

# CleanMaple

Benjamin Kaiser and Robert Clausecker  
Zuse Institute Berlin  
Berlin, Germany  
{kaiser,clausecker}@zib.de

Abstract—This document describes the SAT Solver CleanMaple, which is a refactored version of the SAT Competition 2018 winner `Maple_LCM_Dist_ChronoBT` [1].

Index Terms—SAT, refactoring, CDCL

## I. Overview

The complex nature of the CDCL algorithm and the necessity of high performance implementations encourages a tight coupling of most subroutines and data structures in the source code. However the basic ideas on which CDCL algorithms are based are simple when contrasted to their implementation in `Maple_LCM_Dist_ChronoBT`. Most parts of the solver `Maple_LCM_Dist_ChronoBT` are included in one huge monolithic class and many of its methods are themselves massive, having more than 50, 100 or even 150 lines of dense code, resulting in a single source file for this class of almost 2000 lines of code. This design choice as well as most code of the solver can be traced back to the solver `Minisat` [2] [3] from which `Maple_LCM_Dist_ChronoBT` evolved over a time span of more than ten years with ideas and contributions from many different authors, most notably by the authors of [4], [5], [6], [7], [8] and [9]. The rather complicated code base leads to a steep learning curve for researchers that wish to develop SAT Solvers based on this state-of-the-art solver.

## II. Description

In CleanMaple the two main subroutines

- Unit Propagation and
- the heuristic-based Branching,

and the three main data structures

- the clause database containing all original and learned clauses,
- the variable database containing the three-valued truth-value with respect to the current assignment and the polarities of all variables and
- the implication graph, i.e. the trail, used for fast conflict analysis

have been decoupled from the actual class. This leads to a solver that is much easier to understand. Furthermore, due to the refactoring the size of the binary of the solver was reduced significantly.

## Acknowledgment

We want to express our gratitude towards the organizers of the SAT Competition 2021 for making such an event possible. Additionally we like to thank Florian Schintke for his support and the IT and Data Services members of the Zuse Institute Berlin for providing the infrastructure and their fast help. Also we like to thank the authors of `Maple_LCM_Dist_ChronoBT` and everyone else contributing to this solver.

## References

- [1] Vadim Ryvchin and Alexander Nadel, “`Maple_LCM_Dist_ChronoBT`: Featuring Chronological Backtracking” in Proceedings of SAT Competition, 2018 p.29
- [2] Eén, Niklas and Sörensson, Niklas, “An Extensible SAT-solver” in Theory and Applications of Satisfiability Testing, 2004 p.502–518
- [3] Sörensson, Niklas and Eén, Niklas, “Minisat v1.13-a SAT solver with conflict-clause minimization” in International Conference on Theory and Applications of Satisfiability Testing, 2005
- [4] Audemard, Gilles and Simon, Laurent, “Predicting learnt clauses quality in modern SAT solvers” in Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009 p.399–404
- [5] Chanseok Oh, “COMiniSatPS the Chandrasekhar Limit and GHackCOMSPS” in Proceedings of SAT Competition 2016, p.29–30
- [6] Liang, Jia Hui and Ganesh, Vijay and Poupart, Pascal and Czarnecki, Krzysztof, “Learning Rate Based Branching Heuristic for SAT Solvers” in Theory and Applications of Satisfiability Testing – SAT 2016, 2016, p.123–140
- [7] Mao Luo and Chu-Min Li and Fan Xiao and Filip Manyà and Zhipeng Lü, “An Effective Learnt Clause Minimization Approach for CDCL SAT Solvers” in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, p.703–711
- [8] F. Xiao, M. Luo, C.-M. Li, F. Manyà, and Z. Lü, “MapleLRB\_LCM, Maple\_LCM, Maple\_LCM\_Dist, MapleLRB\_LCMocRestart and Glucose-3.0+width in SAT Competition 2017,” in Proceedings of SAT Competition, 2017 p.22
- [9] Nadel, Alexander and Ryvchin, Vadim, “Chronological Backtracking” in Theory and Applications of Satisfiability Testing – SAT 2018, 2018 p.111–121