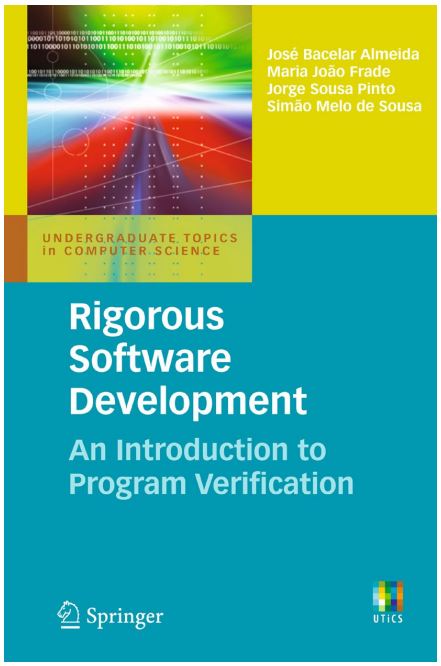

2011, XIII, 307 p. 52 illus.

Printed book

Softcover

34,95 € | £26.99 | $39.95
[1]37,40 € (D) | 38,45 € (A) | CHF 50,55

eBook

26,99 € | £20.99 | $29.99
[2]26,99 € (D) | 26,99 € (A) | CHF 40,00

Available from your library or springer.com/shop

MyCopy [3]

Printed eBook for just
€ | $ 24.99
springer.com/mycopy

Error[en_EN | Export.Bookseller.MediumType | SE]

J.B. Almeida, M.J. Frade, J.S. Pinto, S. Melo de Sousa

Rigorous Software Development

An Introduction to Program Verification

Series: Undergraduate Topics in Computer Science

- Self-contained, offering a concise introduction to formal methods together with an in-depth coverage of model-based and Hoare logic-based methods
- Focuses on two approaches: the Coq proof assistant and the B suite, both of which have proved effective and relevant for industry
- Includes exercises and solutions making it suitable as a course text and for self-study

The use of mathematical methods in the development of software is essential when reliable systems are sought; in particular they are now strongly recommended by the official norms adopted in the production of critical software. Program Verification is the area of computer science that studies mathematical methods for checking that a program conforms to its specification. This text is a self-contained introduction to program verification using logic-based methods, presented in the broader context of formal methods for software engineering. The idea of specifying the behaviour of individual software components by attaching contracts to them is now a widely followed approach in program development, which has given rise notably to the development of a number of behavioural interface specification languages and program verification tools. A foundation for the static verification of programs based on contract-annotated routines is laid out in the book. These can be independently verified, which provides a modular approach to the verification of software. The text assumes only basic knowledge of standard mathematical concepts that should be familiar to any computer science student. It includes a self-contained introduction to propositional logic and first-order reasoning with theories, followed by a study of program verification that combines theoretical and practical aspects - from a program logic (a variant of Hoare logic for programs containing user-provided annotations) to the use of a realistic tool for the verification of C programs (annotated using the ACSL specification language), through the generation of verification conditions and the static verification of runtime errors.