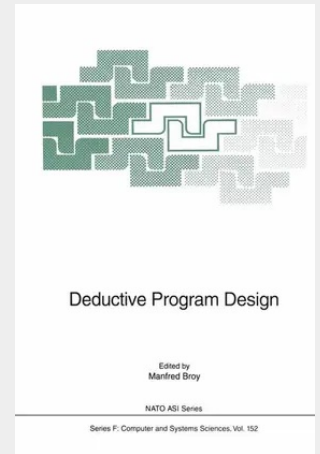


Deductive Program Design

Deductive program design aims at methods that guide and support the development of programs by techniques of deduction. Deduction is a well-known subject of study in logical theories. However, the application of deductive techniques in program design needs methods, heuristics, and understanding of the required forms and formats of the development result that go far beyond those found in logical theories. The summer school presents a very broad spectrum of approaches to these issues. Edsger W. Dijkstra gives a number of beautiful examples that show how to design proofs. Samson Abramsky, Simon Gay, and Rajagopal Nagarajan contribute a fundamental study founding concurrent programming on interaction categories and foundations of type theory. Tony Hoare presents unifying views on mathematical models for computing science. Unifying theories are especially important if the whole development process of programs is to be supported by deductive techniques. The rest of the contributions address applications of deductive program design methods. They are directed to different areas of application; for example Richard Bird's and Oege de Moor's algebra of programming is mainly intended for classical functional programming, and the contribution on data refinement and their transformation by David Gries considers sequential procedural programs.

Deductive program design aims at methods that guide and support the development of programs by techniques of deduction. Deduction is a well-known subject of study in logical theories. However, the application of deductive techniques in program design needs methods, heuristics, and understanding of the required forms and formats of the development result that go far beyond those found in logical theories. The summer school presents a very broad spectrum of approaches to these issues. Edsger W. Dijkstra gives a number of beautiful examples that show how to design proofs. Samson Abramsky, Simon Gay, and Rajagopal Nagarajan contribute a fundamental study founding concurrent programming on interaction categories and foundations of type theory. Tony Hoare presents unifying views on mathematical models for computing science. Unifying theories are especially important if the whole development process of programs is to be supported by deductive techniques. The rest of the contributions address applications of deductive program design methods. They are directed to different areas of application; for example Richard Bird's and Oege de Moor's algebra of programming is mainly intended for classical functional programming, and the contribution on data refinement and their transformation by David Gries considers sequential procedural programs.



213,99 €

199,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9783540609476

Medium: Buch

ISBN: 978-3-540-60947-6

Verlag: Springer

Erscheinungstermin: 18.06.1996

Sprache(n): Englisch

Auflage: 1996

Serie: NATO ASI Subseries F:

Produktform: Gebunden

Gewicht: 1880 g

Seiten: 470

Format (B x H): 157 x 235 mm

