MathSAT5 (Nonlinear) at the SMT Competition 2019

Ahmed Irfan
Stanford University, United States
Email: irfan@cs.stanford.edu

Alberto Griggio, Alessandro Cimatti
Fondazione Bruno Kessler, Italy
Email: {griggio, cimatti}@fbk.eu

Roberto Sebastiani
University of Trento, Italy
Email: roberto.sebastiani@unitn.it

OVERVIEW

MathSAT5 [1] is a lazy SMT solver [2] based on the DPLL(T) architecture [3], and it uses MiniSAT [4] as the underlying SAT solver. It supports most of the SMT-LIB [5] theories and provides many SMT functionalities (e.g. unsatisfiable cores [6], interpolation, ALLSMT). It does not offer support for quantifiers.

In the last couple of years, the support for nonlinear arithmetic and transcendental functions has been added to MathSAT, based on incremental linearization. The main idea of incremental linearization is that of trading the use of expensive, exact solvers for nonlinear arithmetic for much less expensive solvers for linear arithmetic and uninterpreted functions. The approach is based on an abstraction-refinement loop that uses SMT(UFLA) as abstract domain. The uninterpreted functions are used to model nonlinear multiplications, which are incrementally axiomatized, by means of linear constraints, with a lemma-on-demand [7] approach.

Details about incremental linearization can be found in [8], [9], [10], [11], [12] and theory solvers can be found in [13], [14], [15], [16], [17].

PARTICIPATION AND CONFIGURATIONS

MathSAT5 will participate in the single query, incremental and unsat core tracks, entering the following (nonlinear) categories:

Single Query track: QF_ANIA, QF_AUFNIA, QF_NIA, QF_NIRA, QF_NRA, QF_UFNIA, QF_UFNRA.
Incremental track: QF_ANIA, QF_AUFBVNIA, QF_NIA, QF_UFNIA.
Unsat Core track: QF_ANIA, QF_AUFNIA, QF_NIA, QF_NIRA, QF_NRA, QF_UFNIA, QF_UFNRA.

Two versions of MathSAT5 have been submitted: MathSAT-default and MathSAT-na-ext.

MathSAT-default:

This is the public release version 5.5.4 with some fixes. Essentially, it employs the strategy for nonlinear as described in [11].

MathSAT-na-ext:

This is an extension of MathSAT-default. It differs from MathSAT-default in the following ways:

- use of lazier strategy for the instantiation of linearization lemmas;
- try to minimize the boolean assignment that are given to theory solvers;
- use of backward implication in addition to forward implication of the tangent lemma:
  - $v_1 * v_2 < b * v_1 + a * v_2 - a * b \rightarrow ((v_1 > a \wedge v_2 < b) \vee (v_1 < a \wedge v_2 > b))$
  - $v_1 * v_2 > b * v_1 + a * v_2 - a * b \rightarrow ((v_1 < a \wedge v_2 < b) \vee (v_1 > a \wedge v_2 > b))$

  where v_1, v_2 are variables and a, b are rational/integer constants;
- mark linearization lemmas as temporary learnt clauses and therefore these lemmas can be dropped by the learnt clause DB cleaning heuristics.

MAGIC NUMBER: 512

REFERENCES

[1] A. Cimatti, A. Griggio, B. J. Schaafsma, and R. Sebastiani, "The MathSAT5 SMT solver," in *TACAS*, vol. 7795 of *LNCS*, pp. 93–107, Springer, 2013.

[2] C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli, "Satisfiability modulo theories," in *Handbook of Satisfiability*, vol. 185 of *Frontiers in Artificial Intelligence and Applications*, pp. 825–885, IOS Press, 2009.

[3] R. Nieuwenhuis, A. Oliveras, and C. Tinelli, "Solving SAT and SAT modulo theories: From an abstract davis–putnam–logemann–loveland procedure to dpll(*T*)," *J. ACM*, vol. 53, no. 6, pp. 937–977, 2006.

[4] N. Eén and N. Sörensson, "An extensible sat-solver," in *SAT*, vol. 2919 of *Lecture Notes in Computer Science*, pp. 502–518, Springer, 2003.

[5] C. Barrett, P. Fontaine, and C. Tinelli, "The Satisfiability Modulo Theories Library (SMT-LIB)." `www.SMT-LIB.org`, 2016.

[6] A. Cimatti, A. Griggio, and R. Sebastiani, "Computing small unsatisfiable cores in satisfiability modulo theories," *Journal of Artificial Intelligence Research*, vol. 40, pp. 701–728, 2011.

[7] L. De Moura, H. Rueß, and M. Sorea, "Lemmas on demand for satisfiability solvers," *Proc. SAT*, vol. 2, pp. 244–251, 2002.

[8] A. Cimatti, A. Griggio, A. Irfan, M. Roveri, and R. Sebastiani, "Invariant checking of NRA transition systems via incremental reduction to LRA with EUF," in *TACAS*, vol. 10205 of *LNCS*, pp. 58–75, 2017.

[9] A. Irfan, *Incremental Linearization for Satisfiability and Verification Modulo Nonlinear Arithmetic and Transcendental Functions*. PhD thesis, University of Trento, 2018.

[10] A. Cimatti, A. Griggio, A. Irfan, M. Roveri, and R. Sebastiani, "Incremental linearization for satisfiability and verification modulo nonlinear arithmetic and transcendental functions," *ACM TOCL*, vol. 19, pp. 19:1–19:52, 2018.

[11] A. Cimatti, A. Griggio, A. Irfan, M. Roveri, and R. Sebastiani, "Experimenting on solving nonlinear integer arithmetic with incremental linearization," in *SAT*, LNCS, pp. 383–398, Springer, 2018.

[12] A. Reynolds, C. Tinelli, D. Jovanovic, and C. Barrett, "Designing theory solvers with extensions," in *FroCoS*, vol. 10483 of *LNCS*, Springer, 2017.

[13] A. Griggio, *An effective SMT engine for Formal Verification*. PhD thesis, University of Trento, 2009.

[14] A. Franzén, *Efficient solving of the satisfiability modulo bit-vectors problem and some extensions to SMT*. PhD thesis, University of Trento, 2010.

[15] A. Griggio, "A practical approach to satisfiability modulo linear integer arithmetic," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 8, pp. 1–27, 2012.

[16] M. Bromberger and C. Weidenbach, "Fast cube tests for lia constraint solving," in *International Joint Conference on Automated Reasoning*, pp. 116–132, Springer, 2016.

[17] J. Christ and J. Hoenicke, "Weakly equivalent arrays," in *International Symposium on Frontiers of Combining Systems*, pp. 119–134, Springer, 2015.