Innovations Syst Softw Eng (2011) 7:71–72
DOI 10.1007/s11334-011-0153-4

SI: NFM 2010

Preface: special issue on NASA Formal Methods Symposium 2010

César A. Muñoz

Received: 5 May 2011 / Accepted: 10 May 2011 / Published online: 25 May 2011

The NASA Formal Methods Symposium (NFM) is an annual event intended to highlight the state of formal methods art and practice. It is a forum for theoreticians and practitioners from government, academia, and industry, with the goals of identifying challenges and providing solutions for achieving high assurance in safety-critical systems. Within NASA, for example, such systems include autonomous robots, separation assurance algorithms for aircraft, and autonomous rendezvous and docking systems for spacecraft. Moreover, emerging paradigms such as code generation and safety cases are bringing with them new challenges and opportunities. The focus of the symposium is on formal techniques, their theory, current capabilities, and limitations, as well as their application to aerospace, robotics, and other safety-critical systems.

This journal issue contains extended versions of the best papers presented at the Second NASA Formal Methods Symposium (NFM 2010), held at NASA Headquarters, in Washington D.C., USA, April 13–15, 2010.

The gap between tools used by industry and the verification techniques developed in academia is a major obstacle in the general adoption of formal methods technology. Closing this gap is the subject of two papers appearing in this special issue. Pritam Roy and Natarajan Shankar propose the SimCheck tool for checking contract specifications in Simulink, a commercial tool for modeling and simulating embedded systems. These contract specifications are written using an expressive type annotation system. From the Simulink models and their contract specifications, the tool generates verification conditions which can be discharged using static analysers. Dominic Richards and David Lester present a logical embedding of the hardware description language BlueSpec System Verilog (BSV). The embedding, which is specified in the Prototype Verification System (PVS), enables automated and interactive verification of BSV designs through PVS decision procedures and proof strategies.

George Box's quote "*all models are wrong, some are useful*" is particularly meaningful in formal verification. Paolo Arcaini et al. tackle the problem of automatically validating NuSMV models for properties such as consistency, completeness, and minimality. These properties are chosen since their violation typically signals model design errors.

Two novel protocol verification techniques are reported in this journal issue. Concetta Pilotto and Jerome White present a PVS verification framework for a class of distributed algorithms that solve systems of linear equations. The framework assumes a message-passing communication mechanism where messages can be lost, delayed, or delivered out-of-order. Xiaowan Huang et al. report on the verification of clock-synchronization protocols in UPPAAL, a timed automata model-checker. To overcome restrictions on the use of clocks in the timed automata formalism, the authors introduce a new type of values called integer clocks.

The formal analysis of critical cyber-physical systems requires fundamental advances in the state of the art in compositional verification, state-explosion mitigation heuristics, and floating-point verification, among many others areas. These are the issues addressed by the following papers. Assume-Guarantee is a well-known paradigm for reasoning about concurrent systems in a compositional way. Sagar Chaki and Arie Gurfinkel extend the learning-based assume guarantee paradigm to verify liveliness properties. The authors also propose learning algorithms for a class of systems that combines finite and infinite behavior. Yang Zhao and Gianfranco Ciardo describe how a saturation

C. A. Muñoz (✉)
NASA Langley Research Center, Hampton, VA 23681, USA
e-mail: cesar.a.munoz@nasa.gov

technique can be used to improve the performance of algorithms for computing strongly connected components and fair cycles in discrete-state models. In some cases, the improved algorithms make feasible computations that fail using previous approaches. Sylvie Boldo and Thi Minh Tuyen Nguyen propose an approach to estimate floating point errors independently of the architecture and compiler optimization strategy. The approach is implemented in a static analyzer for C and has been used automatically to prove the correctness of a critical component of an aircraft conflict resolution algorithm.