Preface to the special issue “SI: Satisfiability Modulo Theories”

Ofer Strichman · Daniel Kroening

Published online: 4 August 2012
© Springer Science+Business Media, LLC 2012

With this special issue we celebrate the tenth anniversary of the foundation of the *Satisfiability Modulo Theory* (SMT) community. We invited several members of this community to submit articles dedicated to this event, and were happy to see that all of them accepted the invitation and submitted. Each paper was reviewed by at least four expert reviewers. Five out of the six submitted papers were accepted.

SMT is probably the most successful academic community related to logics and verification that were built in the last decade. Indicators of this success are the annual 2-day SMT workshop, which typically attracts more participants than any other adjacent workshop, the SMT-COMP competition that steadily grows in number of submissions, the number of tools and benchmarks, the impact on related research areas, the industrial adoption, and the three text books [3, 6, 7] related to SMT that were published in the last few years.

The academic impact of SMT to date can be quantified by the number of articles it produces. The term “Satisfiability modulo theories”, which was born only ten years ago, has 1730 hits in Google Scholar as of July 2012, and there are probably hundreds of other papers that are associated with SMT but do not use this term explicitly. The impact of this field on the industry can be measured by its use of SMT solvers: Microsoft uses Z3, an SMT solver developed by de Moura and Bjorner, in at least 10 different program analysis tools; Intel is using SMT solvers such as MathSat and Boolector for processor verification and software verification; other companies that are known to use SMT solvers include Galois Connection, Praxis, GrammaTech, NVIDIA, Synopsys, Mathworks, Dassault Aviation, and the list continues. SMT solvers are now standard engines in numerous industrial applications, some of which (like scheduling) go beyond the scope of deductive reasoning and formal verification.

One can trace back the birth of this community to the breakthroughs in SAT solving in early 2001, and specifically to the introduction of the SAT solver Chaff by Moskewicz and others [8]. Almost immediately after its introduction several groups started investigating

O. Strichman (✉) · D. Kroening
Haifa, Israel
e-mail: ofers@ie.technion.ac.il

how its power can be harnessed for deciding first-order theories. Research in computational logic back then was almost exclusively carried out in the context of theorem provers, and decision procedures were rarely available as independent tools. All of these procedures dealt with disjunctions by (lazy) syntactic case splits. SAT contributed several new techniques to deal with the Boolean structure, most notably *learning*. The standard architecture of today's SMT solvers, namely the interplay between a SAT solver and a decision procedure for the conjunctive fragment of the theory, was born back then with the almost-simultaneous introduction of CVC [2, 10] by Stump, Barrett and Dill; ICS-SAT [5] by Filliatre, Owre, Ruess and Shankar; MATHSAT [1] by Audemard, Bertoli, Cimatti, Kornilowicz, and Sebastiani; DLSAT [9] by Mahfoudh, Niebert, Asarin, and Maler; and VERIFUN [4] by Flanagan, Joshi, Ou and Saxe. We are very happy to include in this issue the work of some of these pioneers.

The immediate success of the first SMT solvers led to a need to streamline the research and form a standard format and collect benchmarks—which back then was a major obstacle for further development of these tools. About the same time, at FroCoS 2002, A. Armando suggested to form such a standard and a web page for collecting benchmarks, which eventually led to SMT-LIB: a standard and a corresponding benchmark repository. The rest, as the saying goes, is history.

Ofer Strichman, Daniel Kroening, July 2012

References

1. Audemard G, Bertoli P, Cimatti A, Kornilowicz A, Sebastiani R (2002) A SAT based approach for solving formulas over boolean and linear mathematical propositions. In: Proc 18th international conference on automated deduction (CADE)
2. Barrett CW, Dill DL, Stump A (2002) Checking satisfiability of first-order formulas by incremental translation to SAT. In: Proc 14th intl conference on computer aided verification (CAV'02). LNCS, vol 2404. Springer, Berlin, pp 236–249
3. Bradley AR, Manna Z (2007) The calculus of computation—decision procedures with applications to verification. Springer, Berlin
4. Flanagan C, Joshi R, Ou X, Saxe JB (2003) Theorem proving using lazy proof explication. In: Proc CAV'03, vol 2725. Springer, Berlin, pp 355–367
5. Filliatre JC, Owre S, Rueb H, Shankar N (2001) ICS: integrated canonizer and solver. In: Berry G, Comon H, Finkel A (eds) Proc 13th intl conference on computer aided verification (CAV'01). LNCS. Springer, Berlin
6. Harrison J (2009) Handbook of practical logic and automated reasoning. Cambridge University Press, Cambridge
7. Kroening D, Strichman O (2008) Decision procedures—an algorithmic point of view. Theoretical computer science series. Springer, Berlin
8. Moskewicz M, Madigan C, Zhao Y, Zhang L, Malik S (2011) Chaff: engineering an efficient SAT solver. In: Proc design automation conference (DAC'01)
9. Mahfoudh M, Niebert P, Asarin E, Maler O (2002) A satisfiability checker for difference logic. In: Proceedings of 5-th international conference on theory and applications of satisfiability testing (SAT), pp 222–230
10. Stump A, Barrett C, Dill D (2002) CVC: a cooperating validity checker. In: Proc 14th intl conference on computer aided verification (CAV'02)