
- Open-source implementations of ISMC and DAR in CPAchecker

Hypotheses from Original Publications

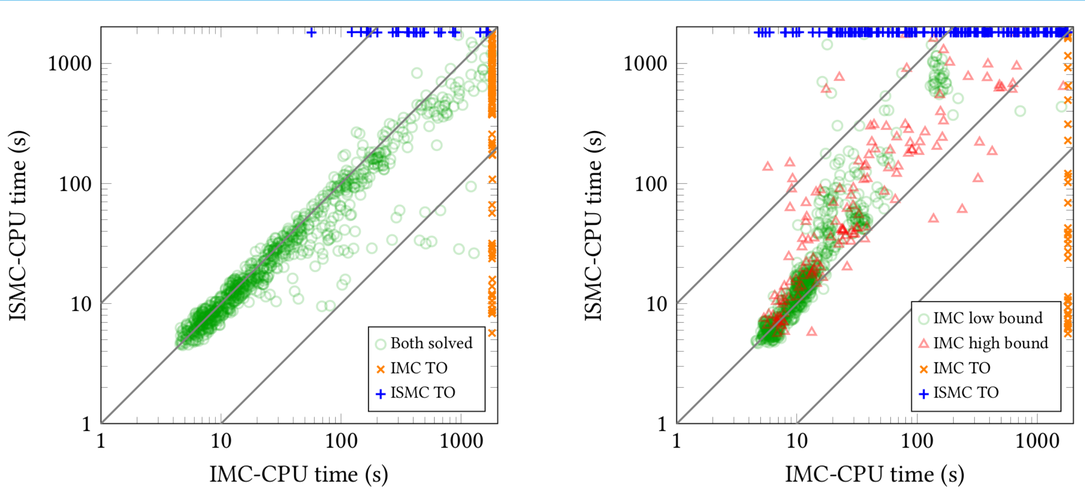
	Hypothesis from ISMC Paper [13]	Confirmed
H1.A	ISMC faster in finding bugs	✓
H1.B	ISMC faster in proving property if high unrolling bound	?
H1.C	ISMC overall faster	?

	Hypothesis from DAR Paper [14]	Confirmed
H2.A	DAR performs more local phases than global	✓
H2.B	DAR faster in proving property	?
H2.C	DAR computes more interpolants	✓
H2.D	DAR's runtime more sensitive to sizes of interpolants	?
H2.E	DAR overall faster than IMC	?

Experimental Setup

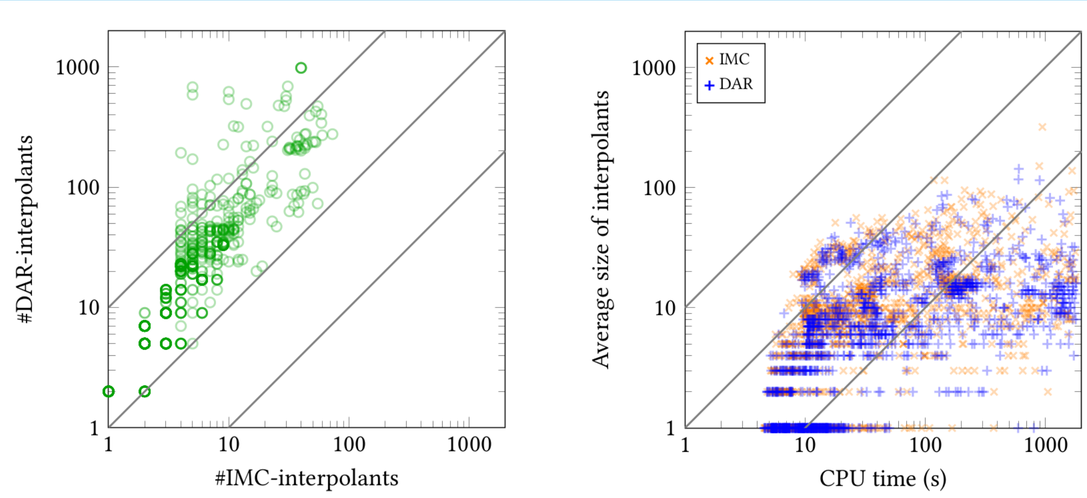
- All compared algorithms implemented in CPAchecker [2]
- Benchmark set: 8813 C programs from SV-COMP [1]
- Comparing to SV algorithms on 4790 programs with at most one loop
- Machines with 3.40 GHz CPU (Intel Xeon E3-1230 v5)
- Cores limit: 2; CPU time limit: 1800 s; memory limit: 15 GB

Evaluation of H1.A and H1.B



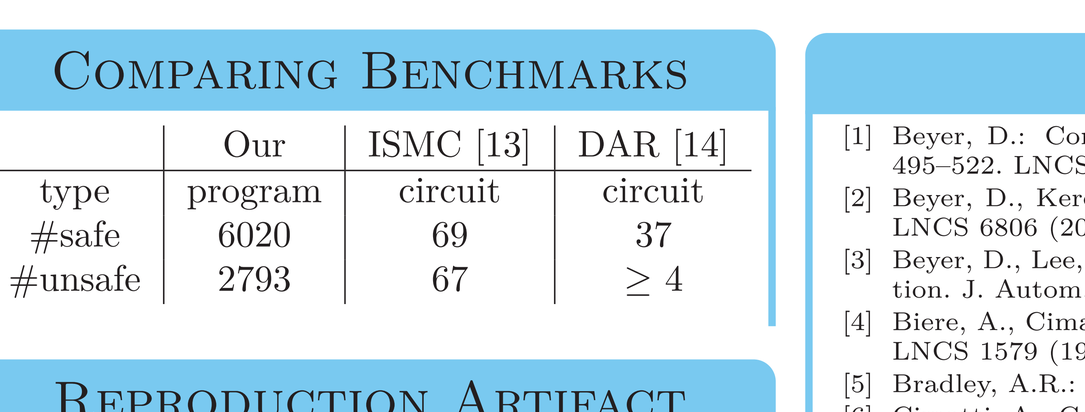
(a) Alarms **(b)** Proofs

Evaluation of H2.C and H2.D

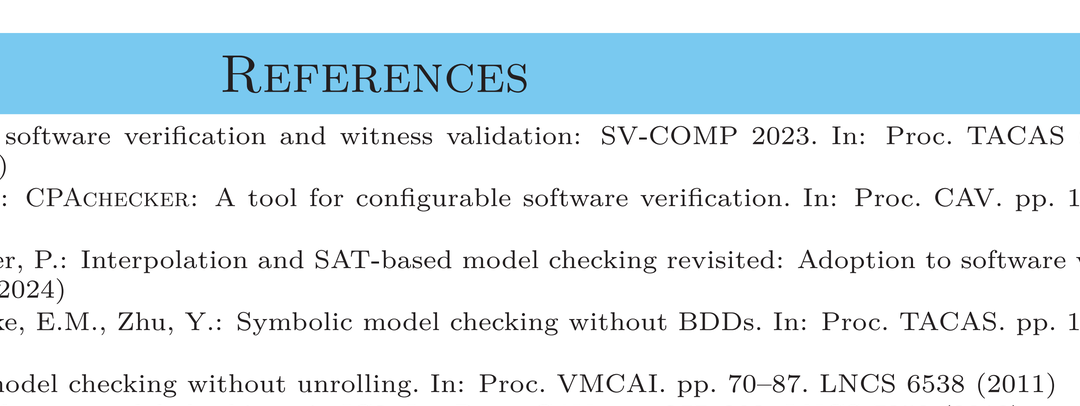


(a) Numbers of interpolants of DAR vs. IMC **(b)** Size of interpolants vs. CPU time

Evaluation of H1.C and H2.E



Comparison Against SV Algorithms



Comparing Benchmarks

	Our	ISMC [13]	DAR [14]
type	program	circuit	circuit
#safe	6020	69	37
#unsafe	2793	67	≥ 4

Reproduction Artifact

Artifact DOI: 10.5281/zenodo.11070973

References

[1] Beyer, D.: Competition on software verification and witness validation: SV-COMP 2023. In: Proc. TACAS (2). pp. 495–522. LNCS 13994 (2023)
[2] Beyer, D., Keremoglu, M.E.: CPAchecker: A tool for configurable software verification. In: Proc. CAV. pp. 184–190. LNCS 6806 (2011)
[3] Beyer, D., Lee, N.Z., Wendler, P.: Interpolation and SAT-based model checking revisited: Adoption to software verification. J. Autom. Reasoning (2024)
[4] Biere, A., Cimatti, A., Clarke, E.M., Zhu, Y.: Symbolic model checking without BDDs. In: Proc. TACAS. pp. 193–207. LNCS 1579 (1999)
[5] Bradley, A.R.: SAT-based model checking without unrolling. In: Proc. VMCAI. pp. 70–87. LNCS 6538 (2011)
[6] Cimatti, A., Griggio, A.: Software model checking via IC3. In: Proc. CAV. pp. 277–293. LNCS 7358 (2012)
[7] Clarke, E.M., Kröning, D., Lerda, F.: A tool for checking ANSI-C programs. In: Proc. TACAS. pp. 168–176. LNCS 2988 (2004)
[8] Donaldson, A.F., Haller, L., Kröning, D., Rümmer, P.: Software verification using k-induction. In: Proc. SAS. pp. 351–368. LNCS 6887 (2011)
[9] Henzinger, T.A., Jhala, R., Majumdar, R., McMillan, K.L.: Abstractions from proofs. In: Proc. POPL. pp. 232–244 (2004)
[10] McMillan, K.L.: Interpolation and SAT-based model checking. In: Proc. CAV. pp. 1–13. LNCS 2725 (2003)
[11] McMillan, K.L.: Lazy abstraction with interpolants. In: Proc. CAV. pp. 123–136. LNCS 4144 (2006)
[12] Sheeran, M., Singh, S., Stålmarck, G.: Checking safety properties using induction and a SAT-solver. In: Proc. FMCAD, pp. 127–144. LNCS 1954 (2000)
[13] Vizel, Y., Grumberg, O.: Interpolation-sequence based model checking. In: Proc. FMCAD. pp. 1–8 (2009)
[14] Vizel, Y., Grumberg, O., Shoham, S.: Intertwined forward-backward reachability analysis using interpolants. In: Proc. TACAS. pp. 308–323. LNCS 7795 (2013)