SMTInterpol and SMTInterpol-remus

Versions 2.5-823-g881e8631 and 2.5-948-gc776ef06

Leonard Fichtner, Jochen Hoenicke, Moritz Mohr, and Tanja Schindler

University of Freiburg
{hoenicke,schindle}@informatik.uni-freiburg.de

June 13, 2021

Description

SMTInterpol is an SMT solver written in Java and available under LGPL v3. It supports the combination of the theories of uninterpreted functions, linear arithmetic over integers and reals, arrays, and datatypes. Furthermore it can produce models, proofs, unsatisfiable cores, and interpolants. The solver reads input in SMT-LIB format. It includes parsers for DIMACS, AIGER, and SMT-LIB version 2.6.

The solver is based on the well-known DPLL(T)/CDCL framework [GHN+04]. It uses variants of standard algorithms for CNF conversion [PG86] and congruence closure [NO05]. The solver for linear arithmetic is based on Simplex [DdM06], the sum-of-infeasibility algorithm [KBD13], and branch-and-cut for integer arithmetic [CH15a, DDA09]. The array decision procedure is based on weak equivalences [CH15b] and includes an extension for constant arrays [HS19]. The solver for quantified formulas performs an incremental search for conflicting and unit-propagating instances of quantified formulas [HS21] which is complemented with a version of enumerative instantiation [RBF18] to ensure completeness for the finite almost uninterpreted fragment [GdM09]. Theory combination is performed based on partial models produced by the theory solvers [dMB08].

This year, a solver for datatypes has been added. It is based on the rules presented in [BST07]. SMTInterpol has also been extended with unsatisfiable core enumeration based on the ReMUS algorithm [BCB18]. The extension allows for choosing a best unsatisfiable core according to several heuristics.

The main focus of SMTInterpol is the incremental track. This track simulates the typical application of SMTInterpol where a user asks multiple queries. As an extension SMTInterpol supports quantifier-free interpolation for all supported theories except for datatypes [CH16, HS18, HS19]. This makes it useful as a backend for software verification tools. In particular, Ultimate Automizer[1] and CPAchecker[2], the winners of the SV-COMP 2016–2021, use SMTInterpol.

Competition Versions

The version of SMTInterpol submitted to the SMT-COMP 2021 contains the new solver for datatypes. A peculiarity in this year's versions is that the proof check mode is enabled, which means that proofs of unsatisfiability are tracked and checked by the internal proof checker (an exception are lemmas of the theory of datatypes).

[1] https://ultimate.informatik.uni-freiburg.de/
[2] https://cpachecker.sosy-lab.org/

SMTInterpol-remus is a variant of SMTInterpol that uses an enumeration algorithm in order to find smaller unsatisfiable cores if possible.

Webpage and Sources

Further information about SMTInterpol can be found at

https://ultimate.informatik.uni-freiburg.de/smtinterpol/

The sources are available via GitHub

https://github.com/ultimate-pa/smtinterpol

Authors

The code was written by Jürgen Christ, Daniel Dietsch, Leonard Fichtner, Joanna Greulich, Matthias Heizmann, Jochen Hoenicke, Moritz Mohr, Alexander Nutz, Markus Pomrehn, Pascal Raiola, and Tanja Schindler.

Logics, Tracks and Magic Number

SMTInterpol participates in the single query track, the incremental track, the unsat core track, and the model validation track. It supports all combinations of uninterpreted functions, linear arithmetic, arrays, and datatypes, and participates in the following logics:

ALIA, ANIA, AUFLIA, AUFLIRA, AUFNIRA, LIA, LRA, QF_ALIA, , QF_ANIA, QF_AUFLIA, QF_AX, QF_DT, QF_IDL, QF_LIA, QF_LIRA, QF_LRA, QF_NIA, QF_RDL, QF_UF, QF_UFDT, QF_UFDTLIRA, QF_UFIDL, QF_UFLIA, QF_UFLRA, QF_UFNRA, QF_UFNIA, UF, UFIDL, UFLIA, UFLRA, UFNIA, UFNRA.

SMTInterpol-remus participates in the following logics of the unsat core track: LIA, AUFLIA, AUFLIRA, LIA, LRA, QF_ALIA, QF_AUFLIA, QF_AX, QF_DT, QF_IDL, QF_LIA, QF_LIRA, QF_LRA, QF_RDL, QF_UF, QF_UFDT, QF_UFDTLIRA, QF_UFIDL, QF_UFLIA, QF_UFLRA, UF, UFIDL, UFLIA, UFLRA

Magic Numbers: 384076683+1

References

[BCB18] Jaroslav Bendík, Ivana Cerná, and Nikola Benes. Recursive online enumeration of all minimal unsatisfiable subsets. In *ATVA*, volume 11138 of *Lecture Notes in Computer Science*, pages 143–159. Springer, 2018.

[BST07] Clark W. Barrett, Igor Shikanian, and Cesare Tinelli. An abstract decision procedure for a theory of inductive data types. *J. Satisf. Boolean Model. Comput.*, 3(1-2):21–46, 2007.

[CH15a] Jürgen Christ and Jochen Hoenicke. Cutting the mix. In *CAV*, pages 37–52, 2015.

[CH15b] Jürgen Christ and Jochen Hoenicke. Weakly equivalent arrays. In *FROCOS*, pages 119–134, 2015.

[CH16] Jürgen Christ and Jochen Hoenicke. Proof tree preserving tree interpolation. *J. Autom. Reasoning*, 57(1):67–95, 2016.

[DDA09] Isil Dillig, Thomas Dillig, and Alex Aiken. Cuts from proofs: A complete and practical technique for solving linear inequalities over integers. In *CAV*, pages 233–247, 2009.

[DdM06] Bruno Dutertre and Leonardo de Moura. A fast linear-arithmetic solver for DPLL(T). In *CAV*, pages 81–94, 2006.

[dMB08] Leonardo de Moura and Nikolaj Bjørner. Model-based theory combination. *Electr. Notes Theor. Comput. Sci.*, 198(2):37–49, 2008.

[GdM09] Yeting Ge and Leonardo Mendonça de Moura. Complete instantiation for quantified formulas in satisfiabiliby modulo theories. In *CAV*, volume 5643 of *Lecture Notes in Computer Science*, pages 306–320. Springer, 2009.

[GHN+04] Harald Ganzinger, George Hagen, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. DPLL(T): Fast decision procedures. In *CAV*, volume 3114 of *Lecture Notes in Computer Science*, pages 175–188. Springer, 2004.

[HS18] Jochen Hoenicke and Tanja Schindler. Efficient interpolation for the theory of arrays. In *IJCAR*, volume 10900 of *Lecture Notes in Computer Science*, pages 549–565. Springer, 2018.

[HS19] Jochen Hoenicke and Tanja Schindler. Solving and interpolating constant arrays based on weak equivalences. In *VMCAI*, volume 11388 of *Lecture Notes in Computer Science*, pages 297–317. Springer, 2019.

[HS21] Jochen Hoenicke and Tanja Schindler. Incremental search for conflict and unit instances of quantified formulas with e-matching. In *VMCAI*, volume 12597 of *Lecture Notes in Computer Science*, pages 534–555. Springer, 2021.

[KBD13] Tim King, Clark Barrett, and Bruno Dutertre. Simplex with sum of infeasibilities for SMT. In *FMCAD*, pages 189–196. IEEE, 2013.

[NO05] Robert Nieuwenhuis and Albert Oliveras. Proof-producing congruence closure. In *RTA*, pages 453–468. Springer, 2005.

[PG86] David A. Plaisted and Steven Greenbaum. A structure-preserving clause form translation. *J. Symb. Comput.*, 2(3):293–304, 1986.

[RBF18] Andrew Reynolds, Haniel Barbosa, and Pascal Fontaine. Revisiting enumerative instantiation. In *TACAS (2)*, volume 10806 of *Lecture Notes in Computer Science*, pages 112–131. Springer, 2018.