SMT-COMP: Satisfiability Modulo Theories Competition

Clark Barrett[1], Leonardo de Moura[2], and Aaron Stump[3]

[1] Department of Computer Science
New York University
[2] Computer Science Laboratory
SRI International
[3] Department of Computer Science and Engineering
Washington University in St. Louis

1 Introduction

Decision procedures for checking satisfiability of logical formulas are crucial for many verification applications (e.g., [2, 6, 3]). Of particular recent interest are solvers for Satisfiability Modulo Theories (SMT). SMT solvers decide logical satisfiability (or dually, validity) with respect to a background theory expressed in classical first-order logic with equality. Background theories useful for verification are supported, like equality and uninterpreted functions (EUF), real or integer arithmetic, and theories of bitvectors and arrays. Input formulas are often syntactically restricted; for example, to be quantifier-free or to involve only *difference constraints*. Some solvers support a combination of theories, or quantifiers.

The Satisfiability Modulo Theories Competition (SMT-COMP) is intended to spark further advances in the SMT field, especially for applications in verification. Public competitions are a well-known means of stimulating advancement in automated reasoning. Examples include the CASC Competition for first-order reasoning, the SAT Competition for propositional reasoning, and the Termination Competition for checking termination of term rewriting systems [4, 1, 7]. Significant improvements in tool capabilities are reported from year to year, which anecdotal evidence suggests the competitions play a strong role in fueling. The primary goals of SMT-COMP at CAV 2005 are:

- To spur development of SMT solver implementations.
- To collect benchmarks in a common format, namely the SMT-LIB format [5].
- To jump start definition of SMT theories, again using the proposed SMT-LIB format.
- To connect implementors of SMT solvers with potential users in the verification community.

The idea of holding SMT-COMP came out of discussions of the SMT-LIB initiative at the 2nd International Workshop on Pragmatics of Decision Procedures in Automated Reasoning (PDPAR) at IJCAR 2004. SMT-LIB is an initiative of the SMT community to build a library of SMT benchmarks in a proposed standard format. SMT-COMP aims to serve this goal by contributing collected benchmark formulas used for the competition to the library, and by providing an incentive for implementors of SMT solvers to support the SMT-LIB format.

Evaluation of SMT solvers entered in SMT-COMP takes place July 6-10, while CAV 2005 is meeting, in the style of CASC [4]. Intermediate results are posted periodically as SMT-COMP proceeds, and final results are announced on the last day of CAV. The local organizers have arranged for SMT-COMP to have exclusive access to a group of GNU Linux machines, which are used to run the competition.

The SMT organizers wish to thank Cesare Tinelli and Silvio Renise for developing the SMT-LIB format and theory specifications for SMT-COMP. Also to be thanked are Sriram Rajamani and Kousha Etessami for helping make SMT-COMP possible at CAV 2005. Finally, thanks go to everyone contributing benchmarks or entering solvers to SMT-COMP, and the entire SMT community for supporting the competition.

2 Rules and Competition Format

This Section presents a summary of the rules and competition format for SMT-COMP. These draw substantially on ideas from the design and organization of CASC [4]. More detailed information can be found on the SMT-COMP web site: `http://www.csl.sri.com/users/demoura/smt-comp/`

2.1 Entrants

An entrant to SMT-COMP is an SMT solver submitted in either source code or binary format to the organizers. The organizers reserve the right to submit their own systems, or other systems of interest, to the competition. For solvers submitted in source code form, the organizers take reasonable precautions to ensure that the source code is not viewed by anyone other than the organizers. Submitters of an SMT-COMP entrant are encouraged to be physically present at SMT-COMP, but are not required to be so to participate or win. The organizers commit to making reasonable efforts to install each system, but they reserve the right to reject an entrant if its installation process proves overly difficult. Finally, an entrant to SMT-COMP must include a short (1-2 pages) description of the system.

2.2 Execution of Solvers

Each SMT-COMP entrant, when executed, must read a single input formula presented on its standard input channel. All formulas are given in the concrete syntax of the SMT-LIB format, version 1.1 [5]. For its given input formula, each SMT-COMP entrant is expected to report on its standard output channel whether the formula is satisfiable or unsatisfiable. An entrant may also report "`unknown`" to indicate that it cannot determine satisfiability of the formula. Each SMT-COMP solver is executed on an unloaded competition machine for each given formula, up to a fixed time limit. This limit is yet to be determined, but expected to be at least 5 minutes.

2.3 Judging and Scoring

Scoring is done using the system of points and penalties in Figure 1. In recognition of the greater difficulty of achieving completeness than soundness in SMT systems, smaller penalties are assessed for incompleteness than for unsoundness. The organizers take responsibility for determining in advance whether formulas are satisfiable or not. In the event of a tie in total number of points, the solver with the lower average CPU time on formulas for which it did not timeout is considered the winner.

Reported	Points for correct response	Penalty for incorrect response
`unsat`	+1	-8
`sat`	+1	-4
`unknown`	0	0
timeout	0	0

Fig. 1. Points and Penalties

2.4 Problem Divisions

Each SMT-COMP problem division consists of well-sorted formulas in SMT-LIB format version 1.1. Divisions and the corresponding theories are defined in SMT-LIB format on the SMT-LIB web page (linked from SMT-COMP's page). The divisions contain a range of problems from relatively easy to difficult. Benchmark formulas for the divisions have been collected by the organizers from other researchers in the field, mostly from verification applications. The organizers reserve the right to cancel a division if there are too few solvers entered or benchmarks collected. For more detailed information on the divisions,

see the SMT-COMP web page. The prefix "QF_" below means the formulas in the division are quantifier-free, and in some cases there are pairs of divisions for integers and reals, respectively.

- QF_UF: uninterpreted functions
- QF_IDL (QF_RDL): integer (real) difference logic
- QF_UFIDL: integer difference logic with uninterpreted functions
- QF_LIA (QF_LRA): linear integer (real) arithmetic
- QF_UFLIA (QF_UFLRA): linear integer (real) arithmetic with uninterpreted functions
- QF_A: non-extensional arrays
- QF_AUFLIA: linear integer arithmetic with uninterpreted functions, arrays
- AUFLIA: linear integer arithmetic with uninterpreted functions, arrays, quantifiers

2.5 Proofs and Models

SMT-COMP recognizes entrants which produce suitable evidence for the results they report. Entrants which can produce proofs for unsatisfiable formulas are recognized as proof-producing, and entrants which can produce models for satisfiable formulas are recognized as model-generating. No award other than this recognition is given on the basis of such capabilities, and such capabilities are strictly optional for SMT-COMP entrants.

References

1. D. Le Berre and L. Simon. The essentials of the SAT 2003 competition. In *Sixth International Conference on Theory and Applications of Satisfiability Testing*, volume 2919 of *LNCS*, pages 452–467. Springer-Verlag, 2003.
2. H. Jin, H. Han, and F. Somenzi. Efficient Conflict Analysis for Finding All Satisfying Assignments of a Boolean Circuit. In L. Zuck and N. Halbwachs, editors, *11th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2005.
3. S. Lerner, T. Millstein, and C. Chambers. Automatically Proving the Correctness of Compiler Optimizations. In R. Gupta, editor, *In ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2003. received best paper award.
4. F.J. Pelletier, G. Sutcliffe, and C.B. Suttner. The Development of CASC. *AI Communications*, 15(2-3):79–90, 2002.
5. Silvio Ranise and Cesare Tinelli. The SMT-LIB standard, version 1.1, 2005. Available from the "Documents" section of http://combination.cs.uiowa.edu/smtlib.
6. M. Velev and R. Bryant. Effective Use of Boolean Satisfiability Procedures in the Formal Verification of Superscalar and VLIW Microprocessors. *Journal of Symbolic Computation*, 35(2):73–106, February 2003.
7. H. Zantema, 2005. personal communication.