



Book Reviews

Computing with Logic: Logic Programming with Prolog

R. B. Abhyankar

Emphasizing theory and implementation issues more than specific applications and Prolog programming techniques, *Computing with Logic: Logic Programming with Prolog* (The Benjamin Cummings Publishing Company, Menlo Park, Calif., 1988, 535 pp., \$27.95) by David Maier and David S. Warren, respected researchers in logic programming, is a superb book. Offering an in-depth treatment of advanced topics, the book also includes the necessary background material on logic and automatic theorem proving, making it self-contained. The only real prerequisite is a first course in data structures, although it would be helpful if the reader has also had a first course in program translation.

The book has a wealth of exercises and would make an excellent textbook for advanced undergraduate or graduate students in computer science; it is also appropriate for programmers interested in the implementation of Prolog. The book presents the concepts of logic programming using theory presentation, implementation, and application of Prolog, Datalog, and Prolog, three logic programming languages of increasing complexity that are based on horn clause subsets of propositional, predicate, and functional logic, respectively. This incremental approach, unique to this book, is effective in conveying a thorough understanding of the subject.

The book consists of 12 chapters grouped into three parts (Part 1: chapters 1 to 3, Part 2: chapters 4 to 6, and Part 3: chapters 7 to 12), an appendix,

and an index. The three parts, each dealing with one of these logic programming languages, are organized the same. First, the authors informally present the language using examples; an interpreter is also presented. Then the formal syntax and semantics for the language and logic are presented, along with soundness and completeness results for the logic and the effects of various search strategies. Next, they give optimization techniques for the interpreter. Each chapter ends with exercises, brief comments regarding the material in the chapter, and a bibliography.

Chapter 1 presents top-down and bottom-up interpreters for Prolog. Chapter 2 offers a good discussion of the related notions: negation as failure, closed-world assumption, minimal models, and stratified programs. Chapter 3 considers clause indexing and lazy concatenation as optimization techniques for the Prolog interpreter in chapter 1. Chapter 4 explains the connection between Datalog and relational algebra. Chapter 5 contains a proof of Herbrand's theorem for predicate logic. Chapter 6 presents structure sharing as an optimization technique for the Datalog interpreter in chapter 4. Chapter 7 gives example Prolog programs dealing with lists and graphs. Chapter 8 discusses the "cut" and valuable predicates and metapredicates of Prolog. Chapter 9 contains a proof of Herbrand's theorem for functional logic. Chapters 10 and 11 describe several optimization techniques for the Prolog interpreter in chapter 7. Chapter 11 also presents the Warren Prolog Machine, a virtual machine whose machine language can serve as the intermediate language for a Prolog compiler. Chapter 12 details the implementation of a relational

database query language in Prolog. The appendix contains suggestions for two courses that can be taught using the book, one emphasizing the theory and applications of logic programming and another stressing the implementation of logic programming.

Although the book provides an excellent treatment of the topics that it covers, it omits some areas. Topics not covered in the book include Prolog programming techniques and applications such as expert systems; approaches to logic programming besides Prolog and its subsets; derivation and verification of logic programs; fix-point theory, deductive databases, and other topics in the theory of logic programming; and Concurrent Prolog. The book is written in a clear and lucid style. The bibliographic entries are up to date, and the index is detailed. The typography is good.

In summary, the book is a most valuable addition to the Prolog literature. It consolidates a lot of material scattered across a large number of technical reports, conference proceedings, and journal articles. Its emphasis on implementation and theory and its incremental approach to the presentation of Prolog distinguish it from other available books on Prolog and make it an excellent text for the computer science student.

How Machines Think: A General Introduction to Artificial Intelligence Illustrated in Prolog

Debra Sloane

How Machines Think: A General Introduction to Artificial Intelligence Illustrated in Prolog (John Wiley &

Sons Ltd., Chichester, England, 1987, 206 pp., \$13.95) by Nigel Ford purports to be both an introduction to AI and an examination of whether machines are cognizant entities. With this pairing, Ford intends to begin at the beginning, answering the question What is AI? and to proceed to his main thesis about whether machines can think. Unfortunately, after introducing AI, Ford is unable to move onto the higher plane of his main thesis. The book is meant for the non-computer specialist; however, I cannot recommend it, especially to those new to the field.

The text is divided into four parts: Basic Processes, Strategies for Intelligent Behavior, Intelligent Systems and Their Development, and Computers as Intelligent Life? Basic Processes provides an elementary review of programming in general using Prolog, and builds to an understanding of complex systems by selectively looking at the details of programming. In so doing, he gives examples and explanations of basic programming processes using both Prolog and English translations. This part might be helpful to the computer novice, but others are likely to find it tedious and nonproductive.

The depth and challenge of the information presented picks up with Strategies for Intelligent Behavior. Although Ford too often resorts to cute tools to get his point across, several clear and enlightening sections are presented in this part. In fact, it is here that he finally delves into the concept of machine learning. His examples are well referenced and useful in presenting his point, especially in the chapter on learning. Here, he uses several decision tree examples developed by J. R. Quinlan that are quite helpful in understanding the possibilities of machine learning.

However, Ford's choice of examples in Intelligent Systems and Their Development is not nearly as successful. In this part, Ford presents eight expert systems in an attempt to give some real-world credibility to his thesis. Unfortunately, Ford not only fails to choose the most current examples (instead, selecting some badly out-of-date examples that take advantage of no new strategies), he is also inconsistent in relating these examples to the

key points made in previous parts. This inconsistency is especially curious given the fact that in earlier parts, he so painstakingly explained each new concept. Although Ford sometimes does reference his material, he fails to elaborate on its usage within the expert system referenced. With a complete referencing strategy, he could have presented the reader with some concrete points of comparison between theory and example.

When Ford admits that his thesis is all pure speculation in Computers as Intelligent Life?, any hope for truly understanding whether machines can learn is dashed. In this quite philosophical chapter in which he quotes such scholarly sources as Margaret Boden and Donald Michie, thought-provoking questions are raised about what differentiates a human being from a machine: Is it understanding, intentionality, or the ability to impose meaning on life? However, no questions are answered. In fact, Ford appears to conclude that if we believe machines think and learn, then maybe they do.

Ford's book was unsuccessful for several reasons. First, in the first half of the book, too much time is spent in proving the link between the need to look at programming details and the understanding of how machines learn. Second, his references, most glaringly his list of suppliers in appendix 3, are often inaccurate and out of date. Third, Ford is a British author; his use of British colloquialisms and examples is tiresome, and often counterproductive to an international audience. What Ford can be praised for, however, is an in-depth current bibliography-review of additional books and references. It is here that the reader might want to begin a search into whether machines learn.

Ford certainly chose a challenging topic for his book, a topic that can be looked at from a hardware, software, philosophical, or moral perspective. It is certainly not a topic to be used as a backdrop for a general introduction to AI. For whatever reason, Ford's book does not succeed with either topic—an introduction to AI or a computer's ability to think. The topic is still available, therefore, for a thorough treatment.

Reasoning about Change

V. S. Subrahmanian

Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence (Cambridge, Mass., MIT Press, 1988, 200 pp., \$25) by Yoav Shoham is a revised version of the author's 1987 doctoral dissertation for Yale University. The principal aim of the book is to solve two problems, namely, the qualification problem and the extended prediction problem, that are closely related to the well-known frame problem in AI.

Often, one wants to use rules of the form, "If p is true at time t_1 , then q is true at time t_2 ." One of the author's favorite examples is, "If a loaded gun is fired at time t , then there is a loud noise at time $(t + \epsilon)$." The author argues that this rule is almost always true (presumably in the author's intended interpretation), but under extraordinary circumstances (for example, the gun has no firing pin), the rule might not be correct. In addition, one can come up with many circumstances that prevent one from inferring the occurrence of a loud noise. With this information, the author explores these two problems. The qualification problem concerns the actual use of the rule to infer the occurrence of a loud noise given that the gun has been fired when no extraordinary circumstances are known to exist. The extended prediction problem concerns making predictions about the truth of propositions over extended periods of time. The author then introduces the reader to some elementary notions in classical and modal logic.

Chapter 2 is the first original contribution of the book. Two logics, a so-called classical one and a modal one, are introduced. It took me some time to understand what the author was saying here because the expression proposition type was used repeatedly before being informally defined on page 47 as a primitive propositional symbol (propositional logic) or a relation with arguments (first-order logic). The logics themselves conveyed little besides a definition of the satisfaction relation. Although I found the classical logic fairly natural to use, the

modal logic seemed unnatural (a good example of the unnaturalness of this logic is the author's example on automatic planning [p. 60]) A brief, non-technical but highly interesting discussion of the expressiveness and complexity of these two logics is given in section 2.5

Chapter 3 contains some conceptually important, though mathematically straightforward, results on non-monotonic reasoning. It is assumed that the set of structures (or interpretations) of a first-order (though possibly nonclassical) language L is (strictly) partially ordered by a relation $<$. Based on this ordering, certain structures are identified as preferred structures, and a sentence A in L can be satisfied preferentially if it has a preferred model. Preferred entailment is the standard property. $T \models_{<} A$ iff A is true in all preferred models of T . Relations between standard satisfaction (respectively entailment) and preferred satisfaction (respectively entailment) are then derived (although in some cases, certain additional restrictions are imposed on the $<$ relation). The author shows how various proposals for nonmonotonic reasoning (for example, Moore's work on autoepistemic logics, Bossu and Siegel's work, Reiter's work on default reasoning) relate to his framework. I consider this chapter important because the theoretic framework developed in it is quite general and is applicable to many different non-monotonic logics (namely, those obtained by varying the relation $<$).

Although throughout this book the author emphasizes model theory (see p. 23), I found the actual model theoretic investigations to be mathematically trivial. No investigations are reported about whether Lowenheim-Skolem theorems, Craig Interpolation Lemmas, and so on, exist for some of the logics introduced in this book (I want to emphasize that these criticisms are not intended to reflect on the author's achievements. It is not clear that the answers to these concerns would have solved the problems Shoham addresses, nor is it clear that the solution to these problems is important to AI. Nonetheless, I would like to have seen these issues tackled.) On the other hand, the transition

from a vague, ill-defined problem to a precisely formulated problem is not always a trivial task. Certainly, the formulation of the various logics proposed in the book must have taken a fair amount of time and effort, and it might be unreasonable on my part to expect a deep investigation of the logics themselves.

Chapter 4 is concerned with a special case from chapter 3 obtained by assuming time to be the set of integers (positive and negative) and assuming a specific ordering, $<$, of interpretations. A (monotonic) logic TK is defined and from it is derived a new logic $TK_{<}$ called the logic of chronologic ignorance. Maximal preferred models (relative to the $<$ ordering) are called chronologically maximally ignorant models. A class of theories in $TK_{<}$ called causal theories is identified; it is shown that such theories have CMI models (moreover, all these CMI models are equivalent in a certain sense made precise in the book; see p. 112). The utility of these causal theories is demonstrated by example. I was unable to definitively judge how useful these causal theories are in practice.

Chapter 5 is concerned with solving the extended prediction problem by introducing the notion of potential histories. Certain theories called inertial theories are shown to have CMI models.

Because the book is essentially a Ph.D. thesis, it might not be fully comprehensible to all readers, although the author has tried commendably to make it readable. Anyone who has taken an introductory course in logic should be able to follow the book quite easily. The book is well organized, and a short description of each chapter's contents is given at the beginning. From a technical point of view, I was unable to find any serious errors. I was, however, a bit unhappy with some of the references—for example, references 73 and 95 are incompletely specified.

The author's purpose, namely, solving the extended prediction and the qualification problems, was achieved (to a significant extent, anyway). The frame problem, which until now has only been vaguely defined, is given a fairly precise definition—the best I

have seen in the literature to date. The logics formulated in the book appear to have captured several of the epistemological intuitions the author had. I was surprised that Farinas del Cerro's Herbrand theorem for various modal logics was ignored. Proof theory was totally ignored. However, as correctly stated by the author, complete proof systems do not exist for most of the logics proposed in the book. Nonetheless, computational procedures need to be devised, and queries need to be processed. Theorem proving in these logics is, thus, an important topic that needs to be addressed in the future.

I strongly recommend this book to anybody who plans to do AI research. It provides a good introduction to reasoning about change and is written in a clear, concise fashion.

Tutorial Syllabi

Tutorials presented at the National Conference for Artificial Intelligence serve the continuing educational needs of AI practitioners and respond to those who will be implementing AI into their firm. Not intended as stand-alone textbooks, these syllabi nevertheless provide notes and a brief tutorial outline.

- Introduction to AI and Expert Systems—What Goes Wrong When You Try to Explain AI
- Natural Language—Beyond Single Sentence Systems
- Programming with the Common Lisp Object System
- Planning and Reasoning about Time
- Managing Expert Systems Projects
- Neural Network Architectures for Artificial Intelligence
- Uncertainty Management in AI Systems
- Truth Maintenance Systems
- Verification and Validation of Expert Systems
- Artificial Intelligence Applications in Hardware Diagnosis
- Knowledge Acquisition for Knowledge-Based Systems
- Knowledge Representation Design Issues
- Task-Specific Problem Solving Architectures
- Introduction to AI and Expert Systems: Their Application and Consequences
- Logic Programming Techniques
- Evaluation Knowledge Engineering Tools
- Constraint Directed Reasoning
- Artificial Intelligence and Design

Each Syllabi: \$9 members,
\$12 nonmembers