
Bibliography

Annotated bibliography

Miranda

Turner D. *Miranda: A Non-Strict Functional Language with Polymorphic Types*

Turner D. *An Overview of Miranda*

The first papers to describe the Miranda system, written by its creator.

Research Software Limited, *Miranda On-line Reference Manual* On-line documentation, available with all Miranda implementations; this also contains details of additional features and built-in library functions.

Holyer I. *Functional Programming with Miranda* Presents an overview of Miranda and some theoretical and implementation issues.

Bird R. & Wadler P. *Introduction to Functional Programming* This book uses a notation similar to Miranda. Its emphasis is on mathematical examples and the algebraic manipulation of functions. It is suitable for a range of abilities from the mathematically-oriented beginner to the more advanced programmer.

History and future of functional programming

Backus J. *Can programming be liberated from the von Neumann style* A seminal paper which discusses the problems of basing programming languages on the Turing computational model; it then outlines a functional language known as FP.

Landin P. *The next 700 programming languages* Apart from having a great title, this paper is important for highlighting the relation between a language's written representation and its meaning. The proposed language ISWIM (*If you See What I Mean*) deals with expressions rather than procedural statements. As such it probably shares with LISP the genesis of modern functional programming languages.

Hudak P. *Conception, Evolution and Application of Functional Programming Languages* This excellent reference has a self-explanatory title!

Functional programming in general

Burge W. *Recursive Programming Techniques* Probably the first book to discuss the functional style of programming and still relevant. It also has several very useful parsing and sorting algorithms which demonstrate the functional programming style.

Darlington J., Henderson P. & Turner D. *Functional Programming and its applications* Contains a number of interesting articles from theoretical background to practical applications.

Eisenbach S. (ed.) *Functional Programming, languages, tools and architectures* A collection of introductory articles covering other Functional programming languages (HOPE and FP), practice, theory and implementation. Chapter 4 is of particular interest in that it shows that the functional style of programming can be used to good effect with a procedural language.

Field A. & Harrison P. *Functional Programming* An intermediate to advanced level book which uses Hope as the base functional language. It has a modern approach and covers a lot of ground, from an introduction to functional programming through to implementation techniques.

Glaser H., Hankin C. & Till D. *Principles of Functional Programming* One of the first general textbooks in this area; it also gives a gentle introduction to the lambda calculus and some related topics. The first chapter gives a language-independent example of functional program development.

Henson M. *Elements of Functional Languages* A general-purpose text with a treatment of program transformation.

MacLennan B. *Functional Programming Theory and Practice* This book introduces the practice and theory of functional language by using mathematical notation in preference to a particular Functional programming language. Discusses performance and implementation issues.

Myers C., Clack C. & Poon E. *Programming with Standard ML* Standard ML is a strict functional programming language, which contrasts to the lazy evaluation of Miranda. The book serves as a companion volume to this text, and should be useful to anyone wishing to learn the strict functional programming style.

Reade C. *Elements of Functional Programming* A relatively advanced textbook, using a language similar to SML. It has the purpose of covering functional programming in general; Chapters 4 and 6 give some extended examples of functional software. The book also introduces more theoretical issues, including: denotational semantics, type systems, lambda calculus and combinators and a discussion of implementation considerations.

Implementation of functional languages

- Diller A. *Compiling Functional Languages* An advanced book, providing an overview of a wide range of techniques used in Functional language compilers. Uses LispKit LISP as the base language.
- Henderson P. *Functional Programming Application and Implementation* One of the first books dealing with functional programming. It has some good examples of programming using higher order functions based on a purely functional subset of LISP (called LispKit). The book discusses how LispKit might be isolated from LISP. Since LISP is still widely used in the world of Artificial Intelligence, the use of a good functional style of programming in LISP is to be encouraged.
- Peyton Jones S. *The Implementation of Functional Programming Languages* Apart from dealing with the implementation of functional programming languages, this book also gives an overview of the lambda calculus and some other theoretical issues required for an understanding of implementations.
- Peyton Jones S. & Lester D. *Implementing Functional Languages* A very good practical book showing different implementation approaches using Miranda as the target language.

Functional programming and formal methods

- Barendregt H. *The Lambda Calculus: Its Syntax and Semantics*
- Michaelson G. *Functional Programming through Lambda Calculus*
- Revesz G. *Lambda-Calculus, Combinators and Functional Programming*
 The lambda calculus is a small but important mathematical language which forms the basis of understanding many of the theoretical issues of functional programming languages and of programming language theory in general.
 The first of these books is the standard reference book for the lambda calculus; it is more of a mathematical interest rather than being essential for a computer scientist. The second text is a gentle guide, whilst the third takes a more rigorous approach which leads to techniques for functional programming implementation.
- Thompson S. *Type Theory and Functional Programming* Constructive type theory is a formal system which covers both logic and functional programming. It can be used a programming language in its own right, a tool to develop functional programs and to reason about their correctness.

References

- Aho A., Hopcroft J. & Ullman J. (1974), *The Design and Analysis of Computer Algorithms*, Addison-Wesley.
- Backus J. (1978), *Can programming be liberated from the von Neumann style?*, Communications of the ACM, vol 21 no 8 pp 613-641.
- Barendregt H. (1984), *The Lambda Calculus: Its Syntax and Semantics*, North Holland.
- Bird R. & Wadler P. (1988), *Introduction to Functional Programming*, Prentice Hall.
- Brooks F. (1975), *The Mythical Man-month*, Addison-Wesley.
- Burge W. (1975), *Recursive Programming Techniques*, Addison-Wesley.
- Darlington J., Henderson P. & Turner D. (1982), *Functional Programming and its Applications*, Cambridge University Press.
- Diller A. (1988), *Compiling Functional Languages*, Wiley.
- Eisenbach S. (ed.), (1987) *Functional Programming, languages, tools and architectures*, Ellis-Horwood.
- Field A. & Harrison P. (1988), *Functional Programming*, Addison-Wesley.
- Glaser H., Hankin C. & Till D. (1984), *Principles of Functional Programming*, Prentice Hall.
- Gordon M. (1979), *The Denotational Description of Programming Languages*, Springer.
- Henderson P. (1980), *Functional Programming, Application and Implementation*, Prentice Hall.
- Henson M. (1987), *Elements of Functional Languages*, Blackwell Scientific Publications.
- Holyer I. (1991), *Programming with Miranda*, Pitman.
- Hudak P. (1989), *Conception, Evolution and Application of Functional Programming Languages*, ACM Computing Surveys, vol 21 no 3 pp 359-411.
- Hughes J. (1984), *Why Functional Programming Matters*, Report 16, Programming Methodology Group, University of Göteborg and Chalmers University of Technology.
- Kelley P. (1989), *Functional Programming for Loosely-coupled Multiprocessors*, Pitman.
- Kernighan B. & Pike R. (1984), *The UNIX Programming Environment*, Prentice Hall.
- Landin P. (1966), *The next 700 programming languages*, Communications of the ACM, vol 9 no 3 pp 157-164.
- MacLennan B. (1990), *Functional Programming Theory and Practice*, Addison Wesley.
- Michaelson G. (1989), *Functional Programming through Lambda Calculus*, Addison-Wesley.
- Minsky, M. (1967), *Computation: Finite and Infinite Machines*, Prentice Hall.
- Myers C., Clack C. & Poon E. (1993), *Programming with Standard ML*, Prentice Hall.

- Park S. & Miller K. (1988), *Random number generators: good ones are hard to find*, Communications ACM 31, pp 1192-1201.
- Peyton Jones S. (1987), *The Implementation of Functional Programming Languages*, Prentice Hall.
- Peyton Jones S. & Lester D. (1991), *Implementing Functional Languages*, Prentice Hall.
- Reade C. (1989), *Elements of Functional Programming*, Addison-Wesley.
- Revesz G. (1988), *Lambda-Calculus, Combinators and Functional Programming*, Cambridge University Press.
- Research Software Limited (1990), *Miranda On-line Reference Manual*.
- Standish T. (1980), *Data Structure Techniques*, Addison-Wesley.
- Thompson S. (1991), *Type Theory and Functional Programming*, Addison-Wesley.
- Turner D. (1976), *SASL Language Manual*, St. Andrews University, Technical Report.
- Turner D. (1979), *A New Implementation Technique for Applicative Languages*, Software Practice and Experience, vol 9, pp 31-49
- Turner D. (1982), *Recursion equations as a programming language*, in (Darlington *et al.*, 1982).
- Turner D. (1985a), *Functional Programs as Executable Specifications*, in Hoare C. & Shepherdson (eds), *Mathematical Logic and Programming Languages* Prentice Hall.
- Turner D. (1985b), *Miranda: A Non-Strict Functional Language with Polymorphic Types*, in Proceedings IFIP Conference on Functional Programming Languages and Computer Architecture, Springer Lecture Notes in Computer Science, vol 201.
- Turner D. (1986), *An Overview of Miranda*, in ACM SIGPLAN Notices, vol 21 no 12.