Solving Higher-Order Equations From Logic to Programming

by Christian Prehofer

Progress in Theoretical Computer Science Series
Birkhäuser Boston, 1998, 186 p.
ISBN 0-8176-4032-0

The book under reviewing starts from the idea that equational logic is a fundamental and natural concept in computer science and mathematics. The author develops techniques for equational reasoning in higher-order logic.

The book consists of **nine Chapters**, a very consistent bibliography and an index of terms.

Chapter 1 (*Introduction*) declares the aim of the book- to present calculi for equational reasoning modulo higher-order equations as rewrite rules. Building on equational logic, the book works for the integration and generalization of declarative programming models (functional and logic) in higher-order logic. The author examines higher-order unification, undecidable in general, and tries to establish decidable situations.

Chapter 2 (*Preview*) gives an informal outline of the work, introducing main concepts and contributions, proceeding from first-order term rewriting and narrowing on higher-order unification and higher-order narrowing.

Chapter 3 (*Preliminaries*) contains a brief introduction to λ-calculus and introduces basic definitions and results for higher-order equational reasoning, concerning abstract reductions and termination, higher-order types and terms, position in λ-terms, substitution, unification, higher-order patterns.

Chapter 4 (*Higher-Order Equational Reasoning*) deals with higher-order unification and term rewriting. A set of transformation rules for full higher-order pre-unification is introduced and a special case of higher-order unification is presented, unification of higher-order patterns, that similarly proceeds on first-order unification. Some important confluence criteria for first-order rewriting are lifted to the higher-order case.

Chapter 5 (*Decidability of Higher-Order Unification*) examines decidability of higher-order unification, developing decidable classes of second-order unification. The restriction is in that one term of the unification problem has to be linear. Also there are examined extensions of this decidability result (extending patterns by linear second-order terms, repeated second-order variables) and briefly mentioned problems open to future examination.

Chapter 6 (*Higher-Order Lazy Narrowing*) discusses the main topic of the book, higher-order lazy narrowing, a goal directed method for solving goals in a top-down manner. This Chapter shows the principles of this approach, shows completeness, deals with terminating higher-order rewrite systems, and introduces refinements: simplification via rewriting, deterministic eager variable elimination, restriction for functional-logic programming, incorporation of conditional equations.

Chapter 7 (*Variations on Higher-Order Narrowing*) discusses alternative approaches to solving higher-order equations by narrowing and problems generated due to higher-order case: general plain narrowing, flattening the terms to patterns adding constraints.

Chapter 8 (*Applications of Higher-Order Narrowing*) presents examples for higher-order rewriting and narrowing to functional-logic programming (hardware synthesis, symbolic computation, a functional-logic parser, encryption, infinite data-structures and eager evaluation, functional difference lists, distributed systems with inherent non-determinism) and equational reasoning (program transformation, type inference).

Chapter 9 (*Concluding Remarks*) summarizes the materials presented in the book, compares them with related works on higher-order narrowing and functional-logic programming and considers further extensions of interest.

The book presents in a rather informal way the basic topics of the field of term rewriting (abstract reduction systems, termination, confluence, completion, and combination problems).

Rewrite systems are directed equations and allow for simplification of terms or expressions. In addition higher-order rewriting is very suitable for symbolic computation with complex structures in mathematics, programs, and programming languages.

The author's aim has been to present methods for higher-order equational reasoning for the integration of higher-order functional and logic programming. The underlying computation rule for this integration, called narrowing, is lifted to the higher-order case and, although undecidable in general, the book shows that the second-order unification as required by functional-logic programming is decidable.

The materials presented in the book form a new basis for higher-order functional-logic programming, possibly with conditions. But the applications of this approach go beyond current functional and logic programming languages.

Although chapters make good introduction to the domain, preliminary knowledge in logic, equational theory and rewriting systems is indispensable. The book is mainly a reference book for professional researchers and advanced students, and a good guide to the literature.

Anca Hotaran