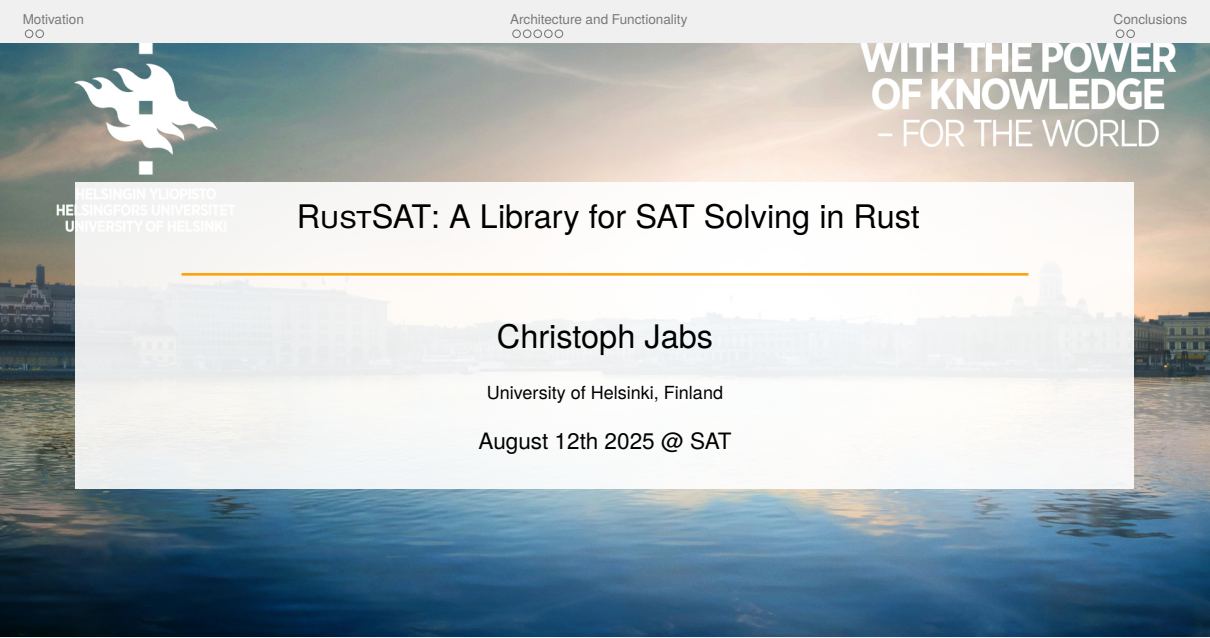


RustSAT: A Library for SAT Solving in Rust

Christoph Jabs

University of Helsinki, Finland

August 12th 2025 @ SAT

SAT Solving is Great

…but not always easy for non-experts

- SAT-based approaches are great for many applications
- Implementing SAT-based tools requires expert knowledge
- Reducing barriers of entry is important for increasing adoption of SAT technology
- Great examples
  - IPASIR: unifying solver interfaces
  - PySAT: making SAT accessible in Python
  - …

RustSAT: SAT solving in Rust

Contributions

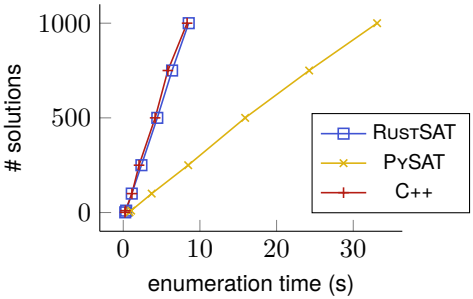
A library for easy and performant SAT solving in Rust

RustSAT: A Rust library for …

- working with SAT (and related) instances
- interfacing state-of-the-art SAT solvers
- encoding higher-level constraints to SAT

…with the aim of …

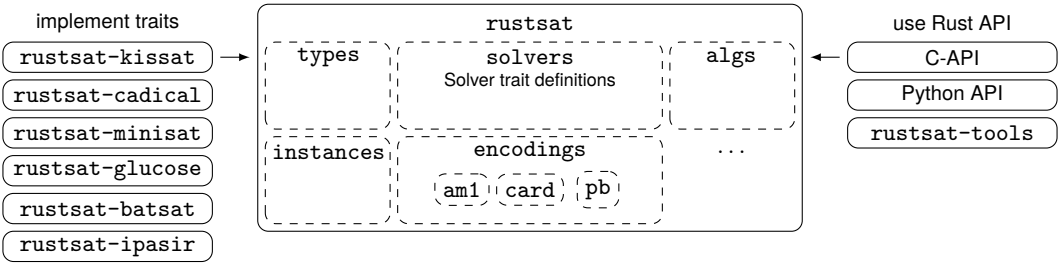
- a good user experience (automatically compiling/linking SAT solvers, good documentation, …)
- little to no performance overhead



`github.com/chrjabs/rustsat`

Architecture Overview

Core library, solver interfaces, external APIs



Working with Instances

Building, parsing, and converting

- Basic types (`Var`, `Lit`, `Clause`, ...)
- Instance types
  - `SatInstance`: Decision problem instance
  - `OptInstance`: Optimization problem instance
  - `Cnf`
- File parsing (DIMACS (W)CNF, OPB)
- Encoding to CNF
- ...

Unified Solver Interface

Plug & play SAT solvers thanks to Rust's type system

Functionality defined as `traits` (interfaces)

- `Solve`: one-shot solving
- `SolveIncremental`: *incremental* solving
- `PhaseLit`: user phases
- `LimitConflicts`, `LimitDecisions`, `LimitPropagations`
- ...

Included solvers

- Kissat (3.0.0 – 4.0.2) [Biere et al. 2024b]
- CaDiCaL (1.5.0 – 2.1.3) [Biere et al. 2024a]
- MiniSat (2.2.0) [Eén and Sörensson 2003]
- Glucose (4.2.1) [Audemard and Simon 2009]
- Batsat (0.6.0) [Cruanes and Hara n.d.]
- IPASIR [Balyo and Biere n.d.]
- Call solver binary

Higher-level Constraint Encodings

Optimized encodings, implemented in Rust, also available in a C and Python API

At-most-one constraints

- Pairwise [Prestwich 2021]
- Ladder [Gent and Nightingale 2004]
- Bitwise [Prestwich 2007]
- Commander [Klieber and Kwon 2007]
- Bimander [Nguyen and Mai 2015]

Cardinality constraints

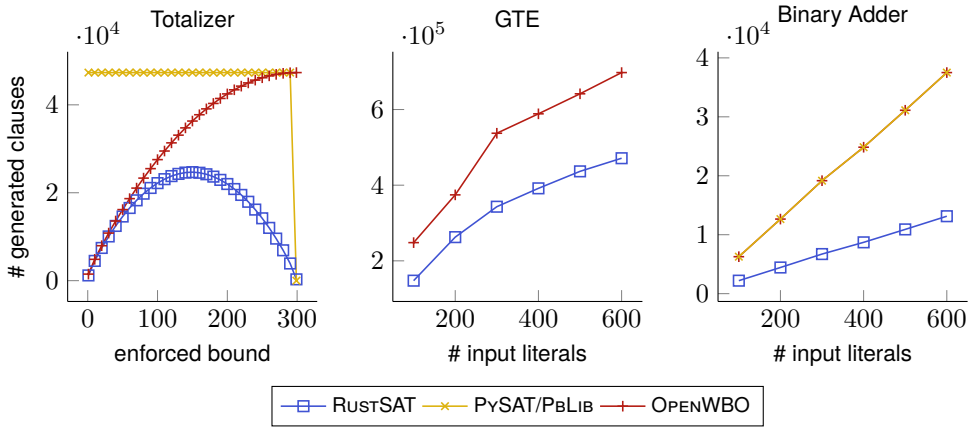
- Totalizer [Bailleux and Boufkhad 2003]

Pseudo-Boolean constraints

- Generalized totalizer [Joshi et al. 2015]
- Binary adder [Warners 1998]
- Dynamic polynomial watchdog [Paxian et al. 2018]

Cone of Influence Optimization

Removing “dangling” clauses from encodings



RustSAT in Use

Thanks to early adopters and their feedback

Rust projects

- Product configuration tool for automotive industry [Bruns 2024]
- Explaining pen and paper puzzles [Espasa et al. 2021]
- Scuttle multi-objective MaxSAT solver [Jabs n.d.]

Constraint encodings through C-API

- Loandra MaxSAT solver [Berg et al. 2024]
- MalloMax distributed MaxSAT solver [Schreiber et al. 2025]

RustSAT: A Library for SAT Solving in Rust

Summary and conclusions

Functionality

- Working with SAT instances
- Solver interfaces
- Higher-level constraint encodings

Design goals

- Polished user experience
- Little to no performance overhead

RustSAT repo, docs, paper, slides, and contact:

`christophjabs.info/sat25`

Bibliography I

Audemard, Gilles and Laurent Simon (2009). 'Predicting Learnt Clauses Quality in Modern SAT Solvers.'. In: *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11–17, 2009*. Ed. by Craig Boutilier, pp. 399–404. URL: http://ijcai.org/Proceedings/09/Papers/074.pdf.

Bailleux, Olivier and Yacine Boufkhad (2003). 'Efficient CNF Encoding of Boolean Cardinality Constraints.'. In: *Principles and Practice of Constraint Programming—CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, September 29 – October 3, 2003, Proceedings*. Ed. by Francesca Rossi. Vol. 2833. Lecture Notes in Computer Science. Springer, pp. 108–122. ISBN: 3-540-20202-1. DOI: 10.1007/978-3-540-45193-8_8.

Balyo, Tomas and Armin Biere (n.d.). *IPASIR: The Standard Interface for Incremental Satisfiability Solving*. https://github.com/biotomas/ipasir.

Berg, Jeremias et al. (2024). 'Loandra in the 2024 MaxSAT Evaluation'. In: *MaxSAT Evaluation 2024: Solver and Benchmark Descriptions*. Ed. by Jeremias Berg et al. Vol. B-2024-2. Department of Computer Science Report Series B. University of Helsinki, pp. 9–10.

Biere, Armin et al. (2024a). 'CaDiCaL 2.0.'. In: *Computer Aided Verification—36th International Conference, CAV 2024, Montreal, QC, Canada, July 24–27, 2024, Proceedings, Part I*. Ed. by Arie Gurfinkel and Vijay Ganesh. Vol. 14681. Lecture Notes in Computer Science. Springer, pp. 133–152. ISBN: 978-3-031-65627-9. DOI: 10.1007/978-3-031-65627-9_7.

Bibliography II

Biere, Armin et al. (2024b). 'CaDiCaL, Gimsatul, IsaSAT and Kissat Entering the SAT Competition 2024'. In: *Proceedings of SAT Competition 2024—Solver, Benchmark and Proof Checker Descriptions*. Ed. by Marijn Heule et al. Vol. B-2024-1. Department of Computer Science Report Series B. University of Helsinki, pp. 8–10.

Bruns, Noah Frederik (2024). 'Application of Multi-Objective MaxSAT-Solvers for Optimizing Highly Configurable Products'. MA thesis. Technische Universität Wien.

Cruanes, Simon and Masaki Hara (n.d.). *BatSat: A (parametrized) Rust SAT solver originally based on MiniSat*. https://github.com/c-cube/batsat.

Eén, Niklas and Niklas Sörensson (2003). 'An Extensible SAT-solver.'. In: *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5–8, 2003 Selected Revised Papers*. Ed. by Enrico Giunchiglia and Armando Tacchella. Vol. 2919. Lecture Notes in Computer Science. Springer, pp. 502–518. ISBN: 3-540-20851-8. DOI: 10.1007/978-3-540-24605-3_37.

Espasa, Joan et al. (2021). 'Using Small MUSes to Explain How to Solve Pen and Paper Puzzles.'. In: *Computing Research Repository* abs/2104.15040. URL: https://arxiv.org/abs/2104.15040.

Bibliography III

Gent, Ian P and Peter Nightingale (2004). 'A new Encoding of Alldifferent into SAT'. In: *International Workshop on Modelling and Reformulating Constraint Satisfaction*. Vol. 3, pp. 95–110.

Jabs, Christoph (n.d.). *Scuttle: A multi-objective MaxSAT solver*. https://bitbucket.org/coreo-group/scuttle.

Joshi, Saurabh et al. (2015). 'Generalized Totalizer Encoding for Pseudo-Boolean Constraints.'. In: *Principles and Practice of Constraint Programming—21st International Conference, CP 2015, Cork, Ireland, August 31 – September 4, 2015, Proceedings*. Ed. by Gilles Pesant. Vol. 9255. Lecture Notes in Computer Science. Springer, pp. 200–209. ISBN: 978-3-319-23218-8. DOI: 10.1007/978-3-319-23219-5_15.

Klieber, William and Gihwon Kwon (2007). 'Efficient CNF Encoding for Selecting 1 from N Objects'. In: *Workshop on Constraints in Formal Verification*.

Nguyen, Van-Hau and Son Thai Mai (2015). 'A New Method to Encode the At-Most-One Constraint into SAT.'. In: *Proceedings of the Sixth International Symposium on Information and Communication Technology, Hue City, Vietnam, December 3–4, 2015*. Ed. by Huynh Quyet Thang et al. ACM, pp. 46–53. ISBN: 978-1-4503-3843-1. DOI: 10.1145/2833258.2833293. URL: http://dl.acm.org/citation.cfm?id=2833258.

Bibliography IV

Paxian, Tobias et al. (2018). 'Dynamic Polynomial Watchdog Encoding for Solving Weighted MaxSAT.'. In: *Theory and Applications of Satisfiability Testing—SAT 2018—21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9–12, 2018, Proceedings*. Ed. by Olaf Beyersdorff and Christoph M. Wintersteiger. Vol. 10929. Lecture Notes in Computer Science. Springer, pp. 37–53. ISBN: 978-3-319-94144-8. DOI: 10.1007/978-3-319-94144-8_3.

Prestwich, Steven D. (2007). 'Finding Large Cliques using SAT Local Search'. In: *Trends in Constraint Programming*. Ed. by Barry O'Sullivan Frédéric Benhamou Narendra Jussien. John Wiley & Sons, Ltd. Chap. 15, pp. 269–274. ISBN: 9780470612309. DOI: https://doi.org/10.1002/9780470612309.ch15. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470612309.ch15.

— (2021). 'CNF Encodings.'. In: *Handbook of Satisfiability—Second Edition*. Ed. by Armin Biere et al. Vol. 336. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 75–100. ISBN: 978-1-64368-161-0. DOI: 10.3233/FAIA200985.

Schreiber, Dominik et al. (Aug. 2025). 'From Scalable SAT to MaxSAT: Massively Parallel Solution Improving Search'. In: *Eighteenth International Symposium on Combinatorial Search, SoCS 2025, Glasgow, UK, August 12–15, 2025*. Ed. by Maxim Likhachev et al. AAAI Press, pp. 127–135. DOI: https://doi.org/10.1609/socs.v18i1.35984.

Warners, Joost P. (1998). 'A Linear-Time Transformation of Linear Inequalities into Conjunctive Normal Form.'. In: *Information Processing Letters* 68, pp. 63–69. DOI: 10.1016/S0020-0190(98)00144-6.