REVIEW OF *FOUNDATIONS OF LOGIC PROGRAMS*, BY J. W. LLOYD*

KRZYSZTOF R. APT

So what is the history of logic programming? Well, first Kowalski published in 1974 his paper *Predicate Logic as a Programming Language*. Shortly thereafter Hill proved soundness and completeness of the resolution method used in it. Next, Van Emden and Kowalski defined in 1976 various types of semantics for logic programs and Clark suggested in 1978 how to deal in its context with negation by failure.

But Hill's proof was very obscure and Clark's paper hardly contained any proofs, so Apt and Van Emden (re)proved most of these results in a uniform framework of fixpoint semantics of Van Emden and Kowalski. Their results on negation as failure were subsequently improved by Lassez et al. Finally, in 1983 Andreka et al. extended some of these results to the case of nonterminating processes. And all this took place in the shadow of PROLOG conceived and implemented by Colmerauer et al. in 1972.

This is at least how the history of the subject looks once one untangles the references in this book. The author chose as the skeleton of the book the paper of Apt and Van Emden, thus treating in turn fixpoint semantics, soundness, and completeness of SLD resolution and of SLD refutation procedures, followed by a discussion of negation as failure; preceded it by a short introduction to logic, including a treatment of unification; and succeeded it by an account of the perpetual (i.e., nonterminating) processes. Then he sprinkled the material at appropriate places with helpful references to PROLOG.

The outcome is a clearly and rigorously written small book that provides a good introduction to the subject.

However, some of the results on logic programs are missing. For example, there is no mention of the folklore result, first proved by Tarnlund in 1977, that logic programs compute exactly all computable functions, with all its subsequent improvements. Also, regrettably, no good unification algorithm is provided.

Treatment of PROLOG throughout the book is a bit erratic. No formal definition of its semantics is given. Nothing is said about the important PROLOG facilities for generating all solutions to a query (like the command *all* of the micro-PROLOG of Clarke and McCabe).

*Springer-Verlag, New York, 1984, 124 pp.

Also, there are no interesting examples of programs. The reason for this is probably the author's assumption stated in the introduction that the reader is familiar with "a logic programming language, such as PROLOG." This makes the book somewhat less self-contained than one would wish. On the other hand, such issues like occur check, cut, control in PROLOG, and the implementation of negation are explained.

By putting these minor shortcomings aside, the book is well written and will surely become a standard reference to the subject.