MergeSAT 3.0

Norbert Manthey
nmanthey@conp-solutions.com
Dresden, Germany

***Abstract*—MERGESAT mission is to simplify keeping up with SAT development from a SAT user perspective. MERGESAT supports incremental solving, as well as being used as a library with various interfaces. MERGESAT is setup to simplify merging solver contributions into one solver. This setup should allow collaboration among solver developers more easily. The sequential SAT solver MERGESAT is a fork of the 2018 competition winner. The tool adds known as well as novel implementation and search improvements, typically based on solver contributions of previous annual competitions.**

I. THE SEQUENTIAL SOLVER MERGESAT

Most SAT solvers change over time, or their maintenance is dropped. From a user perspective, once choosing a SAT backend mostly results in fixing the performance of the selected SAT backend. Migrating between solvers is challenging. Furthermore, more recent successful solvers in SAT competitions do not support being used as a library, not incremental SAT solving. MERGESAT bridges this gap: the solver implements the broadly used MINISAT 2.2 interface, supports incremental solving, but also comes with the most recent solver improvements that have been developed in 2020.

MERGESAT is based on the SAT competition winner of 2018, MAPLE_LCM_DIST_CHRONOBT [7]. Based on recent SAT research, several known techniques as well as novel ideas has been integrated. To make continuing the long list of work that influenced MERGESAT simpler, MERGESAT uses git to combine changes from different solvers. Furthermore, MERGESAT comes with continuous integration to simplify extending the solver further. Starting in 2020, code style was enforced during CI as well, allowing to understand modifications better. MERGESAT has been tested as incremental SAT backend of several tools already. Furthermore, MERGESAT implements the SAT interface for the parallel HORDESAT solver [1], and can hence be used as a portfolio-parallel SAT solver with clause sharing as backend in HORDESAT.

A. *Extensions since 2020*

MERGESAT participated in SAT competition 2020 [5]. This section only details the differences since this version. All extensions can be disabled via the command line interface. Furthermore, the parameters to the solver can be specified via an environment variable. This setup allows to tune the solver even if it is used as a library in the backend.

The following improvements have been integrated into MERGESAT since the 2020 submissions:

- parsing multiple compression formats
- trail saving [3]
- use CCNR to participate in solving [8], but initialize the SLS engine lazily
- use rephasing as done in [8]
- allow to remove more clauses, as proposed in [4], but with a back-off strategy
- support printing the PCS format for configuration

Furthermore, formula simplification, as well as the CCNR engine are disabled in case too large formulas are used. When combining several extensions to the search also uncovered corner cases in the original implementation taken from the original tools. These corner cases have been addressed in MERGESAT to result in a safe search again. A full description of the solver is [6].

B. *Continuous Testing*

The submitted version of MERGESAT compiles on Linux and Mac OS. GitHub allows to use continuous testing, which essentially build MERGESAT, and tests basic functionality: i) producing unsatisfiability proofs, ii) building the starexec package and producing proofs and iii) solving via the IPASIR interface. All these steps are executed by executing the script “tools/ci.sh” from the repository, and the script can be used as a template to derive similar functionality.

C. *Availability*

The source of the solver is publicly available under the MIT license at https://github.com/conp-solutions/mergesat. The submitted starexec package can be reproduced by running “./scripts/make-starexec.sh” on this commit.

D. *Submitted* MERGESAT *Configurations*

The git version “v3.0-13-g76cb34f” has been submitted to the competition. The solver has been submitted to all sequential tracks, including the incremental track. For the main and agile tracks, a configuration *unsat* has been submitted as well. This configuration disabled the CCNR engine as well as rephasing, as experiments showed that these two modification – at least in their current form – degrade the performance on unsatisfiable formulas. Similarly, the configuration *nosimp* disables formula simplification. Comparing the results of the default configuration and *nosimp* should allow to motivate implementing formula simplfication lazily, i.e. eventually starting search without simplification as done in other solvers already [2].

II. THE PARALLEL SOLVER MERGE-HORDESAT

HORDESAT supports multi-threaded, as well as distributed solving with various SAT backends. Learned clauses can be shared among the solver instances. MERGE-HORDESAT extends HORDESAT with integrating MERGESAT. The following modifications have been applied on top:

- share learned clauses lazily, and not interrupting SAT engines, as that misleads the original search below sequential performance
- automatically choosing to run on half the available CPU cores, to allow simple parallel use
- support to print the solver model in the multi-threaded case, to get a full parallel solver
- use MERGESAT's CNF parser that supports compressed CNF files, for more flexible use
- support to build the solver with a single "make" invocation, to simplify usage
- add MERGESAT via *git submodules*, to allow stable version tracking when benchmarking and distributing
- remove all other SAT backends, as MERGESAT based additions have not been provided

Interestingly, exchanging the parser of HORDESAT resulted in a much faster file reading process. To make the solver prepared for future changes, and simple continuous testing setup has been created as well.

A. Availability

The source of the solver is publicly available under the MIT license at https://github.com/conp-solutions/hordesat.

B. Submitted MERGE-HORDESAT *Configurations*

The wrapper package to allow to use MERGE-HORDESAT in AWS (as done in the competition), can be found at: https://github.com/conp-solutions/hordesat-aws. The same version of the tool has been submitted to both the parallel as well as the cloud track. The solvers HORDESAT, as well as its backend MERGESAT, are integrated into the code base via *git submodules*, to allow stable version tracking. The version of MERGE-HORDESAT is "v1.0". MERGESAT is used in version "v3.0-13-g76cb34f", as submitted to the sequential tracks.

ACKNOWLEDGMENT

The author would like to thank the developers of all predecessors of MERGESAT, and all the authors who contributed the modifications that have been integrated. Furthermore, without the work of the authors of HORDESAT, creating MERGE-HORDESAT would not have happened.

REFERENCES

[1] T. Balyo, P. Sanders, and C. Sinz, "Hordesat: A massively parallel portfolio sat solver," in *Theory and Applications of Satisfiability Testing – SAT 2015*, M. Heule and S. Weaver, Eds. Cham: Springer International Publishing, 2015, pp. 156–172.

[2] A. Biere, "PrecoSAT system description," http://fmv.jku.at/precosat/preicosat-sc09.pdf, 2009.

[3] R. Hickey and F. Bacchus, "Trail saving on backtrack," in *Theory and Applications of Satisfiability Testing - SAT 2020 - 23rd International Conference, Alghero, Italy, July 3-10, 2020, Proceedings*, ser. Lecture Notes in Computer Science, L. Pulina and M. Seidl, Eds., vol. 12178. Springer, 2020, pp. 46–61.

[4] S. Kochemazov, "Improving implementation of SAT competitions 2017-2019 winners," in *Theory and Applications of Satisfiability Testing - SAT 2020 - 23rd International Conference, Alghero, Italy, July 3-10, 2020, Proceedings*, ser. Lecture Notes in Computer Science, L. Pulina and M. Seidl, Eds., vol. 12178. Springer, 2020, pp. 139–148.

[5] N. Manthey, "MergeSAT," in *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, ser. Department of Computer Science Report Series B, T. Balyo, N. Froleyks, M. Heule, M. Iser, M. Järvisalo, and M. Suda, Eds., vol. B-2020-1. University of Helsinki, 2020, p. 40.

[6] ——, "The MergeSAT Solver," in *Theory and Applications of Satisfiability Testing - SAT 2021*, 2021, submitted.

[7] V. Ryvchin and A. Nadel, "Maple_LCM_Dist_ChronoBT: Featuring Chronological Backtracking," in *Proceedings of SAT Competition 2018*, 2018. [Online]. Available: http://hdl.handle.net/10138/237063

[8] X. Zhang and S. Cai, "Relaxed backtracking with rephasing," in *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, ser. Department of Computer Science Report Series B, T. Balyo, N. Froleyks, M. Heule, M. Iser, M. Järvisalo, and M. Suda, Eds., vol. B-2020-1. University of Helsinki, 2020, pp. 15–15.