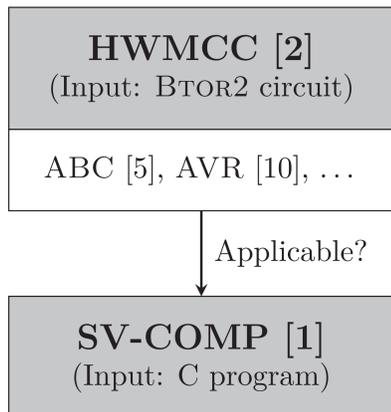
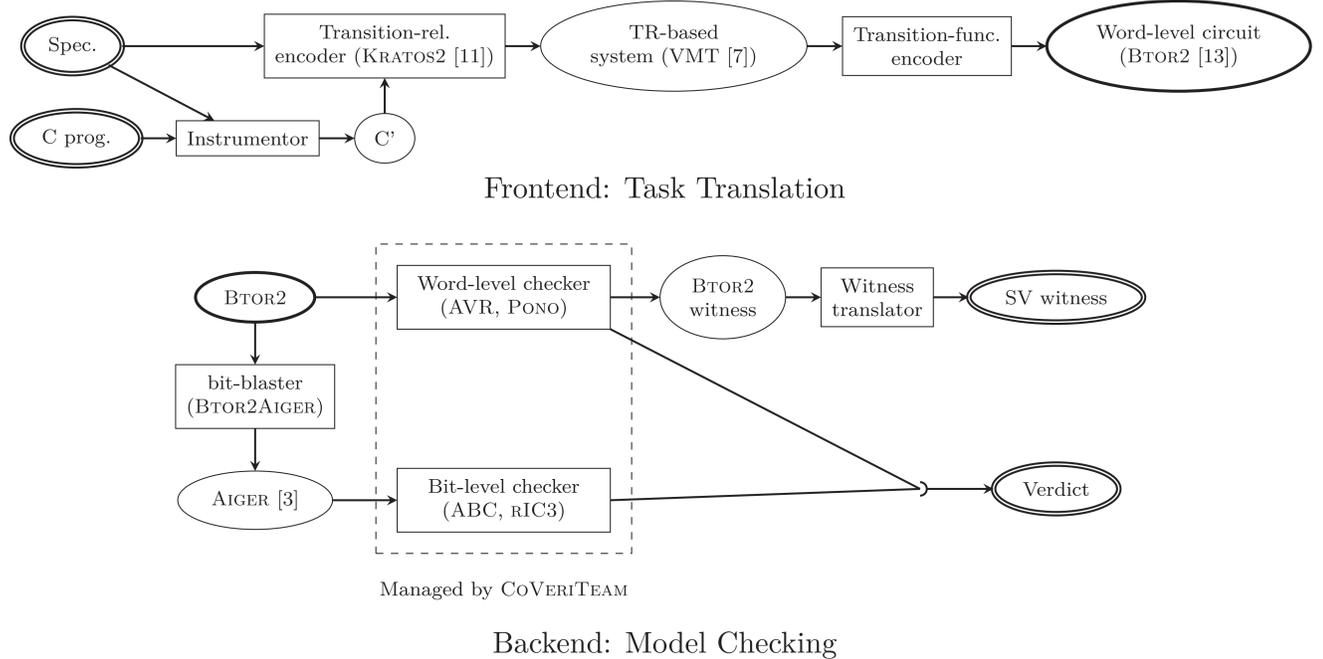


## Motivation



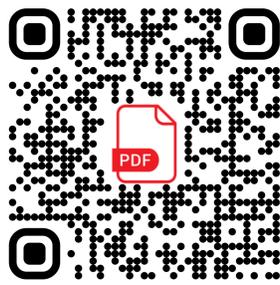
## Software Architecture



## Try CPV!

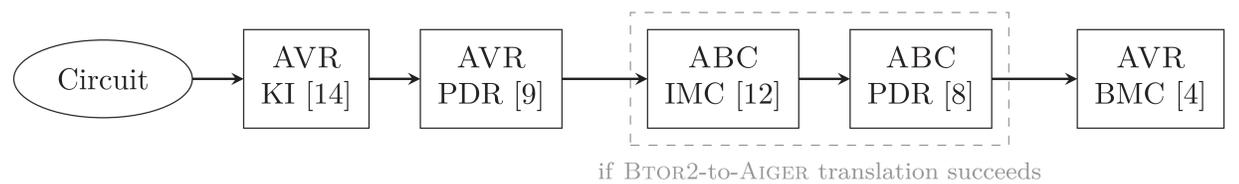


gitlab.com/sosy-lab/  
software/cpv



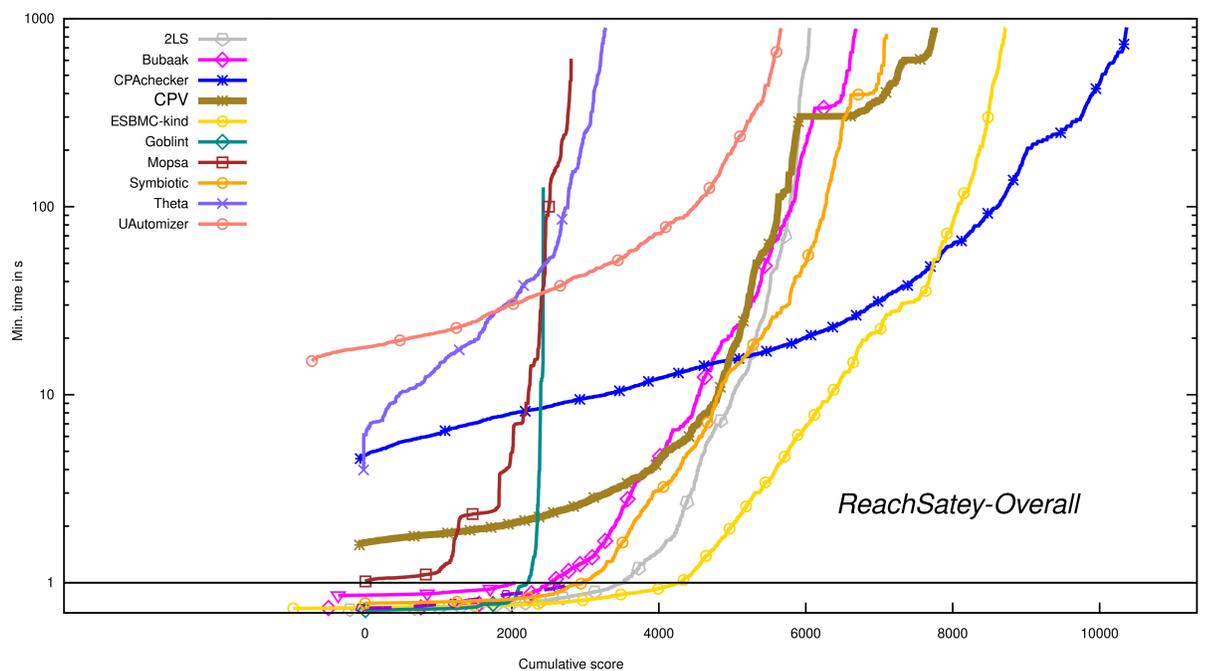
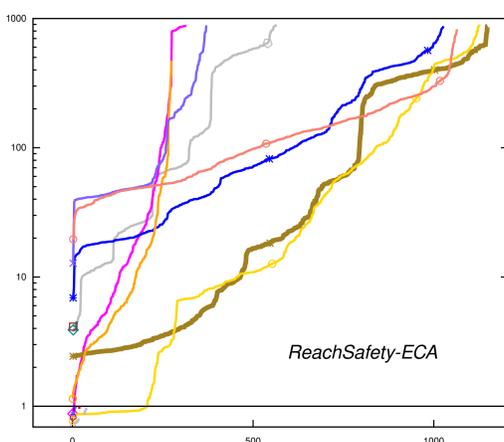
TACAS 2024 [6]

## Strategy for SV-COMP 2025



## Evaluation Results at SV-COMP 2025

- 🏆 3rd place in *ReachSafety-Overall*, including
- 🏆 1st in *ReachSafety-BitVectors* and *-ECA*
- 🏆 2nd in *ReachSafety-Hardware* and *-ProductLines*
- 🏆 3rd in *ReachSafety-Combinations* and *-Hardness*



## Summary

- Sequential circuits can serve as an intermediate representation for software verification.
- Offer different encoding options.
- Leverage powerful word- and bit-level hardware model checkers as backend.
- Perform competitively against established verifiers in SV-COMP.
- Ongoing work:
  - Support more verification properties (e.g., no-overflow and termination)
  - Export correctness witnesses
  - Apply circuit optimization to improve the performance of verification

## References

- [1] Beyer, D., Strejček, J.: Improvements in software verification and witness validation: SV-COMP 2025. In: Proc. TACAS (3). LNCS 15698 (2025)
- [2] Biere, A., Froylyks, N., Preiner, M.: Hardware model checking competition 2024. In: Proc. FMCAD. pp. 7–7 (2024)
- [3] Biere, A., Heljanko, K., Wieringa, S.: AIGER 1.9 and beyond. Tech. Rep. 11/2, Institute for Formal Models and Verification, Johannes Kepler University (2011)
- [4] Biere, A., Cimatti, A., Clarke, E.M., Strichman, O., Zhu, Y.: Bounded model checking. *Advances in Computers* **58**, 117–148 (2003)
- [5] Brayton, R., Mishchenko, A.: ABC: An academic industrial-strength verification tool. In: Proc. CAV. pp. 24–40. LNCS 6174 (2010)
- [6] Chien, P.C., Lee, N.Z.: CPV: A circuit-based program verifier (competition contribution). In: Proc. TACAS (3). pp. 365–370. LNCS 14572 (2024)
- [7] Cimatti, A., Griggio, A., Tonetta, S.: The VMT-LIB language and tools. In: Proc. SMT. CEUR Workshop Proceedings, vol. 3185, pp. 80–89 (2022)
- [8] Eén, N., Mishchenko, A., Brayton, R.K.: Efficient implementation of property directed reachability. In: Proc. FMCAD. pp. 125–134 (2011)
- [9] Goel, A., Sakallah, K.: Model checking of Verilog RTL using IC3 with syntax-guided abstraction. In: Proc. NFM. pp. 166–185 (2019)
- [10] Goel, A., Sakallah, K.: AVR: Abstractly verifying reachability. In: Proc. TACAS. pp. 413–422. LNCS 12078 (2020)
- [11] Griggio, A., Jonáš, M.: KRATOS2: An SMT-based model checker for imperative programs. In: Proc. CAV. pp. 423–436 (2023)
- [12] McMillan, K.L.: Interpolation and SAT-based model checking. In: Proc. CAV. pp. 1–13. LNCS 2725 (2003)
- [13] Niemetz, A., Preiner, M., Wolf, C., Biere, A.: BTOR2, BTORMC, and BOOLECTOR 3.0. In: Proc. CAV. pp. 587–595. LNCS 10981 (2018)
- [14] Sheeran, M., Singh, S., Stålmarck, G.: Checking safety properties using induction and a SAT-solver. In: Proc. FMCAD, pp. 127–144. LNCS 1954 (2000)