

REVIEW FROM ACM COMPUTING REVIEWS

Logic programming with Prolog (2nd ed.)

[Bramer M.](#), Springer Publishing Company, Incorporated, New York, NY, 2014. 255 pp. Type: Book (978-1-447154-86-0)

Date Reviewed: Jun 23 2014

Declarative programming languages, such as Prolog, are much less familiar to professionals and students in computer science than are the usual object-oriented and procedural languages like C/C++, Java, or Fortran. The first languages novices encounter are object oriented and procedural, and the problems used in pedagogy are those for which these languages are quite well suited. Declarative languages, like Prolog, are usually encountered briefly and somewhat superficially because of time constraints in an upper-division programming languages course.

The author selected SWI-Prolog as the implementation of the language on which the book is based. This version is available on the web as a download at no cost to the user. He also identifies and provides links to several other versions of the language. SWI-Prolog is available in 32-bit and 64-bit Windows versions (Windows XP through Windows 8) and for Mac OS. There are also instructions on creating SWI-Prolog on Linux.

The book takes a very hands-on approach to the study of Prolog. Most of the discussion in the chapters is accomplished by brief examples and short case studies that illustrate not only the correct syntax of the language, but also the semantics of programs in Prolog, with deeper discussion on how the language interpreter generates the responses that it displays. The author opens up the lid and lets the reader look inside the logic of the language. However, merely reading the book will not be nearly as fruitful as actually working through the examples in the text while reading the discussion. The reader should place the book right next to the computer's keyboard.

There are 13 chapters and four appendices in the book. After a brief introduction, which should not be overlooked, chapter 1 discusses the user interface of Prolog and data objects (types) in Prolog. Chapter 2 introduces the reader to the fundamental structural components of clauses, rules, and facts, the basic building blocks of any system of logic, and the use of variables. Chapter 3 is on satisfying goals, and is one of the most important chapters, conceptually, in the book. Detailed examples and explanations are given to show how goals are satisfied through backtracking.

The arithmetic and logical operators in Prolog and their precedence in evaluating expressions are explored in chapter 4. Chapter 5 examines input and output to other devices and/or files other than the keyboard or screen. Loops are implemented in Prolog. Chapter 6 shows how they can be constructed. All of the common loop constructions can be implemented in Prolog, but they require a bit more work than those provided in procedural languages. All of the constructions seem to be based on recursion.

Backtracking takes place automatically in order to satisfy goals. In chapter 7, the author discusses how backtracking can be turned off, as well as the appropriate circumstances when this action can be taken. Chapter 8 instructs the reader on how predicates can be added to or deleted from the Prolog database. The Prolog database includes the built-in components of the language plus all user-defined clauses that were added during the session. However, when the session is done, the user-defined predicates are removed. This chapter shows how user-created predicates can be made more permanent.

The list is a fundamental and flexible data structure in Prolog. Chapter 9 shows how a programmer can formulate and manipulate data in lists and process lists using recursion. String processing is explored in chapter 10. Understanding the chapter is necessary in many of the typical applications of Prolog. Chapter 11 teaches the reader how to extend the Prolog language by defining operators and operations on sets, numbers, and strings.

The next two chapters describe how Prolog is used to implement sophisticated applications in artificial intelligence. Chapter 12 uses grammar rules to analyze English sentences (natural language processing). Chapter 13 completes the examples by showing two projects. In the first, an artificial language can be defined and processed; in the second, a shell program is implemented. In both of these examples, all of the previously discussed language features are applied.

Each chapter ends with several problems or projects. There are not many, and they provide the reader an opportunity to experiment using more open-ended exercises. The four appendices contain the built-in predicates, the built-in operators, answers to selected exercises, and a glossary.

This book is a very practical introduction to the Prolog programming language. It would be good for students in a classroom setting or for a professional working at his or her desk. A job well done.

More reviews about this item: [Amazon](#)

Reviewer: [Anthony J. Duben](#)