ArgoLib v3.5: System Description for SMTComp'07*

Filip Marić
Faculty of Mathematics
University of Belgrade
Serbia
filip@matf.bg.ac.yu

Predrag Janičić
Faculty of Mathematics
University of Belgrade
Serbia
janicic@matf.bg.ac.yu

June, 2007.

ARGO-LIB v3.5 is a DPLL(T) based SMT solver developed at the Faculty of Mathematics, University of Belgrade. It includes a flexible and efficient SAT engine and solvers for several theories. This is the first time that ARGO-LIB enters the SMT competition. Since, ARGO-LIB v3.5 shows best results on linear arithmetic over reals, this year it will compete only in QF_LRA and QF_RDL categories.

1 System architecture and implementation

System is implemented in C++ programming language. The only third-party component that it uses is the GNU Multiple Precision Arithmetic Library.

This version of our system follows the DPLL(T) approach [6]. Main components of the system are:

Preprocessor. This module prepares input problems for solving. It parses benchmarks written in SMT-LIB format, performs boolean abstraction, eliminates if-then-else connectives, and performs CNF conversion.

SAT solver. DPLL(X) part of the system is based on our SAT solver called ARGO-SAT. ARGO-SAT is a rational reconstruction of the MiniSat solver [3]. The main design goal of ARGO-SAT was to improve code readability and flexibility of MiniSat without affecting its performance. This makes experimenting in the field of SAT research easier, and also enables us to formally prove the correctness of the solver, which is another goal of our research. We believe that we have successfully achieved this design goal, and some parts of ARGO-SAT are already formally verified. ARGO-SAT supports most state-of-the-art SAT techniques including two-watched-literals propagation scheme, non-chronological backtracking, conflict analysis and learning, VSIDS like literal selection scheme, etc. In order to use ARGO-SAT as an SMT solver engine, several adjustments had to be made. Solver is made online, and theory propagation is enabled.

Theory solvers. ARGO-LIB supports solvers for several theories, and even Nelson-Oppen combination scheme, but, in the current version, only the rational linear

*This work was partially supported by Serbian Ministry of Science and Technology grant 144030.

arithmetic solver is implemented in incremental, state-of-the-art manner. In fact, ARGO-LIB v.3.5 supports two different solvers for QF_LRA. One is based on Fourier-Motzkin elimination and the other is based on Yices Simplex algorithm [2]. Although the Fourier-Motzkin based procedure contains many nontrivial improvements and efficient implementation techniques, the Simplex based procedure performs much better in practice. Still, if the rules of the competition and available computing resources of allow it, we would like to try both solvers. In order to improve performance, equality reasoning is separated into a specialized solver that cooperates with QF_LRA solvers. This solver for the fragment of equality theory is based on the union-find data structure [7].

2 Expected performance

ARGO-LIB will compete in the following divisions: QF_LRA, QF_RDL. Since, this is the first time that we enter the competition, we do not have high expectations. The main motive for our participation in this competition is not to try to win, but to gain experience and learn. At this point, our goal is to show that we have built a system that can solve several nontrivial industrial benchmarks. We hope that our system will correctly solve most easy problems (difficulty 0, 1, 2), and maybe even a few harder ones (difficulty 3, 4, 5) within the given 20 minutes time limit. Since we use the same procedure for both linear arithmetic and difference logic, ARGO-LIB will perform much better on QF_LRA than on QF_RDL.

Seed number: 63432546

References

[1] M. Bozzano et al. MathSAT: Tight Integration of SAT and Mathematical Decision Procedures. *Satisfiability Research in 2005*, pages 265–293.

[2] B. Dutertre, L. de Moura. A Fast Linear-Arithmetic Solver for DPLL(T). *CAV 2006*, pages 81–94, 2006.

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

[4] F. Marić and P. Janičić. ARGO-LIB: A generic platform for decision procedures. *IJCAR*, volume 3097 of *LNCS*, pages 213–217. Springer, 2004.

[5] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S .Malik. Chaff: engineering an efficient sat solver. In *DAC '01*, pages 530–535, 2001.

[6] R. Nieuwenhuis, A. Oliveras, and C. Tinelli. Solving SAT and SAT Modulo Theories: from an Abstract Davis-Putnam-Logemann-Loveland Procedure to DPLL(T). *Journal of the ACM*, 53(6):937–977, 2006.

[7] R. Nieuwenhuis, A. Oliveras. Fast Congruence Closure and Extensions. *Inf. Comput.*, 4: 557-580, 2007.