

Hindley, J. Roger; Seldin, Jonathan P.

Lambda-calculus and combinators, an introduction. 2nd revised and updated ed. (English)

Zbl 1149.03016

Cambridge: Cambridge University Press (ISBN 978-0-521-89885-0/hbk). xi, 345 p. (2008).

The lambda calculus and the theory of combinators are closely related theories developed by Church and Schönfinkel/Curry, respectively, as foundations for (higher-order) logic and function theory. These theories were the source of the first undecidability results and are now of great importance to computer scientists as bases for functional programming languages and proof assistance packages as well as for verification of programs.

The present book, by two of the most eminent workers in the area, is an updated and partly rewritten version of their earlier [*J. R. Hindley* and *J. P. Seldin*, Introduction to combinators and λ -calculus. London Mathematical Society Student Texts, 1. Cambridge etc.: Cambridge University Press (1986; Zbl 0614.03014)]. It provides an excellent introduction to the currently fashionable parts of the work for readers with some knowledge of predicate calculus and recursive functions. The new book has an easy to read LaTeX typesetting, more examples and exercises, an updated list of references and some more detailed notes on difficult or historically interesting points.

The main changes from the earlier book involve much expanded chapters on type theory, an area that has become much more prominent since 1986, including work on generalized type systems. Now missing are chapters on applications to the foundations of logic and the consistency of arithmetic.

I highly recommend this book!

Reviewer: [Martin W. Bunder \(Wollongong\)](#)

MSC:

- [03B40](#) Combinatory logic and lambda calculus
- [03-01](#) Introductory exposition (textbooks, tutorial papers, etc.) pertaining to mathematical logic and foundations
- [03-02](#) Research exposition (monographs, survey articles) pertaining to mathematical logic and foundations

Cited in **28** Documents

Keywords:

[combinator](#); [combinatory logic](#); [lambda calculus](#); [type theory](#)