Journal of Automated Reasoning manuscript No.
(will be inserted by the editor)

Tests and Proofs

Preface of the Special Issue

Bernhard Beckert · Reiner Hähnle

Received: date / Accepted: date

To prove the correctness of a program is to demonstrate, through impeccable mathematical techniques, that it has no bugs; to test a program is to run it with the expectation of discovering bugs. The two techniques seem contradictory: if you have proved correctness of your program, it is fruitless to comb it for bugs; and if you are testing it, that is surely a sign that you have given up any hope to prove its correctness. Accordingly, proofs and tests have, since the onset of software engineering research, been pursued by distinct communities using rather different techniques and tools. And yet the development of both approaches leads to the discovery of common issues and to the realization that each may need the other. The emergence of model checking has been one of the first signs that contradiction may yield to complementarity, but in the past few years an increasing number of research efforts have encountered the need for combining proofs and tests, dropping earlier dogmatic views of incompatibility and taking instead the best of what each of these software engineering domains has to offer.

This special issue collects current advances in the ongoing attempt to obtain synergies from the combination of Tests and Proofs. It arose from the Second International Conference on Tests and Proofs that took place in Prato (near Florence), Italy, in April 2008. Our Call for Papers was addressed to the research community as a whole, but in addition we invited contributors to the conference to submit revised versions of their papers, and they have provided two of the five papers published here. All submissions were refereed to the standard of an archival journal.

The paper by Andrea Calvagna and Angelo Gargantini "A formal logic approach to constrained combinatorial testing" presents a novel approach to generating test suites for pairwise combinatorial testing (and k-wise combinatorial testing in an extended version). It uses logic-based techniques to control test construction. The method has been implemented, and an evaluation of the implemented tool is presented.

Bernhard Beckert
Department of Informatics, Karlsruhe Institute of Technology, Germany
E-mail: beckert@kit.edu

Reiner Hähnle
Department of Computer Science and Engineering, Chalmers University of Technology, Sweden
E-mail: reiner@chalmers.se

The paper by Andriy Dunets, Gerhard Schellhorn, and Wolfgang Reif "Automated flaw detection in algebraic specifications" describes the application of finite-model finding techniques for testing. They are used to identify bugs in simplification rules of abstract data types, which are used in the KIV software verification system.

The paper by Damiano Angeletti, Enrico Giunchiglia, Massimo Narizzano, Alessandra Puddu, and Salvatore Sabina entitled "Using bounded model checking for coverage analysis of safety-critical software in an industrial setting" presents a case study on using bounded model checking for test-case generation. Specifically, bounded model checking is employed as an automatic test generator with coverage analysis. It is applied to safety-critical embedded software that is part of the European Train Control System (ETCS).

The paper by Lydie Du Bousquet, Yves Ledru, Olivier Maury, Catherine Oriat, and Jean-Louis Lanet "Reusing a JML specification dedicated to verification for testing, and vice-versa: case studies" considers two case studies in which *both* testing and verification techniques are used, based on the same JML specifications of the underlying programs. For each case study a number of different testing *and* verification tools were considered. One interesting question for which this paper tries to find first answers is to what extent JML specifications are reusable for different systems and activities.

The paper by Delphine Longuet, Marc Aiguier, and Pascale Le Gall "Proof-guided test selection from first-order specifications with equality" is about the generation of test cases for systems that are specified by finite sets of axioms in classical first-order logic. An algorithm is given that can be executed mechanically, and works by refining an initial set of test cases using the axioms that specify the system.

Together, the five papers selected for this Special Issue demonstrate the wide range of possibilities where Tests and Proofs can benefit from each other: three of the papers use theorem-proving technology to improve the performance or the results of testing/test generation, which is perhaps the most natural and obvious thing to do. The paper by Dunets et al., however, highlights the fact that the formalisations which theorem provers work upon can benefit from systematic testing as much as programs do. Finally, the paper by Du Bousquet et al. asks the question of whether we can expect to use the same formalisation as a basis for both testing and proving.

These papers present first results in an exciting new field, and we are convinced that this is merely a (promising) start: we can expect much more to come from the combination of Tests and Proofs.

Karlsruhe and Gothenburg, April 2010

BERNHARD BECKERT and REINER HÄHNLE