Proceedings of SAT Competition 2025: Solver and Benchmark Descriptions, Proceedings of SAT Competitions, Vienna, 2025.

CaDiCaL, Gimsatul, IsaSAT and Kissat Entering the SAT Competition 2025

Armin Biere*, Tobias Faller*, Mathias Fleury*, Nils Froleyks‡, Florian Pollitt*
* *University of Freiburg*, Germany ‡ *Johannes Kepler University*, Linz, Austria

In essence the same versions of Gimsatul, IsaSAT and Kissat were submitted to the SAT Competition 2025 as those submitted in 2024 [1]. Note, that Kissat was dominating the competition in the main track in 2024. However it lacks many features supported by CaDiCaL. For instance Kissat does not support incremental solving. Thus, instead of trying to improve Kissat, we put our efforts into porting missing features from Kissat to CaDiCaL. This new submitted version of CaDiCaL is available on the development branch on Github. Its code base increased roughly from 46 KLOC to 64 KLOC of C++, needed to support the following techniques ported from Kissat:

- *clausal congruence closure* [2]
- *bounded variable addition* [3], [4]
- *ticks based scheduling* [5]
- *revisited vivification* [6]
- *look-ahead lucky phases* [1]
- *clausal equivalence sweeping* [7]
- *semantic definition mining* [8]

The latter two techniques rely on the internal sub-solver Kitten (from Kissat), which adds another roughly 3 KLOC of C.

On the practical side a major challenge was to produce LRAT proofs with antecedents for all the advanced inprocessing we added, particularly for clausal congruence closure. This is in contrast to Kissat, which only supports easier to produce, more compact but much slower to check DRUP proofs.

On the theoretical side, even though not important for the SAT competition, we had to come up with a new clausal proof calculus [9]. It allows to add specific blocked clauses during incremental SAT solving, including some form of extended resolution, necessary to produce proofs when using bounded variable addition in an incremental setting. Also the API changed slightly as the user now needs to ask the solver for new available variables indices instead of using them freely.

Preliminary results suggest that this improves run-time of stand-alone solving substantially, allows to solve benchmarks which require bounded variable addition or congruence closures, but still falls behind Kissat, unless we also take proof checking into account. The combined time of proof production and checking for CaDiCaL is much smaller than for Kissat.

This work was supported in part by the state of Baden-Württemberg through bwHPC, the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG.

References

[1] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks, and F. Pollitt, “CaDiCaL, Gimsatul, IsaSAT and Kissat entering the SAT Competition 2024,” in *Proc. of SAT Competition 2024 – Solver, Benchmark and Proof Checker Descriptions*, ser. Department of Computer Science Report Series B, M. Heule, M. Iser, M. Järvisalo, and M. Suda, Eds., vol. B-2024-1. University of Helsinki, 2024, pp. 8–10.

[2] A. Biere, K. Fazekas, M. Fleury, and N. Froleyks, “Clausal congruence closure,” in *27th International Conference on Theory and Applications of Satisfiability Testing, SAT 2024, August 21-24, 2024, Pune, India*, ser. LIPIcs, S. Chakraborty and J. R. Jiang, Eds., vol. 305. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 6:1–6:25.

[3] N. Manthey, M. Heule, and A. Biere, “Automated reencoding of boolean formulas,” in *Hardware and Software: Verification and Testing - 8th International Haifa Verification Conference, HVC 2012, Haifa, Israel, November 6-8, 2012. Revised Selected Papers*, ser. Lecture Notes in Computer Science, A. Biere, A. Nahir, and T. E. J. Vos, Eds., vol. 7857. Springer, 2012, pp. 102–117.

[4] A. Haberlandt, H. Green, and M. J. H. Heule, “Effective auxiliary variables via structured reencoding,” in *26th International Conference on Theory and Applications of Satisfiability Testing, SAT 2023, July 4-8, 2023, Alghero, Italy*, ser. LIPIcs, M. Mahajan and F. Slivovsky, Eds., vol. 271. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 11:1–11:19.

[5] A. Biere, “CaDiCaL, Lingeling, Plingeling, Treengeling, YalSAT Entering the SAT Competition 2017,” in *Proc. of SAT Competition 2017 – Solver and Benchmark Descriptions*, ser. Department of Computer Science Series of Publications B, T. Balyo, M. J. H. Heule, and M. Järvisalo, Eds., vol. B-2017-1. University of Helsinki, 2017, pp. 14–15.

[6] F. Pollitt, M. Fleury, A. Biere, M. Heule, K. Sakallah, J. Chen, and Y. Fisseha, “Revisiting clause vivification,” in *Joint Proceedings of the 23rd International Workshop on Satisfiability Modulo Theories (SMT’25) and of the 16th Pragmatics of SAT International Workshop (POS’25), co-located with the 28th International Conference on Theory and Applications of Satisfiability Testing (SAT’23), Glasgow, UK, July, 11, 2025*, ser. CEUR Workshop Proceedings, A. N. Jochen Hoenicke, Mikoláš Janota and S. Tourret, Eds. CEUR-WS.org, 2025, to be published.

[7] A. Biere, K. Fazekas, M. Fleury, and N. Froleyks, “Clausal equivalence sweeping,” in *Proceedings 24th International Conference on Formal Methods in Computer-Aided Design (FMCAD’24)*, N. Narodytska and P. Rümmer, Eds. TU Wien Academic Press, 2024, pp. 236–241.

[8] M. Fleury and A. Biere, “Mining definitions in Kissat with Kittens,” *Formal Methods Syst. Des.*, vol. 60, no. 3, pp. 381–404, 2022.

[9] K. Fazekas, F. Pollitt, M. Fleury, and A. Biere, “Incremental inprocessing rules beyond resolution,” in *Joint Proceedings of the 23rd International Workshop on Satisfiability Modulo Theories (SMT’25) and of the 16th Pragmatics of SAT International Workshop (POS’25), co-located with the 28th International Conference on Theory and Applications of Satisfiability Testing (SAT’23), Glasgow, UK, July, 11, 2025*, ser. CEUR Workshop Proceedings, A. N. Jochen Hoenicke, Mikoláš Janota and S. Tourret, Eds. CEUR-WS.org, 2025, to be published.