

LearnSAT: A SAT Solver for Education

Mordechai (Moti) Ben-Ari¹

¹ Weizmann Institute of Science

DOI: [10.21105/joss.00639](https://doi.org/10.21105/joss.00639)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 25 February 2018

Published: 26 April 2018

Licence

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

Many problems can be solved by encoding them as formulas in propositional logic which are checked for satisfiability by a program called a SAT solver. If found, a satisfying assignment provides the answer to the problem. Donald Knuth (Knuth 2015) calls SAT solvers a *killer app* because diverse practical problems can be solved by a single, highly-optimized, program. Despite their importance, SAT solvers are described only in research papers. LearnSAT is designed for teaching and learning SAT solving by providing a concise, well-documented, implementation, fine-grained tracing (both textual and graphical) of the algorithms and a detailed tutorial.

LearnSAT implements the core algorithms of modern SAT solvers (Biere et al. 2009): the Davis-Putnam-Logemann-Loveland (DPLL) algorithm with conflict-driven clause learning (CDCL) and non-chronological backtracking (NCB). CDCL is implemented by backwards resolution from a conflict clause to a unique implication point (UIP) (Marques-Silva, Lynce, and Malik 2009). It is also possible to compute dominators in the implication graph. In addition, two heuristics for lookahead are implemented.

The user of LearnSAT can choose any subset of 24 display options in order to tailor the output to a specific learning context. The display options include elementary steps like decision assignments, unit propagations and identifying conflict clauses, as well as the advanced steps of CDCL: the resolution steps used to obtain a learned clause and the search for UIPs.

LearnSAT can generate two types of graphs that are rendered using the `dot` tool: trees showing the search through the assignments and implication graphs that display the process for learning clauses from conflicts.

LearnSAT is implemented in Prolog and was developed using `SWI-Prolog` which available for Windows, MacOSX and Linux, although almost all the source code is compatible with the standard.

Three documents are included in the project: * `learnsat-overview` An overview of LearnSAT. * `learnsat-ug` User's guide and software documentation. * `learnsat-tutorial` Tutorial on SAT solving using LearnSAT.

About 20 examples are included of encodings of problems that can be solved by LearnSAT.

References

Biere, A., M. Heule, H. Van Maaren, and T. Walsh, eds. 2009. *Handbook of Satisfiability*. IOS Press.

Knuth, D.E. 2015. *The Art of Computer Programming, Volume 4, Fascicle 6: Satisfiability*. Pearson.

Marques-Silva, J. P., I. Lynce, and S. Malik. 2009. “Conflict-Driven Clause Learning Sat Solvers.” In *Handbook of Satisfiability*, edited by A. Biere, M. Heule, H. Van Maaren, and T. Walsh, 131–53. IOS Press. <https://doi.org/10.3233/978-1-58603-929-5-131>.