veriT at SMT-COMP 2020

Haniel Barbosa[1], Daniel El Ouraoui[2], Pascal Fontaine[2,3], Hans-Jörg Schurr[2]*

[1] Department of Computer Science, Universidade Federal de Minas Gerais (UFMG)
hbarbosa@dcc.ufmg.br
[2] CNRS, Inria, and the University of Lorraine, Nancy, France;
{daniel.elouraoui, pascal.fontaine, hans-jorg.schurr}@loria.fr
[3] Université de Liège, Belgium

URL : http://www.verit-solver.org — Seed : 153

veriT is a satisfiability modulo theory (SMT) solver developed by University of Lorraine, Inria (Nancy, France). veriT provides an open, trustable and reasonably efficient decision procedure [4] for the logic of quantifier-free formulas over uninterpreted symbols, linear real arithmetics, and the combination thereof. It also handles linear arithmetics over integers, and has quantifier reasoning using trigger- and conflict-based instantiation [2] as well as enumerative instantiation [8]. Recently, veriT was extended to higher order logic [3]. Also, veriT is proof-producing [6, 1]. In the past year proof-production was further refined to facilitate reconstructed in the Isabelle/HOL proof assistant [7].

veriT is written in C and accepts the input formats SMT-LIB 2.6 and DIMACS. It integrates a CDCL($\mathcal{T}$)-based boolean satisfiability engine with a Nelson-Oppen like combination of decision and semi-decision procedures with propagation of model equalities, and implements simplifications such as symmetry-based reductions [5]. The tool is open-source and distributed under the BSD licence.

The veriT version competing in SMT-COMP 2020 features new quantifier instantiation heuristics and an improved scheduling script.

To prevent saturating the solver with unhelpful instances created by trigger-based and enumerative instantiation, we added a parameter to limit the number of generated instances for each quantified formula. A second parameter restricts skolemization. To eliminate existential quantifiers veriT introduces skolem constants. This can lead to runaway behavior. In particular, when an existential quantifier appears under a universal quantifier. In this case a new skolem constant is introduced for every new instance, creating as meany constants as new instances. To avoid this phenomenon a counter is attached to each skolemized formula and the parameter is used to decide whether the formula should be skolemized.

As in earlier versions of veriT, a scheduling script runs the executable according to a fixed list of command line options together with a timeout. In the past this list was handcrafted, which complicated updates. We now generate it automatically. To do so we first collected the necessary solving time of the SMT-LIB benchmarks for a predefined list of options. Using a fixed set of timeouts

* The author order is strictly alphabetic.

we then encoded this as an integer programming problem that maximizes the number of problems solved in the total timeout. In a second step we calculated the order of the resulting option list that minimizes the total CPU time. We used this process to generate a second scheduler optimized for a timeout of 24 second. This variant participates out of competition as *veriT+vite*.

veriT participates in the following divisions: ALIA AUFLIA AUFLIRA LIA UF UFIDL UFLIA UFLRA QF_ALIA QF_AUFLIA QF_IDL QF_LIA QF_LRA QF_RDL QF_UF QF_UFIDL QF_UFLIA QF_UFLRA.

Acknowledgements The development of veriT is funded by the European Research Council (ERC) starting grant Matryoshka (713999).

References

1. H. Barbosa, J. C. Blanchette, and P. Fontaine. Scalable fine-grained proofs for formula processing. In L. de Moura, editor, *Proc. Conference on Automated Deduction (CADE)*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2017.
2. H. Barbosa, P. Fontaine, and A. Reynolds. Congruence closure with free variables. In A. Legay and T. Margaria, editors, *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, volume 10206 of *Lecture Notes in Computer Science*, pages 214–230, 2017.
3. H. Barbosa, A. Reynolds, D. E. Ouraoui, C. Tinelli, and C. W. Barrett. Extending SMT solvers to higher-order logic. In P. Fontaine, editor, *Proc. Conference on Automated Deduction (CADE-27)*, volume 11716 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2019.
4. T. Bouton, D. C. B. de Oliveira, D. Déharbe, and P. Fontaine. veriT: an open, trustable and efficient SMT-solver. In R. A. Schmidt, editor, *Proc. Conference on Automated Deduction (CADE-22)*, 2009.
5. D. Déharbe, P. Fontaine, S. Merz, and B. Wolzenlogel Paleo. Exploiting Symmetry in SMT Problems. In N. Bjørner and V. Sofronie-Stokkermans, editors, *Proc. Conference on Automated Deduction (CADE-23)*, 2011.
6. D. Déharbe, P. Fontaine, and B. Wolzenlogel Paleo. Quantifier Inference Rules for SMT proofs. In *First Workshop on Proof eXchange for Theorem Proving (PxTP)*, 2011.
7. M. Fleury and H. Schurr. Reconstructing veriT proofs in Isabelle/HOL. In G. Reis and H. Barbosa, editors, *Sixth Workshop on Proof eXchange for Theorem Proving, (PxTP)*, volume 301 of *EPTCS*, pages 36–50, 2019.
8. A. Reynolds, H. Barbosa, and P. Fontaine. Revisiting enumerative instantiation. In D. Beyer and M. Huisman, editors, *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, volume 10806 of *Lecture Notes in Computer Science*, pages 112–131. Springer, 2018.