Z3-Parti-Z3++ at SMT-COMP 2024

Mengyu Zhao[1,2] 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
[2] University of Chinese Academy of Sciences, Beijing, China

1 Introduction

Z3-Parti-Z3++ is a derived SMT solver from Z3 [3] and participates in the Parallel Track and Cloud Track of the Arithmetic theories, namely QF_RDL, QF_IDL, QF_LRA, QF_LIA, QF_NRA, and QF_NIA logics. For SMT-COMP 2024, you can find the solver, experimental scripts, and Docker files we have prepared at GitHub-Z3-Parti-Z3++-at-SMT-COMP-2024.

Z3-Parti-Z3++ comprises the following three primary components: The master is implemented by Python for task management and scheduling in distributed solving. The partitioner is derived from Z3 (v4.12.1). As for base solvers, we apply our method to Z3++ [1], which participates in SMT-COMP 2023. Z3++ is also developed by our team and derived from Z3, and further information about Z3++ can be found at GitHub-Z3++.

Further insights into our tool can be found in our paper, "Distributed SMT Solving Based on Dynamic Variable-level Partitioning" [5]. We have provided our solver, evaluation scripts, and related experimental results in GitHub-AriParti.

2 Features

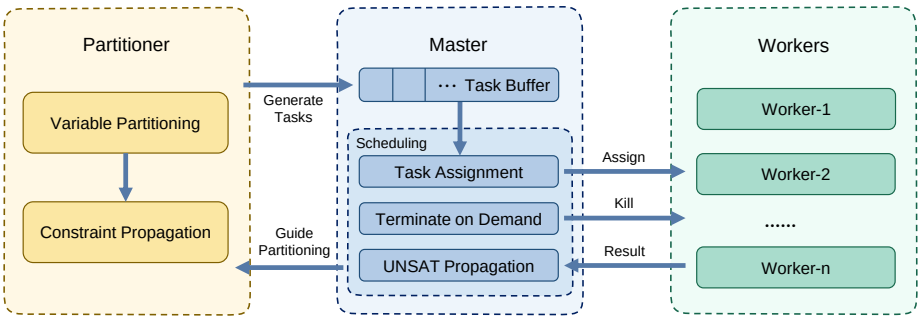

Fig. 1. Our dynamic parallel framework.

Dynamic parallel framework. We propose a dynamic parallel framework based on arithmetic variable-level partitioning. This framework ensures full utilization of computing resources, preventing idle core resources from lacking executable tasks. The dynamic parallel framework provides flexibility for parallel trees to grow. Thus, it can easily collaborate with other partitioning strategies — any sub-problem yielded previously by pre-partitioning strategies can be partitioned further.

Algorithm 1: Arithmetic Variable-level Partitioning

```
1  φ ← choose a leaf node from the partition tree
2  x ← choose a partitioning variable for the node φ
3  {φ_l, φ_r} ← perform interval partitioning on variable x within φ
4  {R_l, φ̂_l} ← perform the BICP on φ_l
5  if R_l ≠ UNSAT then
6      Add node φ̂_l into the partition tree
7      Send φ̂_l to the task buffer
8  {R_r, φ̂_r} ← perform the BICP on φ_r
9  if R_r ≠ UNSAT then
10     Add node φ̂_r into the partition tree
11     Send φ̂_r to the task buffer
```

Variable-level partitioning. This is the first attempt to perform variable-level partitioning for arithmetic theories. Each time it picks a variable and partitions the problem by dividing the feasible domain of the variable, leading to sub-problems, which can be further simplified via constraint propagation. 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.

Improved constraint propagation. The effectiveness of our partition strategy is closely related to the underlying constraint propagation techniques to simplify the sub-problems. We propose an improved version of Interval Constraint Propagation (ICP) [2,4], named Boolean and Interval Constraint Propagation (BICP), and integrate it within our variable-level partitioning strategy. The BICP conducts arithmetic feasible interval reasoning and successfully integrates Boolean propagation, allowing stronger propagation.

References

1. Cai, S., Li, B., Zhan, B., Zhang, X., Zhao, M.: Z3++ at SMT-COMP 2023. 18th International Satisfiability Modulo Theories Competition (SMT-COMP 2023) (2023)
2. Kulisch, U.W.: Complete interval arithmetic and its implementation on the computer. In: Cuyt, A., Krämer, W., Luther, W., Markstein, P. (eds.) Numerical Val-

idation in Current Hardware Architectures. pp. 7–26. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
3. de Moura, L., Bjørner, N.: Z3: An efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 337–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
4. Schupp, S., Ábrahám, E., Rossmanith, P., Loup, D.I.U.: Interval constraint propagation in SMT compliant decision procedures. Master's thesis, RWTH Aachen (2013)
5. 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)