STP-Parti-Bitwuzla at SMT-COMP 2025

Mengyu Zhao[1,2], Zhenghang Xu[1] Jinkun Lin[3], and Shaowei Cai[1,2]

[1] Key Laboratory of System Software (Chinese Academy of Sciences) and
State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing, China
{zhaomy, caisw}@ios.ac.cn, zhenghangxu97@gmail.com
[2] University of Chinese Academy of Sciences, Beijing, China
[3] SeedMath Technology Limited
linjk@seedmath.com

1 Introduction

STP-Parti-Bitwuzla is a wrapper SMT solver based on STP [1] (commit SHA 0510509) and Bitwuzla [2] (v0.8.0). It participates in the Parallel Track of QF_BV logic. For SMT-COMP 2025, you can find the solver, experimental scripts, and Docker files that we have prepared at GitHub-STP-Parti-Bitwuzla-at-SMT-COMP-2025.

STP-Parti-Bitwuzla comprises three primary components: master, partitioner, and base solvers.

1. The master is implemented by Python for task management and scheduling in distributed solving.
2. We have customized STP to function as our partitioner, aiding in the partitioning of sub-problems.
3. Our tool does not restrict the choice of the base solver used in solving subtasks; it simply requires the executable binary file. For the SMT-COMP 2025, we have selected Bitwuzla as our base solver.

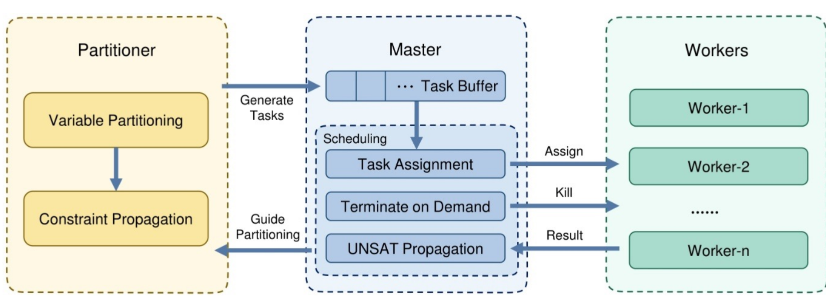

Fig. 1. Our dynamic parallel framework.

STP-Parti-Bitwuzla is the practical implementation of our innovative concept of variable-level partitioning, which is applied to the Bit-Vectors theory.

This technique was introduced for the first time in our recently published paper at CAV 2024, titled "Distributed SMT Solving Based on Dynamic Variable-level Partitioning" [3]. In this paper, we investigate variable-level partitioning for arithmetic theories with our proposed tool, AriParti, which is available on GitHub-AriParti. STP-Parti-Bitwuzla represents expanding and exploring the variable-level partitioning concept in the Bit-Vectors (BV) theory.

2 Features

Compared to STP and Bitwuzla, STP-Parti-Bitwuzla innovates in the following aspects:

Dynamic distributed framework. Our dynamic distributed framework is built upon a combination of three types of thread. This architecture allows for versatile configurations where individual workers can be assigned different SMT solvers or parameter settings. This heterogeneity leverages the complementary performance of various solvers to enhance the overall efficiency of parallel solving.

Integration of term-level and variable-level partitioning. The core of our variable-level partitioning strategy is to select a specific variable during the solving process and split its feasible domain. This action decomposes the original formula into two sub-formulas with more restrictive constraints. A key advantage of this method is its robustness, as it remains effective regardless of the Boolean structure of the formula. Furthermore, it works in synergy with term-level partitioning to produce a more diverse and valuable set of subtasks. Our proposed variable-level partitioning permits robust, comprehensive partitioning. Regardless of the Boolean structure of any given instance, our partitioning algorithm can keep partitioning to the last moment of the solving process.

Enhanced preprocessing and constraint propagation for the BV theory. Beyond traditional bit-vector simplification techniques, we have tightly integrated constraint propagation into our partitioning framework. Following each variable-level split, both Boolean and Interval Constraint Propagation are immediately applied to the resultant sub-formulas. This dual propagation approach serves to simplify the formulas and narrow the value ranges of their variables.

A load balancing mechanism for enhanced parallel solving efficiency. Our load balancing mechanism employs several strategies to maximize core utilization. First, the Partitioner maintains a task buffer to ensure that any idle core is immediately dispatched a new sub-formula. To handle bottlenecks, the Master can dynamically terminate stalled tasks, which are then returned to the Partitioner to be split into simpler sub-problems. Finally, drawing inspiration from work-stealing, the framework dynamically adjusts the partition granularity. This allows the Partitioner to focus on breaking down more complex tasks, thereby ensuring that the computational load is distributed equitably across all cores.

References

1. Ganesh, V., Dill, D.L.: A decision procedure for bit-vectors and arrays. In: Damm, W., Hermanns, H. (eds.) Computer Aided Verification. pp. 519–531. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
2. Niemetz, A., Preiner, M.: Bitwuzla. In: Enea, C., Lal, A. (eds.) Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13965, pp. 3–17. Springer (2023). https://doi.org/10.1007/978-3-031-37703-7_1, https://doi.org/10.1007/978-3-031-37703-7_1
3. Zhao, M., Cai, S., Qian, Y.: Distributed smt solving based on dynamic variable-level partitioning. In: CAV 2024, Montreal, Canada, July 24-27, 2024, Proceedings. Lecture Notes in Computer Science, Springer (2024), (to appear)