

Application of constraints to formal verification and artificial intelligence

Miroslav N. Velev · John Franco

Published online: 28 April 2014
© Springer International Publishing Switzerland 2014

We are excited to present the AMAI special issue on Application of Constraints to Formal Verification and Artificial Intelligence. The issue contains three papers:

In the first paper, entitled *Establishing Flight Software Reliability: Testing, Model Checking, Constraint-Solving, Monitoring and Learning*, Alex Groce from Oregon State University, Klaus Havelund, Gerard Holzmann, and Rajeev Joshi from the NASA Jet Propulsion Laboratory (JPL), and Ru-Gang Xu from the University of California at Los Angeles discuss the application of a range of techniques to the verification of mission-critical flight software at JPL. They present insights on the combination of random testing, model checking, constraint-solving, monitoring, and finite-state machine learning, in addition to static code analysis for the efficient verification of properties of a flash file system and other software for space missions.

In the second paper, entitled *Polynomial Function Intervals for Floating-Point Software Verification*, Jan Duracz from Halmstad University in Sweden and Michal Konecny from Aston University in the U.K. present a new method for verification of numerical software, supporting a substantially more expressive language for specifications than other publicly available automated tools. Their experiments indicate that the additional expressivity does not come at a performance cost when comparing the results with those obtained with other publicly available state-of-the-art provers. They also include a scalability study of their algorithm when proving tight functional specifications of progressively larger randomly generated programs.

In the third paper, entitled *Tradeoffs in the Complexity of Backdoors to Satisfiability: Dynamic Sub-Solvers and Learning During Search*, Bistra Dilkina from the Georgia Institute of Technology, Carla P. Gomes from Cornell University, and Ashish Sabharwal from IBM study the potential to capture succinctly hidden problem structure in Boolean Satisfiability (SAT) problems using different notions of backdoors—a set of key variables, such

M. N. Velev (✉)
Aries Design Automation, Chicago, IL, USA
e-mail: mvelev@gmail.com

J. Franco
University of Cincinnati, Cincinnati, OH, USA

that once they are instantiated the remaining problem simplifies to a tractable class. The authors show that backdoors with respect to syntactic tractable classes—such as Horn and 2CNF—are too weak, since these classes fail to capture key aspects of SAT solvers such as empty class detection, and that the notion of backdoors with respect to dynamic David-Logemann-Loveland (DLL) style propagation subsolvers can be strengthened even further by incorporating learning during search. They formally characterize the complexity and backdoor size effects of incorporating empty class detection into 2CNF and Horn and learning into DLL subsolvers, as well as empirically demonstrate that using DLL backdoors over Horn backdoors, combined with learning over traditional DLL backdoors, can result in an order of magnitude reduction of the backdoor sizes. The paper compares the relevance and strength of different backdoors notions, and points to the potential to exploit the existence of small DLL-style backdoors with learning in the design of state-of-the-art SAT solvers.

We thank the authors for their contributions, and the reviewers for their insightful comments,

Miroslav N. Velev, Aries Design Automation, Chicago, Illinois, U.S.A., December 2013

John Franco, University of Cincinnati, Cincinnati, Ohio, U.S.A., December 2013