

## Book Reviews

---

Roy L. Crole

### Categories for Types

Cambridge University Press, Cambridge 1993, pp. xvii, 335, ISBN 0-521-45701-7

This is a mathematical book on category theoretic treatment of (polymorphic) type theory, a subject with a strong connection to type systems of programming languages, in fact growing out of an attempt to design a comprehensive mathematical theory of such systems, but rooted also in independent motivations internal to mathematics proper. The book is written by an expert in the field, and is intended for a reader of considerable mathematical sophistication (or a will to achieve it) - the author aims the bulk of the book at British graduate students in mathematics and computer science and at researchers in the field.

Technically, the book attempts to be self contained, starting with overview chapters on ordered sets and category theory. The latter chapter uses the former as a source of examples, and together they already form an advanced undergraduate course.

Chapter 3 develops the correspondence of algebraic theories and categories with finite products, laying down the formal framework for developments to follow.

Chapter 4 develops the correspondence of cartesian closed categories and simply typed lambda-calculus, including the technique of categorical gluing.

Chapter 5 deals with second order polymorphism and the correspondence of hyperdoctrines and indexed cartesian closed categories. Both PER models and domain theoretical models are treated.

Chapter 6 follows the structure of chapter 5, for higher order polymorphism, bringing the reader to a topic of current research.

Appropriately to its subject, this is a disciplined mathematical book defining precisely its concepts and proving meticulously its results. Well endowed with numerous well selected exercises, which push the reader to improving and extending his understanding, the book requires of the reader to work hard, but it may also bring him far in this topical subject on the borderline between pure mathematics and theoretical computer science.

*Dean Rosenzweig  
Faculty of Mechanical Engineering  
and Naval Architecture  
University of Zagreb  
Zagreb, Croatia*

Leslie B. Wilson, Robert G. Clark

### Comparative Programming Languages

Second edition, Addison-Wesley Publishing Company, Inc., Reading, 1993, pp. xii, 400, ISBN 0-201-56885-3

This is one of several "a little on everything" books on programming languages on the market, which has in its first incarnation been there since 1988. It is intended as a textbook for a programming languages course, but also for practising computer programmers. This reviewer cannot hide his scepticism towards this kind of book (or course), since, to his taste, many things get mentioned but few (if any) can be properly explained, such explanations couldn't possibly fit one book. It also seems difficult to

assess the audience such books are good for: the programming novice is likely to get bewildered by the breadth of their scope, by the sheer number of concepts mentioned, while the programmer with some experience in several distinct languages will find some information on many things, but is likely not to find the answers to precisely those subtle questions that trouble him.

With the reviewer's bias towards its kind established, let us examine the book.

The main changes from the first edition are a fuller treatment of object-oriented languages, and an introduction to formal definition of syntax. Among other changes are increased emphasis on C and C++, introduction of ML as a main example of a functional language, introduction of lazy evaluation, and several revisions, updates and extensions.

The book has 14 chapters and 2 appendices. The introductory chapter and a historical survey from pre-Fortran times to the present, are followed by a chapter on simple types and their declaration, covering the typical Pascal-like offering.

A chapter on statements covers expressions, assignment and the usual set of control constructs, with a brief discussion of exception handling in Ada and Eiffel. Evaluation of expressions is assumed to have no side effects, except in case of function calls, which seems to be at odds with the declared emphasis on C and C++. Discussion of such side effects is deferred to the chapter on subprograms, and their use is condemned as "bad programming style" which "should be strongly discouraged".

The chapter on subprograms reviews the program structure of Fortran, Algol-like languages and C, with a brief overview of principal information passing mechanisms and storage management schemes.

The chapter on data structures presents essentially the Pascal data structures, with the concept of abstract data type deferred to the chapter on modules.

The chapter on modules explains the notions of module, abstract data type and object by exemplifying them, as usual, with stack and complex numbers.

The chapter on object-oriented languages reviews the notions of inheritance, overloading, information hiding and dynamic binding with an example from Eiffel.

The chapter on functional languages introduces lists (which would better fit the chapter on data structures), exemplifies the notion of "higher order function", and briefly presents Lisp, Logo, FP systems and ML, with a very brief (1 page) mention of lazy evaluation and infinite data structures.

The chapter on logic programming contains a brief (10 pages) overview of Prolog, without really mentioning its logical underpinnings, but with a discussion of its future.

The chapter on concurrency mentions Unix pipes, monitors, Ada rendezvous, Occam, guarded commands and real time programming, all in 19 pages.

The chapter on syntax and semantics is one of the principal novelties with respect to the first edition. A 7 pages section on (extended) BNF definitions is followed by a 1.5 pages section on denotational semantics and a page on axiomatic semantics.

Last two chapters discuss input and output and the future of programming languages.

Appendix 1 summarizes Ada, Algol 60, Algol 68, C, C++, Cobol, Eiffel, Fortran, Lisp, ML, Modula 2, Pascal, PL/1, Prolog, Smalltalk 80 according to 8 characteristics. Appendix 2 lists a few recommended references on each of those.

*Dean Rosenzweig  
Faculty of Mechanical Engineering  
and Naval Architecture  
University of Zagreb  
Zagreb, Croatia*