Yices 2 in SMT-COMP 2020

Bruno Dutertre, Dejan Jovanović,
Ian A. Mason, Stéphane Graham-Lengrand

Computer Science Laboratory, SRI International

Introduction

Yices 2 is an open-source SMT solver developed and distributed by SRI International. It is available for download at `http://yices.csl.sri.com` and on our GitHub repository at `https://github.com/SRI-CSL/yices2`. Yices 2 supports linear and non-linear arithmetic, bit-vectors, uninterpreted functions, and arrays.

Yices 2 relies on the standard CDCL(T) architecture and uses a variant of the Nelson-Oppen method for combining decision procedures. Details are presented in [1]. Yices 2 also includes a solver that implements the Model-Construction Satisfiability Calculus (MC-SAT) [4, 5]. By default, MC-SAT is used for all theories that require non-linear arithmetic and CDCL(T) is used for everything else. Yices 2 is mostly focused on quantifier-free theories, but it supports a limited form of quantifier reasoning known as exists/forall solving [2].

Competition Version

In the 2020 SMT competition, we are entering the latest development version of Yices 2.6.2, in all the logics and divisions it supports, including the incremental, model-validation, and unsat-core tracks.

Compared to the version that we entered last year, this year's version includes support for a new backend SAT solver for bit-vector problems, new rewriting and simplifications, and the use of interval analysis in non-linear arithmetic problems. We have also fixed many bugs. We have made many improvements to the MC-SAT solver for bit-vectors [3], but competition rules prevent us from entering this MC-SAT solver in 2020.

For the QF_BV logic, Yices 2 now supports four backend solvers:

Armin Biere's CaDiCaL We use version `1.2.1` from the master branch of CaDiCaL's git repository `https://github.com/arminbiere/cadical`

Mate Soos's Cryptominisat [6] We use a fork of Cryptominisat 5 that provides a new C API. The main Cryprominisat repository is at `https://github.com/msoos/cryptominisat` and our fork is at `https://github.com/BrunoDutertre/cryptominisat`.

Our own improved CDCL-based solver. This SAT solver implements known techniques from the literature (e.g., variable elimination and other forms of preprocessing, modern restart heuristics, and LBD-based estimates of clause quality).

Armin Biere's Kissat This is a recent solver similar to CaDiCaL but implemented in C instead of C++. The source code for this solver is at `http://fmv.jku.at/kissat/`.

In the competition, we picked Kissat as backend solver for the single-query track in logic QF_BV. In all other logics, we run Yices 2 with its default configuration.

Acknowledgment

Recent Yices developments were supported by the Defense Advance Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center Pacific (SSC Pacific) under Contract No. N66001-18-C-4011. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or SSC Pacific

References

[1] Bruno Dutertre. Yices 2.2. In Armin Biere and Roderick Bloem, editors, *Computer-Aided Verification (CAV'2014)*, volume 8559 of *Lecture Notes in Computer Science*, pages 737–744. Springer, July 2014.

[2] Bruno Dutertre. Solving exists/forall problems with yices. In *13th International Workshop on Satisfiability Modulo Theories (SMT 2015)*, July 2015.

[3] Stéphane Graham-Lengrand, Dejan Jovanović, and Bruno Dutertre. Solving bitvectors with MCSAT: explanations from bits and pieces. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Proceedings of the 10th International Joint Conference on Automated Reasoning (IJCAR'20)*, Lecture Notes in Computer Science. Springer-Verlag, July 2020. Accepted for publication.

[4] Dejan Jovanović, Clark Barrett, and Leonardo de Moura. The design and implementation of the model model constructing satisfiability calculus. In *Formal Methods in Computer-Aided Design (FMCAD)*, pages 173–180. IEEE, October 2013.

[5] Dejan Jovanović and Leonardo de Moura. Solving non-linear arithmetic. In *International Joint Conference on Automated Reasoning*, pages 339–354. Springer Berlin Heidelberg, 2012.

[6] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In *Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings*, pages 244–257, 2009.