Z3++ at SMT-COMP 2022

Shaowei Cai[1], Bohan Li[1], Jinkun Lin[2], Zhonghan Wang[1], Bohua Zhan[1], Xindi Zhang[1], and Mengyu Zhao[1]

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

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_BV, QF_LIA, QF_IDL, QF_NIA and QF_NRA logics.

For solving QF_BV, z3++ is able to use the off-the-shelf SAT solvers, and it also introduces additional word-level and bit-level rewriting rules to simplify the constraints. For solving QF_LIA, QF_IDL and QF_NIA, a local search solver dedicated for integer arithmetic logic is developed, and it is combined with Z3 using a simple portfolio. In the QF_NRA division, we implement a feasible region consistency checker before the main search, and implement sample-cell projection in the NLSAT. Additionally, we tune the QF_NRA tactic with a sophisticated flow.

2 Features

In this section, we introduce the new features of Z3++ (compared to Z3) on the divisions we mention in this document.

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-end by default. We add several new rewriting rules in Z3++, inspired by the Bitwuzla solver [4]. Before encoding the instance into a CNF Boolean formula, Z3++ applies word-level rewriting rules to simplify the bit-vector constraints. In addition, we transform the Boolean disjunction into conjunction before using and-inverter graph (AIG) to simplify the bit-level constraints, as the resulting formulas are easier to solve for the SAT solver.

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 extended version of LS-LIA [2], which is a local search solver for QF_LIA (including QF_IDL). 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, and propose a two-level operation selection heuristic. LS-IA is integrated into Z3 as an tactic. If Z3 fails to solve the instance within a certain time limit, LS-IA is called to tackle it.

QF_NRA: We noticed that the order of variables to be assigned greatly affects the performance on many instances, so we employ different solving flows, mainly including various heuristic arithmetic variable reordering strategies from CAD. Also, as DPLL(T) framework works particularly well for multilinear instances, we extend the time limit of DPLL(T) for such instances.

As for NRA theory solving, we implement sample-cell projection in 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 unit before the main search. The checker obtains the feasible region of variables through real root isolation and continuously takes the intersection of the regions. If the feasible region of some variable becomes empty, it means the instance is unsatisfiable. Otherwise, the obtained feasible regions are added to the instance as lemmas.

3 Webpage

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

4 Contributors

This project is a result of team work. Shaowei Cai launches and leads the Z3++ project. Shaowei Cai (proposing the local search methodology), Bohan Li (main developer) and Xindi Zhang (code review in early stage, and participates in idea discussions) are contributors to integer arithmetic theories (including QF_IDL, QF_LIA, QF_NIA), and the algorithm is a result of many discussions among them. Jinkun Lin is the main contributor and developer for the QF_BV division, while Xindi Zhang provides some help on integrating CaDiCaL. Zhonghan Wang and Mengyu Zhao are main developers for the QF_NRA division, under the supervisions of Shaowei Cai and Bohua Zhan.

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.: Local search for smt on linear integer arithmetic. In: Proc. of CAV 2022. p. accepted (2022)

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`