

Book reviews

The art of Prolog—second edition by Leon Sterling and Ehud Shapiro, MIT Press, Cambridge, MA 1994, pp. 509, £19.95 (paperback), £44.94 (hardback), ISBN 0-262-19338-8.

This is a revised and extended version of what is for many *the* standard text book on Prolog. The most immediately obvious difference with the original edition is the revised formatting, with more white-space and a clearer distinction between the fonts used for Prolog code and for text. Seemingly superficial, such things can make a big difference to the usability of a book, and the changes are successful here. More substantive changes include the removal of a number (hopefully all, but who would be so bold as to claim this!) of typographical errors, and best attempts to conform to the newly emerging Standard Prolog. A new chapter on program transformation has been added and the chapters on interpreters and logic grammars extensively revised.

The new chapter on program transformation is an important complement to the earlier chapter on program development. Together they provide some basic tools for making this transition from a writer of neat Prolog predicates to a writer of effective Prolog programs in the large. “Effective” means efficient, reliable and maintainable. Gone are the days when Prolog was slow. O’Keefe (1990) gives good demonstrations of this, and compiler technology has improved further since he wrote his book. Reliability is what systematic declarative program development can give you. Maintainability is a little under-sold in Sterling and Shapiro. Declarative programming can make for easily maintained code. However, the code still needs to be explicitly laid out and supported by effective commenting. These topics are not explicitly handled in this book to any great extent, but they are of vital importance (e.g., Dodd, 1990; Krause, 1995).

Sterling and Shapiro continues to evolve as the first, and largest, step on the way to a deep understanding of Prolog as a valuable tool for program development. Start here, and then read O’Keefe (1990) and Dodd (1990).

References

- Dodd, A, 1990, *Prolog: A Logical Approach*. Oxford University Press.
Krause, PJ, 1995. *Guidelines for the Development of Prolog Programs*. ICRF Technical Report St/ICRF/1800/14.
O’Keefe, RA, 1990. *The Craft of Prolog*. MIT Press.

Reviewed by Paul J. Krause, Imperial Cancer Research Fund, London, UK

Executable modal and temporal logics by M. Fisher and R. Owens, Springer-Verlag, Germany, 1995, 180 pp. DM 46.

This edited book contains updated and extended versions of the papers presented at the *Workshop on Executable Modal and Temporal Logics*. There is a trend in the field of computer science of using modal and temporal logics in the formalisation of various real-time systems, as well as in knowledge base representation. Naturally, there is a need for efficient execution of these kinds of representations. The current volume is a contribution towards this need.

The volume is a collection of ten papers, including one from the editors themselves introducing and surveying the field of modal and temporal logics. The other contributions start with the work of

Brzoska, who translates temporal logic programs with past and metric operators into a constraint logic program over a suitable algebra. This helps to use constraint logic programming as temporal logic programming. The next paper, by Kono, describes a tableau based implementation of an interval temporal logic. Fruhwirth introduces a family of logic programming languages for representing and reasoning about time by combining both qualitative and quantitative temporal reasoning with time points and periods. The paper lacks both formalisation and any implementation detail. An interesting paper is by Merz, who considers fixed-point temporal logics of linear time and defines a restricted class of safety properties that afford a linear model-construction algorithm.

MetateM is an executable temporal logic programming language developed by Gabbay and others. Concurrent MetateM extends MetateM by the introduction of concurrency and communications for implementing reactive systems. There are two papers in the volume on Concurrent MetateM. Fisher's paper describes a semantics of Concurrent MetateM, and the other paper by Reynold is on first-class Concurrent MetateM.

Koehler and Treinen have proposed yet another interval-based modal temporal logic LLL employing some conventional modal operators, including the Schops operator. The last two papers in the volume have practically nothing to do with temporal logic and little to do with mainstream modal logics. Although the title of the paper by Calvalcanti claims to have solved air-traffic problems with possible worlds, very little evidence is found in the paper itself. The concluding paper discusses the result of a study for representation of Dutch traffic law using deontic modalities.

Overall the papers in the volume are not well polished. The volume has a severe lack of papers describing languages suitable for representing real life problems and their efficient implementation. The area of modal and temporal logics has a lot to offer to the development of computer science. Too much emphasis on theoretical correctness of the logics and not enough attention on how to make them work may severely affect its progress, as in some other fields of artificial intelligence.

Reviewed by S. K. Das, Imperial College, London, UK

Artificial intelligence techniques in Prolog by Yoav Shoham, Morgan Kaufmann, 1994, 327 pp, £38.50, ISBN 1-55860-319-0.

This is another good book from Morgan Kaufmann, a fine textbook, and a suitable companion volume to Matt Ginsberg's *Essentials of Artificial Intelligence* from the same publisher. In this book, Shoham provides a very practical introduction to a large range of well-established methods. The coverage is extensive. Descriptions of the basic technique, such as the various types of search and forward and backward chaining, are accompanied by other methods less often encountered in AI textbooks. These include the belief propagation algorithm for polytrees due to Pearl (1988), a whole selection of reason maintenance systems, and even methods for temporal reasoning (although the latter are not so surprising given the author's research interests).

This breadth of techniques is possible because, unlike, for instance, Bratko's *Prolog Programming for Artificial Intelligence*, this book does not contain much material on Prolog itself. This is not a Prolog textbook; rather it is a study of how various algorithms may be implemented in Prolog. It makes use of the declarative nature of the language to achieve a clear exposition of the algorithms, as well as benefitting from the support that Prolog offers to those writing meta-interpreters and handling definite clause grammars.

The book is suitable for anyone who is interested in the practical application of methods from AI. It could be used as the basis of a very practical course on AI for students who already have a grounding in Prolog, or given the on-line availability of all the code, as a sourcebook of researchers who want to investigate new methods. With the large number of quite advanced exercises it contains, the book could possibly even find a place on an advanced Prolog course.