1.

Record Nr.	UNINA9910480311803321	
Autore	Back Ralph-Johan	
Titolo	Refinement Calculus [[electronic resource]] : A Systematic Introduction / / by Ralph-Johan Back, Joakim Wright	
Pubbl/distr/stampa	New York, NY : , : Springer New York : , : Imprint : Springer, , 1998	
ISBN	1-4612-1674-5	
Edizione	[1st ed. 1998.]	
Descrizione fisica	1 online resource (XVI, 520 p. 11 illus.)	
Collana	Texts in Computer Science, , 1868-0941	
Disciplina	005.1/4	
Soggetti	Computers	
	Theory of Computation	
Lingua di pubblicazione	Inglese	
Formato	Materiale a stampa	
Livello bibliografico	Monografia	
Note generali	Bibliographic Level Mode of Issuance: Monograph	
Nota di bibliografia	Includes bibliographical references and index.	
Nota di contenuto	1 Introduction -- 1.1 Contracts -- 1.2 Using Contracts -- 1.3 Computers as Agents -- 1.4 Algebra of Contracts -- 1.5 Programming Constructs -- 1.6 Specification Constructs -- 1.7 Correctness -- 1.8 Refinement of Programs -- 1.9 Background -- 1.10 Overview of the Book -- I Foundations -- 2 Posets, Lattices, and Categories -- 3 Higher-Order Logic -- 4 Functions -- 5 States and State Transformers -- 6 Truth Values -- 7 Predicates and Sets -- 8 Boolean Expressions and Conditionals -- 9 Relations -- 10 Types and Data Structures -- II Statements -- 11 Predicate Transformers -- 12 The Refinement Calculus Hierarchy -- 13 Statements -- 14 Statements as Games -- 15 Choice Semantics -- 16 Subclasses of Statements -- 17 Correctness and Refinement of Statements -- III Recursion and Iteration -- 18 Well-founded Sets and Ordinals -- 19 Fixed Points -- 20 Recursion -- 21 Iteration and Loops -- 22 Continuity and Executable Statements -- 23 Working with Arrays -- 24 The N-Queens Problem -- 25 Loops and Two-Person Games -- IV Statement Subclasses -- 26 Statement Classes and Normal Forms -- 27 Specification Statements -- 28 Refinement in Context -- 29 Iteration of Conjunctive Statements -- References.	
Sommario/riassunto	Much current research in computer science is concerned with two questions: is a program correct? And how can we improve a correct program preserving correctness? This latter question is known as the refinement of programs and the purpose of this book is to consider	

these questions in a formal setting. In fact, correctness turns out to be a special case of refinement and so the focus is on refinement. Although a reasonable background knowledge is assumed from mathematics and CS, the book is a self-contained introduction suitable for graduate students and researchers coming to this subject for the first time. There are numerous exercises provided of varying degrees of challenge.