Journal on Satisfiability, Boolean Modeling and Computation 5 (2008)

Editor's Introduction to the Special Volume on Application of Constraints to Formal Verification

Miroslav N. Velev miroslav.velev@aries-da.com
*Aries Design Automation,
Chicago, IL,
U.S.A.*

During the last eight years, tremendous progress was made in the field of Boolean Satisfiability (SAT). Now SAT solvers are 4 to 5 orders of magnitude faster, and can solve formulas that are 4 to 5 orders of magnitude bigger. SAT is the enabling technology for formal verification—the mathematical proof of correctness of computer systems. Statistics from industrial circuit designs indicate that up to 90% of the engineering effort is spent on verification, which increasingly becomes the bottleneck in developing new products. Formal verification, gaining wider acceptance in industry, has the potential to significantly reduce the design time, while also guaranteeing complete correctness and avoiding costly design bugs that can easily drive a company bankrupt. The seven regular papers and two research notes in this special volume present exciting work on applying SAT to formal verification and related domains.

In the first paper, entitled *Improved SAT-based Reachability Analysis with Observability Don't Cares*, Sean Safarpour and Andreas Veneris from the University of Toronto (Canada), and Rolf Drechsler from Bremen University (Germany) present a SAT-based method for reachability analysis. By accounting for observability don't cares—variables whose values do not affect the formula given the values of other variables—it was possible to achieve up to 4× speedup for unbounded model checking problems, and $1 - 2$ orders of magnitude reduction of trace sizes, thus simplifying the subsequent debugging.

The second paper, *Abstraction Refinement with Craig Interpolation and Symbolic Pushdown Systems*, is by Javier Esparza, Stefan Kiefer, and Stefan Schwoon from the Technical University of Munich (Germany). They studied how Craig interpolants can be computed efficiently in counterexample-guided abstraction refinement for software model checking. They proposed a new type of interpolant and showed how to treat multiple counterexamples in one refinement cycle, achieving exponential speedups.

The third paper, *Dependence Graph Based Verification and Synthesis of Hardware/ Software Co-Designs with SAT Related Formulation*, is by Masahiro Fujita, Kenshu Seto, and Thanyapat Sakunkonchak from the University of Tokyo (Japan). The authors describe verification and synthesis techniques based on the analysis of System Dependence Graphs by translating the problems to SAT and ILP. The experimental results indicate that the state-of-the-art SAT and ILP solvers can scale for reasonably large designs.

The fourth paper is entitled *Stressing Symbolic Scheduling Techniques within Aircraft Maintenance Optimization* and is by Viviana Bruno, Luz Garcia, Sergio Nocco, and Stefano Quer from the Polytechnic University of Turin (Italy). They developed and compared several scheduling techniques for aircraft maintenance based on, respectively, SAT solvers, Binary Decision Diagrams, Timed Automata, and Colored Petri Nets. The SAT-based approach proved to be faster and more scalable, and was able to solve instances 3 – 4 times larger than the rest of the approaches.

The fifth paper is *A Probabilistic and Approximated Approach to Circuit-Based Formal Verification* by Sergio Nocco and Stefano Quer from the Polytechnic University of Turin (Italy). The authors present a probabilistic approach for manipulating circuit-level formulas by using either an under-approximation or an over-approximation, as determined by the specific formal verification problem, in order to produce a more compact and expressive representation. They applied the approach to hard instances of bounded model checking and backward unbounded model checking, and observed up to an order of magnitude speedup.

The sixth paper, *QBF-Based Formal Verification: Experience and Perspectives*, is by Marco Benedetti from the University of Orleans (France), and Hratch Mangassarian from the University of Toronto (Canada). In the first part of the paper, they summarize techniques for Quantified Boolean Formulas (QBF) based formal verification. In the second part, they study the benefits from restricted quantifiers, QBF certificates, alternative encodings for classical model checking problems, and encodings with free variables. Their results include the first case studies where QBF compares favorably with SAT. QBF is even shown to outperform SAT in some tasks, such as automated debugging of large circuits. Furthermore, when the problem sizes increase, the SAT encodings run out of memory, while the more compact QBF encodings continue to be manageable and solvable.

The seventh paper is *Solving Partial Order Constraints for LPO Termination* by Michael Codish from the Ben-Gurion University (Israel), and Vitaly Lagoon and Peter J. Stuckey from the University of Melbourne (Australia). The authors introduce a propositional encoding for lexicographic path orders (LPOs) and the corresponding LPO termination property of term rewrite systems. Given this encoding, termination analysis can be performed using a SAT solver. The experimental results show orders of magnitude speedup relative to previous approaches for LPO termination.

The eighth paper, *A Resolution Based SAT-solver Operating on Complete Assignments*, is by Eugene Goldberg from the Cadence Research Labs in Berkeley (U.S.A.). He presents a SAT solver that operates on complete assignments, and that is competitive with the recent efficient SAT solvers MiniSat and BerkMin on large bounded model checking formulas. The benefit from operating on complete assignments is that a set of points expressing a resolution proof can be dramatically smaller than the entire search space.

The ninth and final paper is *Boosting SAT Solver Performance via a New Hybrid Approach* by Lei Fang and Michael S. Hsiao from Virginia Tech (U.S.A.). They combine a local-search-based SAT solver and a DPLL-based SAT solver. The local search identifies a set of clauses that are hard to satisfy and are then passed to the complete DPLL SAT solver, which if successful to solve them passes the solution back to the local-search solver, or otherwise identifies this set as an unsatisfiable core that implies that the original formula is unsatisfiable. The process is repeated: if the local-search SAT solver cannot solve the formula with the previous subset of clauses—which was solved by the DPLL SAT solver—restricted to the solution found for it, then at each iteration the previous subset of clauses is extended with at least one new clause, and the resulting subset of clauses is again passed to the DPLL SAT solver. This combination of the two kinds of solvers is proved to be a complete SAT procedure. Up to an order of magnitude speedup is achieved for satisfiable instances, with smaller speedups for unsatisfiable ones because of the overhead.

On behalf of the Editorial Board of the Journal on Satisfiability, Boolean Modeling and Computation, I thank the authors for their contributions, and the 51 reviewers for their insightful comments. There were 14 submissions of which 9 were accepted, such that each accepted paper went through between 3 and 6 rounds of review. Special thanks to Joao Marques-Silva for his exceptional help with the editing of this volume. The nine papers are representative of the recent exciting advances in the field of SAT and its applications to formal verification and related domains.