TAP 2008
2nd International Conference on Tests and Proofs

How does deduction help testing? How does testing help deduction?
How can the combination of testing and deduction increase the reach of both?

April 9-11, 2008, Prato, Italy

Conference Chair
B. Meyer (ETH Zurich, Switzerland)

Program Co-Chairs
B. Beckert (U of Koblenz, Germany)
R. Hähnle (Chalmers U of Technology, Sweden)

Program Committee
B. Aichernig (TU Graz, Austria)
M. Butler (U of Southampton, UK)
T. Y. Chen (Swinburne U of Technology, Australia)
Y. Gurevich (Microsoft Research, USA)
W. Howden (U of California at San Diego, USA)
B. Meyer (ETH Zurich, Switzerland)
P. Müller (Microsoft Research, USA)
T. Nipkow (TU München, Germany)
Robby (Kansas State U, USA)
D. Rosenblum (U College London, UK)
W. Schulte (Microsoft Research, USA)
N. Sharygina (CMU & U of Lugano, Switzerland)
B. Wolff (ETH Zurich, Switzerland)

– more to be added –

Steering Committee
Y. Gurevich (Microsoft Research, USA)
B. Meyer (ETH Zurich, Switzerland)

Organizing Committee
C. Gladisch (U of Koblenz, Germany)
P. Rümmer (Chalmers U of Technology, Sweden)

CALL FOR PAPERS

The TAP conference is devoted to the convergence of proofs and tests. It combines ideas from both sides for the advancement of software quality.

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 your program, it’s fruitless to comb it for bugs; and if you are testing it, that is surely a sign that you have given up on 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.

How does deduction help testing? How does testing help deduction?
How can the combination of testing and deduction increase the reach of both?

The scope of TAP includes topics such as

• Generation of test data, oracles, or preambles by deductive techniques such as theorem proving, model checking, symbolic execution, constraint logic programming, etc.
• Generation of specifications by deduction
• Verification techniques combining proofs and tests
• Program proving with the aid of testing techniques
• Transfer of concepts from testing to proving (e.g., coverage criteria)
• Automatic bug finding
• Formal frameworks
• Tool descriptions and experience reports
• Case studies

Submissions should describe previously unpublished work (completed or in progress), including descriptions of research, tools, and applications. Papers must be formatted following the Springer LNCS guidelines and be at most 15 pages long. Submission of papers is via EasyChair at http://www.easychair.org/TAP2008/. The proceedings are planned to be published within Springer’s LNCS series. They will be available at the conference.

Conference Web Page: www.uni-koblenz.de/tap2008