veriT and veriT+raSAT+Redlog: System Description for SMT-COMP 2019

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

[1] University of Iowa, Iowa City, USA
haniel-barbosa@uiowa.edu
[2] CNRS, Inria, and the University of Lorraine, Nancy, France;
{daniel.elouraoui, pascal.fontaine, hans-jorg.schurr}@loria.fr

http://www.verit-solver.org

veriT [5] 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 [6] 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 [13]. Finally, veriT is proof-producing [9, 1]. 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 [7], and implements simplifications such as symmetry-based reductions [8]. The tool is open-source and distributed under the BSD licence.

veriT participates in the competition with two variants: the standalone veriT and veriT+raSAT+Redlog.

In the last year veriT has gained experimental support for higher-order logic [3] and machine learning for instance selection. We also improved the proof output and are currently improving the instantiation techniques. Since those changes are not relevant for the competition, the version of veriT competing in SMT-COMP 2019 is similar to the entry of 2018. We do not expect major changes in the competition performance.

veriT+raSAT+Redlog is a combination of veriT with the following two solvers for solving nonlinear arithmetic.

1. The raSAT loop [14, 12], which is an extension of Interval Constraint Propagation [4] with testing and the application of the Intermediate Value Theorem (IVT).

2. Redlog [10] is a key component of the open-source computer algebra system Reduce. It supplements Reduce's comprehensive collection of methods from symbolic computation with 100+ functions operating on formulas in interpreted first-order logic. Formulas coexist and share data structures with conventional objects of symbolic computation within one homogeneous system. Within a rich infrastructure of methods on first-order formulas, Redlog has a strong focus on quantifier elimination and decision procedures for various algebraic theories.

The combination is similar to the idea from SMT-RAT [11] where when each box of ICP becomes smaller than a threshold ϵ, a complete framework (CAD) is called to solve the remaining unknown constraints over such a small box. The main difference here is that when all boxes of the raSAT loop become smaller than ϵ, Redlog is utilized to solve the unknown constraints

[*] The author order is strictly alphabetic.

with the box contracted from $[-\infty, \infty]^n$ where n is the number of variables. As a result, for a conjunction of polynomial constraints, Redlog is called only once from the raSAT loop. While Redlog complements the raSAT loop in the completeness, the small box of the raSAT loop assists Redlog to prune its searching space. At this moment, ϵ is set to $\frac{1}{64}$ by default, which is experimentally selected based on the performances of different values of ϵ.

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.

veriT+raSAT+Redlog participates in the following divisions: QF_NRA, QF_UFNRA.

Acknowledgements We thank David Déharbe, Maximilian Jaroschek, Marek Kosta, Mizuhito Ogawa, Thomas Sturm, and Vu Xuan Tung. Their contributions have been instrumental to the development of veriT.

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

References

[1] Haniel Barbosa, Jasmin Christian Blanchette, and Pascal Fontaine. Scalable fine-grained proofs for formula processing. In Leonardo de Moura, editor, *Proc. Conference on Automated Deduction (CADE)*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2017.

[2] Haniel Barbosa, Pascal Fontaine, and Andrew Reynolds. Congruence closure with free variables. In Axel Legay and Tiziana 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] Haniel Barbosa, Andrew Reynolds, Daniel El Ouraoui, Cesare Tinelli, and Clark Barrett. Extending SMT solvers to higher-order logic. In *Proc. Conference on Automated Deduction (CADE-27)*, 2019.

[4] Frédéric Benhamou and Laurent Granvilliers. Continuous and interval constraints. In P. van Beek F. Rossi and T. Walsh, editors, *Handbook of Constraint Programming*, pages 571–604. Elsevier, 2006.

[5] Thomas Bouton, Diego Caminha B. De Oliveira, David Déharbe, and Pascal Fontaine. veriT: An open, trustable and efficient SMT-Solver. In *Proceedings of the 22nd International Conference on Automated Deduction*, CADE-22, pages 151–156, Berlin, 2009. Springer.

[6] Thomas Bouton, Diego Caminha B. de Oliveira, David Déharbe, and Pascal Fontaine. veriT: an open, trustable and efficient SMT-solver. In Renate A. Schmidt, editor, *Proc. Conference on Automated Deduction (CADE-22)*, 2009.

[7] Diego Caminha B. de Oliveira, David Déharbe, and Pascal Fontaine. Combining decision procedures by (model-)equality propagation. *Science of Computer Programming*, 77(4):518 – 532, 2012.

[8] David Déharbe, Pascal Fontaine, Stephan Merz, and Bruno Wolzenlogel Paleo. Exploiting Symmetry in SMT Problems. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *Proc. Conference on Automated Deduction (CADE-23)*, 2011.

[9] David Déharbe, Pascal Fontaine, and Bruno Wolzenlogel Paleo. Quantifier Inference Rules for SMT proofs. In *First Workshop on Proof eXchange for Theorem Proving (PxTP)*, 2011.

[10] Andreas Dolzmann and Thomas Sturm. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, 1997.

[11] Corzilius Florian, Loup Ulrich, Junges Sebastian, and Ábrahám Erika. *SMT-RAT: An SMT-Compliant Nonlinear Real Arithmetic Toolbox*, pages 442–448. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[12] To Van Khanh and Mizuhito Ogawa. Third workshop on tools for automatic program analysis (tapas' 2012) smt for polynomial constraints on real numbers. *Electronic Notes in Theoretical Computer Science*, 289:27 – 40, 2012.

[13] Andrew Reynolds, Haniel Barbosa, and Pascal Fontaine. Revisiting enumerative instantiation. volume 10806 of *Lecture Notes in Computer Science*, pages 112–131. Springer, 2018.

[14] Vu Xuan Tung, To Van Khanh, and Mizuhito Ogawa. *raSAT: An SMT Solver for Polynomial Constraints*, pages 228–237. Springer International Publishing, Cham, 2016.