CVC4-SymBreak: Derived SMT solver at SMT Competition 2019

Saket Dingliwal
Indian Institute of Technology, Delhi
cs1150254@iitd.ac.in

Ronak Agarwal
Indian Institute of Technology, Delhi
cs1150252@iitd.ac.in

Happy Mittal
Indian Institute of Technology, Delhi
csz138233@cse.iitd.ac.in

Parag Singla
Indian Institute of Technology, Delhi
parags@cse.iitd.ac.in

***Abstract*—This is the System Description Document for CVC4-SymBreak, which is a derived SMT solver based on CVC4 for entering SMT-COMP 2019. We list below only the major extensions/technical innovations from the CVC4 solver used. For more details about the basic algorithms used, please refer to the CVC4 original paper [1] or the CVC4 website [2]. The link for the derived solver is https://www.starexec.org/starexec/secure/details/solver.jsp?id=23704**

I. Overview

CVC4-SymBreak extends the 2018 version of CVC4 [3], by additionally exploiting symmetries inherent in the SMT problem. It also uses SyMT tool [4] for detecting symmetries as well as extends to the idea of adding symmetry breaking clauses to prune search space in SMT problems as used in [5]. It is a non-competing entry to the Single Query Track in the categories of AUFLIA, AUFLIRA, AUFNIRA, UFNIA, QF_AUFLIA, QF_AX, QF_LIA, QF_NIA, QF_NRA, QF_LRA.

II. Basic Algorithm

The basic algorithm for SMT solving used by CVC4 involves conversion of SMT problem into propositional SAT problem (referred to as boolean skeleton in this document) by assigning different boolean variables to different theory constraints. It is followed by partially assigning truth values to the boolean variables and checking their consistency with the theory constraints at each step. The problem is considered solved when either it is proved that no assignment satisfying the boolean skeleton is consistent with the theory or a truth assignment for the boolean variables is found which is consistent with their respective theory constraints.

III. Technical Innovations

We add the following two major extensions to the basic algorithm described above by the CVC4 solver.

A. Symmetry Breaking Techniques

Symmetry breaking technique was introduced in propositional logic in [9]. It was based on the idea of reducing search space for finding a truth assignment to the propositional problem. Similar idea was used in [5] for QF_UF category in SMT problems by imposing a variable ordering on the constants. We extend the idea to variables in the boolean skeleton formed in any SMT problem in general. This helps to reduce the search space significantly and is particularly helpful in categories where theory-consistency checks are computationally expensive. We impose a global variable-ordering for the skeleton variables formed by CVC4, use SyMT tool [4] to detect their symmetric permutations, add a lex-leader symmetry breaking predicate (SBPs) as proposed in [9] to the skeleton clause list and then solve the new problem again by using CVC4. We have also been careful in choosing the variable ordering that maximises the impact of the SBPs formed. We also restrict the size of the SBPs formed to reduce the overheads [10].

B. Ensemble Techniques

On experimenting with the symmetry breaking technique for the SMT solvers, we found that in some problems, it helps to reduce time taken by the solver significantly while in others, it in-fact increases the time used. This behaviour could be explained by the fact that sometimes, pruning the search-space rejects a model that might have been found early-on in the search path, while sometimes it helps to find a solution early by cutting out parts in the search tree searched before reaching to that solution. To get the best of both the worlds, we ran CVC4 with SBPs for some fixed portion of time in the time-limit and then restarted the CVC4 without the additional clauses. This helps to significantly boost the performance.

IV. Tools Used

We acknowledge the use of the following tools/solvers for making CVC4-SymBreak

A. CVC4 and it's dependencies

The base solver used is CVC4 2018 version. We use the competition build with Cadical [8] and ABC solvers integrated into it. CVC4 is modified to add new clauses(SBPs) to the boolean skeleton formed.

B. SyMT and it's dependencies

SyMT [4] is used to detect symmetries in the SMT problem. It converts the problem into graph automorphism problem and uses SAUCY [7] internally for solving it. It also requires binary for VeriT [6] for it's functioning.

REFERENCES

[1] Barrett, Clark, et al. "Cvc4." International Conference on Computer Aided Verification. Springer, Berlin, Heidelberg, 2011.

[2] CVC4 website. http://cvc4.cs.stanford.edu, 2018.

[3] Barrett, Clark, et al. "CVC4 at the SMT Competition 2018." arXiv preprint arXiv:1806.08775 (2018).

[4] Areces, Carlos, et al. "SyMT: finding symmetries in SMT formulas." SMT Workshop 2013 11th International Workshop on Satisfiability Modulo Theories. 2013.

[5] Dharbe, David, et al. "Exploiting symmetry in SMT problems." International Conference on Automated Deduction. Springer, Berlin, Heidelberg, 2011.

[6] Bouton, Thomas, et al. "veriT: an open, trustable and efficient SMT-solver." International Conference on Automated Deduction. Springer, Berlin, Heidelberg, 2009.

[7] http://vlsicad.eecs.umich.edu/BK/SAUCY/

[8] http://fmv.jku.at/cadical/

[9] Crawford, James, et al. "Symmetry-breaking predicates for search problems." KR 96 (1996): 148-159.

[10] Aloul, Fadi A., Karem A. Sakallah, and Igor L. Markov. "Efficient symmetry breaking for boolean satisfiability." IEEE Transactions on Computers 55.5 (2006): 549-558.