CleanMaple_PriPro, CaDiCaL_PriPro and CaDiCaL_PriPro_no_bin

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

Abstract—Our experimental results suggest that some methods of rearranging the order in which clauses are propagated increase the performance in CDCL-solvers. CleanMaple_PriPro, CaDiCaL_PriPro and CaDiCaL_PriPro_no_bin are alterations of state-of-the-art SAT-solvers in which a novel approach of propagating some clauses with a severe priority increases their performance.

Index Terms—SAT, CDCL, propagation order

I. Priority Propagation

In all three solvers a second two-watch-literal-scheme of locally watched clauses is introduced. Newly learned conflict clauses are not registered to the standard two-watch-literal-scheme, but instead locally watched. Similarly, during conflict analysis each conflicting clause with an LBD of less than 7 not yet locally watched is deregistered from the standard two-watch-literal-scheme and instead registered to be locally watched. During propagation at each level the implications from all locally watched clauses at all levels are computed similarly to how binary clauses are propagated first in the SAT competition 2018 winner Maple_LCM_Dist_ChronoBT [1]. Every 10k conflicts all locally watched clauses are downgraded, i. e. deregistered from the second two-watch-literal-scheme and re-registered in the standard two-watch-literal-scheme.

II. Description of the solvers

The solver CleanMaple_PriPro is based on CleanMaple [2], which itself is based on Maple_LCM_Dist_ChronoBT. The solvers CaDiCaL_PriPro and CaDiCaL_PriPro_no_bin are based on CaDiCal [3]. In CleanMaple_PriPro binary clauses are never locally watched, but instead propagated immediately after locally watched clauses. This is due to the fact that binary clauses are watched in a separate watch-list in Maple_LCM_Dist_ChronoBT and have already been propagated with increased priority before. The solvers CaDiCaL_PriPro and CaDiCaL_PriPro_no_bin differ only by the fact whether binary clauses are considered for being locally watched or not. In all three solvers some in-processing steps needed to be removed or slightly altered, or enforce an early downgrading of all locally watched clauses.

Acknowledgment

We want to express our gratitude towards the organizers of the SAT Competition 2021 for making such an event possible. Additionally we would 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 would like to thank the authors of Maple_LCM_Dist_ChronoBT and everyone else contributing to this solver. In particular, we would like to thank the authors of [4] who discuss a different alteration of the propagation order and from whom we borrowed the idea of restricting our approach to clauses with small LBD. Benjamin Kaiser thanks Marc Hartung for introducing him to this wonderful subject of SAT-Solving and being a marvelous mentor during the past 18 months.

References

[1] Vadim Ryvchin and Alexander Nadel, “Maple_LCM_Dist_ChronoBT: Featuring Chronological Backtracking” in Proceedings of SAT Competition, 2018 p.29

[2] Benjamin Kaiser and Robert Clausecker, “CleanMaple” in Proceedings of SAT Competition, 2021 if accepted for submission

[3] Armin Biere, Katalin Fazekas, Mathias Fleury, Maximilian Heisinger and Johannes Kepler, “CADICAL, KISSAT, PARACOOBA, PLINGELING and TREENGELING Entering the SAT Competition 2020” in Proceedings of SAT Competition, 2020 p.50

[4] Jingchao Chen “Core First Unit Propagation” in arXiv, cs.LO, 2019, 1907.01192