SPASS-SATT v1.1

Martin Bromberger[1,2], Mathias Fleury[1,2], Simon Schwarz[1,2], and Christoph Weidenbach[1]

[1] Max Planck Institute for Informatics and Saarland University, Saarland Informatics Campus, Germany {mbromber,mfleury,sschwarz,weidenb}@mpi-inf.mpg.de
[2] Graduate School of Computer Science, Saarland Informatics Campus, Germany

Abstract

SPASS-SATT is a CDCL(LA) solver for linear rational and linear mixed/integer arithmetic. This system description summarizes its main features.

SPASS-SATT (v1.1) is a sound and complete CDCL(LA) [3, 12, 13, 14] solver for quantifier-free linear rational and linear mixed/integer arithmetic. It is a from-scratch implementation except for some basic data structures taken from the SPASS [17] superposition theorem prover. It is available through the SPASS-Workbench [1].

The techniques that appeared first in SPASS-SATT are the unit cube test [6, 7] and bounding transformations [4]. Concerning preprocessing, SPASS-SATT is the first SMT solver implementing the small-clause-normal-form algorithm [15]. Further important techniques implemented in SPASS-SATT have already been available in other SMT solvers such as CVC4 [2], MathSAT [8], Yices [10], and Z3 [9], but not all in one tool: (*i*) the implementation of branch and bound as a separate theory solver and a number of improvements to the simplex implementation such as a priority queue for pivot selection, integer coefficients instead of rational coefficients, dynamically switching between native and arbitrary precision integers, and backing-up versus recalculating simplex states, (*ii*) weakened early pruning [16], decison recommendations, unate propagations [11], and bound refinements [11] for the interaction between the SAT and theory solver, and (*iii*) preprocessing techniques for if-then-else operators and pseudo-boolean inequalities. Although these techniques are contained in existing SMT solvers, not all have been described in the respective literature. We remedy this in an extended version of this system description [5].

For the 14th International Satisfiability Modulo Theories Competition (SMT-COMP 2019), SPASS-SATT runs in its default configuration, i.e., without any command line options. By default all mentioned techniques are turned on.

References

[1] The SPASS Workbench. https://www.spass-prover.org/.

[2] C. Barrett, C. Conway, M. Deters, L. Hadarean, D. Jovanović, T. King, A. Reynolds, and C. Tinelli. CVC4. In *CAV*, volume 6806 of *LNCS*. 2011.

[3] C. Barrett, R. Nieuwenhuis, A. Oliveras, and C. Tinelli. Splitting on demand in SAT modulo theories. In *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 4246 of *LNCS*, pages 512–526. 2006.

[4] M. Bromberger. A reduction from unbounded linear mixed arithmetic problems into bounded problems. In *Automated Reasoning*, volume 10900 of *LNCS*, 2018.

[5] M. Bromberger, M. Fleury, S. Schwarz, and C. Weidenbach. SPASS-SATT a CDCL(LA) solver. In *CADE-27*, Lecture Notes in Computer Science. Springer, 2019. (Forthcoming).

[6] M. Bromberger and C. Weidenbach. Fast cube tests for LIA constraint solving. In *IJCAR 2016*, volume 9706 of *LNCS*. 2016.

[7] M. Bromberger and C. Weidenbach. New Techniques for Linear Arithmetic: Cubes and Equalities. *Formal Methods in System Design*, 51(3), 2017.

[8] A. Cimatti, A. Griggio, B. Schaafsma, and R. Sebastiani. The MathSAT5 SMT Solver. In *TACAS*, volume 7795 of *LNCS*, 2013.

[9] L. M. de Moura and N. Bjørner. Z3: an efficient SMT solver. In C. R. Ramakrishnan and J. Rehof, editors, *TACAS 2008*, volume 4963 of *LNCS*, pages 337–340. Springer, 2008.

[10] B. Dutertre. Yices 2.2. In *CAV 2014*, volume 8559 of *LNCS*, 2014.

[11] B. Dutertre and L. M. de Moura. A fast linear-arithmetic solver for DPLL(T). In *CAV*, volume 4144 of *LNCS*. 2006. Extended version: Integrating simplex with DPLL(T). Tech. rep., CSL, SRI INTERNATIONAL (2006).

[12] H. Ganzinger, G. Hagen, R. Nieuwenhuis, A. Oliveras, and C. Tinelli. DPLL(T): fast decision procedures. In *CAV*. Springer, 2004.

[13] R. Nieuwenhuis and A. Oliveras. DPLL(T) with exhaustive theory propagation and its application to difference logic. In *CAV*, pages 321–334. Springer, 2005.

[14] 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*, 53(6):937–977, 2006.

[15] A. Nonnengart and C. Weidenbach. Computing small clause normal forms. In *Handbook of Automated Reasoning*, volume 1. Elsevier, 2001.

[16] R. Sebastiani. Lazy satisability modulo theories. *JSAT*, 3(3-4):141–224, 2007.

[17] C. Weidenbach, D. Dimova, A. Fietzke, M. Suda, and P. Wischnewski. SPASS version 3.5. In R. A. Schmidt, editor, *22nd International Conference on Automated Deduction (CADE-22)*, volume 5663 of *Lecture Notes in Artificial Intelligence*, pages 140–145, Montreal, Canada, August 2009. Springer.