SONOLAR SMT-Solver

Florian Lapschies, Jan Peleska, Elena Gorbachuk and Tatiana Mangels
University of Bremen, FB3/AGBS, 28359 Bremen, Germany
{florian,jp,elenav,tatiana}@informatik.uni-bremen.de

SONOLAR is an SMT-solver for the theory of bit-vectors and for bit-vectors with arrays. It uses bit-blasting to translate problems to SAT. To this end, the input formula is first represented as a directed acyclic formula graph on which word-level rewrite rules, normalization and variable substitution are applied. The result is then translated to an And-inverter-Graph (AIG) which allows to apply bit-level rewrite rules. After translation to CNF the problem is finally fed into MiniSat 2.2[2].

In order to reason about arrays we adopted the approach described in [1] which over-approximates reads, writes and equalities on arrays and incrementally adds *lemmas on demand* until the SAT solver produces a consistent model or decides the problem as unsatisfiable.

In addition to the standard bit-vector operations SONOLAR also supports IEEE-754 floating point operations, including all rounding modes, that are bit-blasted to SAT as well.

In comparison with last years version the solver received the following improvements:

- Many new rewrite-rules have been added.
- The code for applying substitutions has been rewritten to apply all current substitutions in one rewrite pass.
- A new library for performing multiple precision modular arithmetic more efficiently by using word-level arithmetic has been developed.
- The AIG implementation has been rewritten to only submit nodes to the SAT solver that are relevant for deciding satisfiability. After solving, the model of the SAT solver is then extended as needed.
- The SMTLIB2 front-end now supports the *get-value* and *get-assignment*-commands, the *:named*-attribute and the definition of interpreted functions.
- A rich C++ interface has been added to use the solver as a library.

Moreover – this is currently work in progress and not activated for the competition – SONOLAR performs constraint propagation on the formula graph using two domains, bit vector intervals and symbolic bit vectors. In our symbolic bit vector approach each individual bit of a vector is a member of a set of symbolic bits, where they optionally may be inverted. For example, the set $\{b_0, b_1, \neg b_2\}$ is interpreted as constraint $b_0 = b_1 \wedge b_2 = \neg b_1$. While initially every bit is contained in its own set, these sets successively get merged during

constraint propagation. Propagation is performed in a forward-backward fashion until a fix-point is reached or an inconsistency has been found. Finally this information is used to generate a more efficient SAT encoding.

The solver is currently used for automated test data generation in model based testing [7, 6, 5] as well as for C/C++ programs [4, 3].

The latest version of SONOLAR is available at `http://www.informatik.uni-bremen.de/~florian/sonolar/`. Binary releases are available for Linux and Windows and may be used for research and evaluation purposes in an academic environment.

References

[1] Robert Brummayer and Armin Biere. Lemmas on Demand for the Extensional Theory of Arrays. In *Proc. 6th Intl. Workshop on Satisfiability Modulo Theories (SMT'08)*, New York, NY, USA, 2008. ACM.

[2] Niklas Eén and Niklas Sörensson. An Extensible SAT-solver. In *SAT*, pages 502–518, 2003.

[3] Jan Peleska. Integrated and automated abstract interpretation, verification and testing of C/C++ modules. In Dennis R. Dams, Ulrich Hannemann, and Martin Steffen, editors, *Correctness, Concurrency and Compositionality – Festschrift for Willem-Paul de Roever*, LNCS Festschrift series. Springer, 2008. To appear.

[4] Jan Peleska. A unified approach to abstract interpretation, formal verification and testing of c/c++ modules. In John S. Fitzgerald, Anne E. Haxthausen, and Husnu Yenigun, editors, *Theoretical Aspects of Computing - ICTAC 2008, 5th International Colloquium*, volume 5160 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2008.

[5] Jan Peleska, Artur Honisch, Florian Lapschies, Helge Löding, Hermann Schmid, Peer Smuda, Elena Vorobev, and Cornelia Zahlten. Embedded systems testing benchmark, 2011. http://www.mbt-benchmarks.org.

[6] Jan Peleska, Artur Honisch, Florian Lapschies, Helge Löding, Hermann Schmid, Peer Smuda, Elena Vorobev, and Cornelia Zahlten. A real-world benchmark model for testing concurrent real-time systems in the automotive domain. In Burkhart Wolff and Fatiha Zaidi, editors, *Testing Software and Systems. Proceedings of the 23rd IFIP WG 6.1 International Conference, ICTSS 2011*, volume 7019 of *LNCS*, pages 146–161, Heidelberg Dordrecht London New York, November 2011. IFIP WG 6.1, Springer.

[7] Jan Peleska, Elena Vorobev, and Florian Lapschies. Automated test case generation with SMT-solving and abstract interpretation. In Mihaela Bobaru, Klaus Havelund, Gerard J. Holzmann, and Rajeev Joshi, editors, *Nasa Formal Methods, Third International Symposium, NFM 2011*, volume 6617 of *LNCS*, pages 298–312, Pasadena, CA, USA, April 2011. Springer.