veriT at SMT-COMP 2022

Bruno Andreotti[1], Haniel Barbosa[1], Pascal Fontaine[2,3], Hans-Jörg Schurr[2⋆]

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

URL : http://www.verit-solver.org — Seed : 109

veriT is a satisfiability modulo theory (SMT) solver developed at the University of Lorraine, Inria (Nancy, France), and Université de Liège (Liège, Belgium). 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-, enumeration- and conflict-based instantiation [2] as well as enumerative instantiation [7]. Recently, veriT was extended to higher order logic [3].

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.

Also, veriT is proof-producing [6, 1]. Since the 2021 release of the proof assistant Isabelle/HOL can reconstruct proofs generated by veriT [9]. Isabelle/HOL users can now use veriT as a proof tactic. At SMT-COMP 2022 veriT also participates in the proof exhibition track. The solver outputs proofs in the Alethe format [8].[1]. The proofs are then checked by the Alethe proof checker[2], a dedicated high-performance proof checker written in Rust.

Since the SMT-COMP 2020, we generate the scheduling scripts for veriT automatically. Our schedule-generator now supports multiple, nested timeouts. Hence, veriT's schedule is first optimized for a 24 second timeout and then optimized to use the remaining time optimally. The schedule generation toolbox is now publicly available[3].

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.

⋆ The author order is strictly alphabetic.
[1] The current format specification is available at https://verit.loria.fr/documentation/alethe-spec.pdf
[2] Available at https://github.com/ufmg-smite/alethe-proof-checker.
[3] Available at https://gitlab.uliege.be/verit/schedgen/.

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. 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.
8. H.-J. Schurr, M. Fleury, H. Barbosa, and P. Fontaine. Alethe: Towards a Generic SMT Proof Format (extended abstract). In *PxTP 2021 - 7th Workshop on Proof eXchange for Theorem Proving*, volume 336 of *EPTCS*, pages 49–54, Pittsburgh, PA / virtual, United States, Sept. 2021.
9. H.-J. Schurr, M. Fleury, and M. Desharnais. Reliable reconstruction of fine-grained proofs in a proof assistant. In *CADE 28*, LNCS, 2021.