Z3++ at SMT-COMP 2022

Shaowei Cai, Bohan Li, Jinkun Lin, Zhonghan Wang, Bohua Zhan,
Xindi Zhang, and Mengyu Zhao

State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
caisw@ios.ac.cn

1 Introduction

Z3++ is a derived SMT solver based on Z3-4.8.15[3]. It participates in the Single Query and Model Validation track of QF_LIA, QF_IDL, QF_NIA, QF_BV and QF_NRA logics.

A local search algorithm dedicated for integer arithmetic logic is proposed, and it is combined with Z3 using a simple portfolio. For solving QF_BV, z3++ is able to use the off-the-shelf SAT solvers. It also uses world-level and bit-level rewriting rules to simplify the constraints. In the QF_NRA division, we tune the QF_NRA tactic with a sophisticated flow. On this basis, many simplification and optimization techniques in arithmetic theory are introduced to Z3++.

2 Features

In this section, we introduce the features of Z3++ depending on the divisions.

Integer Arithmetic: We developed a novel local search solver called LA-IA for integer arithmetic, including QF_LIA, QF_IDL and QF_NIA. It is an improved version of LS-IDL [2], which participated in SMT-COMP2021 and won the QF_IDL division. LS-IA directly operates on variables, breaking through the traditional framework. We propose a local search framework by considering the distinctions between Boolean and integer variables. Moreover, we design a novel operator and scoring functions tailored for LIA, and propose a two-level operation selection heuristic. LS-IA is designed as an tactic of Z3. If Z3 fails to determine the instance within certain time limit, LS-IA is called to handle it.

QF_BV: The Z3++ bit-vector solver is now able to use off-the-shelf SAT solvers. Currently, it uses CaDiCaL[1] as the SAT back-ends by default. Before encoding the instance into a CNF Boolean formula, Z3++ applies word-level rewriting rules to simplify the bit-vector constraints. We add several new rewriting rules in Z3++, inspired by the Bitwuzla solver[4]. In addition, we transform the Boolean disjunction into conjunction before using and-inverter graph to simplify the bit-level constraints, as the resulting formulas are easier to solve for the SAT solver.

QF_NRA: We noticed that the order of variables assignment greatly affects the solving speed, and CDCL(NRA) framework works well when solving instances whose polynomials are multilinear. So we tune the QF_NRA tactic in Z3, and configure different solving flows for different instance scale, mainly including many heuristic arithmetic variable reordering strategies from CAD, and extend the attempt duration assigned to CDCL(T).

In theory, we implement sample-cell projection in Z3 nlsat explain module. Although the new operator is also a CAD-like projection operator which computes the cell (not necessarily cylindrical) containing a given sample such that each polynomial from the problem is sign-invariant on the cell, it is of singly exponential time complexity.

Also we implement a feasible region consistency checker for clauses which are univariate and currently unit before the main search. For best performance, it works in a loop and propagates as much as possible. The checker obtains the feasible region of variables through real root isolation and continuously takes the intersection of the regions. When the feasible region of some variable is empty, it means the instance is unsat. At last the checker will add the feasible regions into clauses which can guide the as lemmas.

3 Webpage

Further information can be found at
`https://z3-plus-plus.github.io/`

4 Contributors

Here is respective contribution of authors. Shaowei Cai supervised the whole project. Bohan Li accomplished the local search algorithm for integer arithmetic logics(including QF_IDL, QF_LIA, QF_NIA), and tuned the algorithm to combine with Z3. Jinkun Lin accomplished the QF_BV division. Xindi Zhang participated in the discussion and provided important idea in designing local search for integer arithmetic logics, and participated in QF_BV. Mengyu Zhao, Zhonghan Wang and Bohua Zhan are responsible for the QF_NRA division.

References

1. Biere, A., Fazekas, K., Fleury, M., Heisinger, M.: CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. In: Balyo, T., Froleyks, N., Heule, M., Iser, M., Järvisalo, M., Suda, M. (eds.) Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions. Department of Computer Science Report Series B, vol. B-2020-1, pp. 51–53. University of Helsinki (2020)
2. CAI, S., LI, B., ZHANG, X.: Yicesls on smt comp2021
3. Moura, L.d., Bjørner, N.: Z3: An efficient smt solver. In: International conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 337–340. Springer (2008)
4. Niemetz, A., Preiner, M.: Bitwuzla at the SMT-COMP 2020. CoRR **abs/2006.01621** (2020), https://arxiv.org/abs/2006.01621