# DPS-Kissat

Hidetomo Nabeshima Tsubasa Fukiage Yuto Obitsu Xiao-Nan Lu
*University of Yamanashi*
Yamanashi, JAPAN
{nabesima,xnlu}@yamanashi.ac.jp

Katsumi Inoue
*National Institute of Informatics*
Tokyo, JAPAN
inoue@nii.ac.jp

***Abstract*—DPS is a framework for easily constructing efficient deterministic parallel SAT solvers, providing the delayed clause exchange technique introduced in ManyGlucose. We applied DPS to Kissat to construct a portfolio parallel SAT solver DPS-Kissat.**

## I. Introduction

DPS is a framework for easily implementing deterministic portfolio parallel SAT solvers for shared memory multicore environment, that guarantee reproducible behavior. Reproducibility means that the execution result (the running time and a found model if satisfiable ) does not change across runs. DPS is a successor to the deterministic parallel SAT solver ManyGlucose [1], from which it extracts and generalizes the mechanisms necessary to achieve reproducible behavior. We applied DPS to Kissat [2], one of the state-of-the-art sequential SAT solvers, to construct a portfolio parallel SAT solver DPS-Kissat.

## II. Delayed Clause Exchange

In parallel SAT solvers, reproducibility is lost when learnt clauses are exchanged asynchronously. Synchronous clause exchange ensures reproducibile behavior, but increases latency. The delayed clause exchange introduced in ManyGlucose allows a certain delay in the timing of clause exchanges, thereby absorbing fluctuations in the exchange interval and can reduce reducing the waiting time. However, implementing delayed clause exchange requires expert knowledge of concurrent programming, so introducing it into existing sequential SAT solvers is a time-consuming task. We have extracted the delayed clause exchange method from ManyGlucose and developed a framework DPS with a generic interface to facilitate its integration into existing sequential solvers.

DPS-Kissat is a deterministic parallel SAT solver that applies the delayed clause exchange provided by our framework to Kissat.

## III. Portfolio Strategy

The diversity strategy of DPS-Kissat consists of the following three elements:

1) random variable selection until the first conflict occurs except for the first thread. The random seeds use different values for each thread.
2) 24 different search strategy settings shown in the portfolio parallel SAT solver PaKis [3].
3) disabled elimination in half of threads.

The first strategy was introduced in ManySAT 2.0 [4], the first deterministic parallel SAT solver. Clause exchange in non-deterministic parallel SAT solvers is one of the causes of search diversity due to its asynchronous nature, but this is not expected in deterministic solvers, so strategies such as random decision are necessary to ensure diversity. PaKis executes 24 Kissat processes with different strategies in parallel without clause exchange, and has won the parallel SAT track in the SAT 2021 competition. The last strategy was introduced because there were some instances where a lot of time was spent on in-processing.

## IV. Implementation

DPS-Kissat parallelizes Kissat-SC2021 [5], which required about 400 lines of modification to Kissat and about 400 lines for the wrapper class to incorporate Kissat into DPS. The version submitted to SAT Competition 2022 launches 32 threads. The results are guaranteed to be reproducible. DPS supports non-deterministic mode, which is also entered in the competition as NPS-Kissat.

## Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers JP20H05963, JP20K11934. In this research work we used the supercomputer of ACCMS, Kyoto University.

## References

[1] H. Nabeshima and K. Inoue, "Reproducible efficient parallel SAT solving," in *Proceedings of the 23rd International Conference on Theory and Applications of Satisfiability Testing (SAT 2020), LNCS 12178*, 2020, pp. 123–138.
[2] A. Biere, K. Fazekas, M. Fleury, and M. Heisinger, "CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT competition 2020," http://hdl.handle.net/10138/318450, 2020, SAT Competition 2020 Solver Description.
[3] R. K. Tchinda and C. T. Djamegni, "hKis, hCaD, PaKis and PaInleSS_ExMapleLCMDistChronoBT in the SC21," http://hdl.handle.net/10138/333647, 2021, SAT Competition 2021 Solver Description.
[4] Y. Hamadi, S. Jabbour, C. Piette, and L. Sais, "Deterministic parallel DPLL," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 7, no. 4, pp. 127–132, 2011.
[5] A. Biere, M. Fleury, and M. Heisinger, "CaDiCaL, Kissat, Paracooba entering the SAT competition 2021," http://hdl.handle.net/10138/333647, 2021, SAT Competition 2021 Solver Description.