

Tests and Proofs

Second International Conference, TAP 2008, Prato, Italy, April 9-11, 2008, Proceedings

This volume contains the research papers, invited papers, and abstracts of - torials presented at the Second International Conference on Tests and Proofs (TAP 2008) held April 9–11, 2008 in Prato, Italy. TAP was the second conference devoted to the convergence of proofs and tests. It combines ideas from both areas 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. On the surface, the two techniques seem contradictory: if you have proved 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 on any hope of proving 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 their incompatibility and taking instead the best of what each of these software engineering domains has to offer.

This volume contains the research papers, invited papers, and abstracts of - torials presented at the Second International Conference on Tests and Proofs (TAP 2008) held April 9–11, 2008 in Prato, Italy. TAP was the second conference devoted to the convergence of proofs and tests. It combines ideas from both areas 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. On the surface, the two techniques seem contradictory: if you have proved 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 on any hope of proving 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 their incompatibility and taking instead the best of what each of these software engineering domains has to offer.



53,49 €

49,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9783540791232

Medium: Buch

ISBN: 978-3-540-79123-2

Verlag: Springer Berlin Heidelberg

Erscheinungstermin: 31.03.2008

Sprache(n): Englisch

Auflage: 2008

Serie: Programming and Software Engineering

Produktform: Kartoniert

Gewicht: 324 g

Seiten: 193

Format (B x H): 155 x 235 mm

