Clause And Effect Prolog Programming For The Working Programmer

BRODY KYLEE

Problem Solving With Prolog Springer Science & Business Media
This book is an introduction to Prolog (£rQgramming in ~ic). It presents the basic foundations of Prolog and basic and fundamental programming methods. This book is written for programmers familiar with other programming languages, as well as for novices in computer science, willing to have an original introduction to programming. The approach adopted in this book is thus based on methodological elements together with some pragmatic aspects. The book is composed of two parts. In the flrst part the major aspects of programming in Prolog are presented step by step. Each new aspect is illustrated by short examples and exercises. The second part is composed of more developed examples, which are often games, that illustrate major aspects of artiflcial intelligence. More advanced books are given in the bibliography and will allow the reader to deepen his or her know ledge of Prolog. Prolog was first designed in France at OJ.A., Marseille, with a specific syntax. We have adopted here a more common notation, deflned at Edinburgh, which tends to be an implicit norm. At the end of each chapter of the flrst part, there are exercises that the reader is invited to do and to test on his or her machine. Complete answers are given in Appendix A, at the end of the book.
A High Performance Architecture for Prolog Addison-Wesley Longman
This new edition of The Art of Prolog contains a number of important changes. Most background sections at the end of each chapter have been updated to take account of important recent research results, the references have been greatly expanded, and more advanced exercises have been added which have been used successfully in teaching the course. Part II, The Prolog Language, has been modified to be compatible with the new Prolog standard, and the chapter on program development has been significantly altered: the predicates defined have been

moved to more appropriate chapters, the section on efficiency has been moved to the considerably expanded chapter on cuts and negation, and a new section has been added on stepwise enhancement—a systematic way of constructing Prolog programs developed by Leon Sterling. All but one of the chapters in Part III, Advanced Prolog Programming Techniques, have been substantially changed, with some major rearrangements. A new chapter on interpreters describes a rule language and interpreter for expert systems, which better illustrates how Prolog should be used to construct expert systems. The chapter on program transformation is completely new and the chapter on logic grammars adds new material for recognizing simple languages, showing how grammars apply to more computer science examples.

Answer Set Programming Springer Science & Business Media

Prolog is important as one of the major programming languages. Beginning with a chapter on logic (which makes the book particularly useful to undergraduate students), Prolog for Computer Science provides a comprehensive tutorial that assumes no prior knowledge of programming. There are lots of realistic examples and case-studies, including an English-Dutch translator.

Adventure in Prolog Springer Science & Business Media

Logic Programming is the name given to a distinctive style of programming, very different from that of conventional programming languages such as C++ and Java. By far the most widely used Logic Programming language is Prolog. Prolog is a good choice for developing complex applications, especially in the field of Artificial Intelligence. Logic Programming with Prolog does not assume that the reader is an experienced programmer or has a background in Mathematics, Logic or Artificial Intelligence. It starts from scratch and aims to arrive at the point where quite powerful programs can be written in the language. It is intended both as a textbook for an introductory course and as a self-study book. On completion readers will know enough to use Prolog in their own research or practical projects. Each chapter has self-assessment exercises so that readers may check their own progress. A glossary of the technical terms used completes the book. This second edition has been revised to be fully compatible with SWI-Prolog, a popular multi-platform public domain implementation of the language. Additional chapters have been added covering the use of Prolog to analyse English sentences and to illustrate how Prolog can be used to implement applications of an 'Artificial Intelligence' kind. Max Bramer is Emeritus Professor of Information Technology at the University of Portsmouth, England. He has taught Prolog to undergraduate computer science students and used Prolog in his own work for many years.

Clause and Effect Bookboon

Logic program synthesis and transformation are topics of central importance to the software industry. The demand for software can not be met by the current supply, in terms of volume, complexity, or reliability. The most promising solution seems to be the increased automation of software production: programmer productivity would improve, and correctness could be ensured by the application of mathematical methods. Because of their mathematical foundations, logic programs lend themselves particularly well to machine-assisted development techniques, and therefore to automation. This volume contains the proceedings of the second International Workshop on Logic Program Synthesis and Transformation (LOPSTR 92), held at the University of Manchester, 2-3 July 1992. The LOPSTR workshops are the only international meetings devoted to these two important areas. A variety of new techniques were described at the workshop, all of which promise to revolutionize the software industry once they become standard practise. These include techniques for the transformation of an inefficient program into an equivalent, efficient one, and the synthesis of a program from a formal specification of its required behaviour. Among the topics covered in this volume are: optimal transformation of logic programs; logic program synthesis via proof planning; deductive synthesis of programs for query answering; efficient compilation of lazy narrowing into Prolog; synthesis of narrowing programs; Logimix: a self-applicable partial evaluator for Prolog; proof nets; automatic termination analysis. Logic Program Synthesis and Transformation describes the latest advances in machine-assisted development of logic programs. It will provide essential reading for researchers and postgraduate students concerned with these two important areas.

Programming in Prolog Pearson Scott Foresman

This exciting new text reveals both the evolution of this programming paradigm since its inception and the impressively broad scope of current research in the field. The contributors to this book are all leading world experts in Logic Programming, and they deal with both theoretical and practical issues. They address such diverse topics as: computational molecular biology, machine learning, mobile computing, multi-agent systems, planning, numerical computing and dynamical systems, database systems, an alternative to the "formulas as types" approach, program semantics and analysis, and natural language processing. XXXXXXX Neuer Text Logic Programming was founded 25 years ago. This exciting book reveals both the evolution of this programming paradigm and its impressively broad scope of current research. The contributions by leading computer scientists deal with both theoretical and practical issues. They address diverse topics such as: computational molecular biology, machine learning, mobile computing, multi-agent systems, numerical computing and dynamical systems, database systems, program semantics, natural language processing, and promising future directions.

Productive Prolog Programming Springer Science & Business Media

The emphasis in The Craft of Prolog is on using Prolog effectively. It presents a loose collection of topics that build on and elaborate concepts learned in a first course. Hacking your program is no substitute for understanding your problem. Prolog is different, but not that different. Elegance is not optional. These are the themes that unify Richard O'Keefe's very personal statement on how Prolog programs should be written. The emphasis in The Craft of Prolog is on using Prolog effectively. It presents a loose collection of topics that build on and elaborate concepts learned in a first course. These may be read in any order following the first chapter, "Basic Topics in Prolog," which provides a basis for the rest of the material in the book. Richard A. O'Keefe is Lecturer in the Department of Computer Science at the Royal Melbourne Institute of Technology. He is also a consultant to Quintus Computer Systems, Inc.Contents: Basic Topics in Prolog. Searching. Where Does the Space Go? Methods of Programming. Data Structure Design. Sequences. Writing Interpreters. Some Notes on Grammar Rules. Prolog Macros. Writing Tokenisers in Prolog. All Solutions.

PROLOG, Programming Techniques and Applications Springer

Written for those who wish to learn Prolog as a powerful software development tool, but do not necessarily have any

background in logic or AI. Includes a full glossary of the technical terms and self-assessment exercises.
Specialization in Logic Programming CRC Press
Written for those who wish to learn Prolog as a powerful software development tool, but do not necessarily have any background in logic or AI. Includes a full glossary of the technical terms and self-assessment exercises.
The Art of Prolog MIT Press
Not long ago" Dennis Merritt wrote one of the best books that I know of about implementing expert systems in Prolog, and I was very glad he published it in our series. The only problem is there are still some unfortunate people around who do not know Prolog and are not sufficiently prepared either to read Merritt's book, or to use this extremely productive language, be it for knowledge-based work or even for everyday programming. Possibly this last statement may surprise you if you were under the impression that Prolog was an "artificial intelligence language" with very limited application potential. Please believe this editor's statement that quite the opposite is true: for at least four years, I have been using Prolog for every programming task in which I am given the option of choosing the language. Therefore, I 'am indeed happy that Dennis Merritt has written another good book on my language of choice, and that it meets the high standard he set with his prior book, Building Expert Systems in Prolog. All that remains for me to do is to wish you success and enjoyment when taking off on your Adventure in Prolog.
Logic Programming with Prolog Oxford ; Boston : Blackwell Scientific Publications ; Brookline Village, Mass. : Distributors, USA, Publishers Business Services
This book is for people who have done some programming, either in Prolog or in a language other than Prolog, and who can find their way around a reference manual. The emphasis of this book is on a simplified and disciplined methodology for discerning the mathematical structures related to a problem, and then turning these structures into Prolog programs. This book is therefore not concerned about the particular features of the language nor about Prolog programming skills or techniques in general. A relatively pure subset of Prolog is used, which includes the 'cut', but no input/output, no assert/retract, no syntactic extensions such as if then-else and grammar rules, and hardly any built-in predicates apart from arithmetic operations. I trust that practitioners of Prolog program ming who have a particular interest in the finer details of syntactic style and language features will understand my purposes in not discussing these matters. The presentation, which I believe is novel for a Prolog programming text, is in terms of an outline of basic concepts interleaved with worksheets. The idea is that worksheets are rather like musical exercises. Carefully graduated in scope, each worksheet introduces only a limited number of new ideas, and gives some guidance for practising them. The principles introduced in the worksheets are then applied to extended examples in the form of case studies.
Prolog for Programmers MIT Press
This is a practical introduction to PROLOG for the reader with little experience. It presents problem-solving techniques for program development in PROLOG based on case analysis and the use of a toolkit of PROLOG techniques. The development of larger scale programs and the techniques More...for solving them using the methodology and tools described, through the presentation of several case studies of typical programming problems is also discussed.
Programming in Prolog Springer Science & Business Media
Prolog is a programming language, but a rather unusual one. Prolog'' is short for Programming with Logic'', and the link with logic gives Prolog its special character. At the heart of Prolog lies a surprising idea: don't tell the computer what to do. Instead, describe situations of interest, and compute by asking questions. Prolog will logically deduce new facts about the situations and give its deductions back to us as answers. Why learn Prolog? For a start, its say what the problem is, rather than how to solve it'' stance, means that it is a very high level language, good for knowledge rich applications such as artificial intelligence, natural language processing, and the semantic web. So by studying Prolog, you gain insight into how sophisticated tasks can be handled computationally. Moreover, Prolog requires a different mindset. You have to learn to see problems from a new perspective, declaratively rather than procedurally. Acquiring this mindset, and learning to appreciate the links between logic and programming, makes the study of Prolog both challenging and rewarding. Learn Prolog Now! is a practical introduction to this fascinating language. Freely available as a web-book since 2002 (see www.learnprolognow.org) Learn Prolog Now! has became one of the most popular introductions to the Prolog programming language, an introduction prized for its clarity and down-to-earth approach. It is widely used as a textbook at university departments around the world, and even more widely used for self study. College Publications is proud to present here the first hard-copy version of this online classic. Carefully revised in the light of reader's feedback, and now with answers to all the exercises, here you will find the essential material required to help you learn Prolog now.
Prolog Programming Springer Science & Business Media
Addressed to readers at different levels of programming expertise, The Practice ofProlog offers a departure from current books that focus on small programming examples requiringadditional instruction in order to extend them to full programming projects. It shows how to designand organize moderate to large Prolog programs, providing a collection of eight programmingprojects, each with a particular application, and illustrating how a Prolog program was written tosolve the application. These range from a simple learning program to designing a database formolecular biology to natural language generation from plans and stream data analysis.Leon Sterlingis Associate Professor in the Department of Computer Engineering and Science at Case Western ReserveUniversity. He is the coauthor, along with Ehud Shapiro, of The Art of Prolog.Contents: A SimpleLearning Program, Richard O'Keefe. Designing a Prolog Database for Molecular Biology, Ewing Lusk,Robert Olson, Ross Overbeek, Steve Tuecke. Parallelizing a Pascal Compiler, Eran Gabber. PREDITOR: AProlog-Based VLSI Editor, Peter B. Reintjes. Assisting Register Transfer Level Hardware Design, PaulDrongowski. Design and Implementation of aPartial Evaluation System, Arun Lakhotia, Leon Sterling.Natural Language Generation from Plans, Chris Mellish. Stream Data Analysis in Prolog, Stott Parker.
The Logic Programming Paradigm Prentice Hall
Answer set programming (ASP) is a programming methodology oriented towards combinatorial search problems. In such a problem, the goal is to find a solution among a large but finite number of possibilities. The idea of ASP came from research on artificial intelligence and computational logic. ASP is a form of declarative programming: an ASP program describes what is counted as a solution to the problem, but does not specify an algorithm for solving it. Search is performed by sophisticated software systems called answer set solvers. Combinatorial search problems often arise

in science and technology, and ASP has found applications in diverse areas—in historical linguistic, in bioinformatics, in robotics, in space exploration, in oil and gas industry, and many others. The importance of this programming method was recognized by the Association for the Advancement of Artificial Intelligence in 2016, when AI Magazine published a special issue on answer set programming. The book introduces the reader to the theory and practice of ASP. It describes the input language of the answer set solver CLINGO, which was designed at the University of Potsdam in Germany and is used today by ASP programmers in many countries. It includes numerous examples of ASP programs and present the mathematical theory that ASP is based on. There are many exercises with complete solutions.

Prolog Programming Springer Science & Business Media

This book constitutes the refereed proceedings of the 19th International Conference on Logic Programming, ICLP 2003, held in Mumbai, India in December 2003. The 23 revised full papers and 19 poster papers presented together with 5 invited full contributions and abstracts of 4 invited contributions were carefully reviewed and selected from 81 submissions. All current issues in logic programming are addressed.

The Reasoned Schemer, second edition Springer Nature

The first book on Prolog ++, an important new language combining object-orientation with logic programming. Includes tutorial style with worked examples, exercises, summaries, etc., significant applications coverage, state-of-the-art coverage of other approaches including parallel language, and distributed databases.

Programming in Prolog MIT Press

Since the first publishing of Programming in Prolog in 1981, Pro log has continued to attract an unexpectedly great deal of interest in the computer science community and is now seen as a potential ba sis for an important new generation of programming languages and systems. We hope that Programming in Prolog has partially satisfied the increasing need for an easy, yet comprehensive introduction to the language as a tool for practical programming. In this second edition we have taken the opportunity to improve the presentation and to correct various minor errors in the original. We thank the many people who have given us suggestions for corrections and improvement. W. F. C. C. S. M. Cambridge, England August, 198-1 Preface to the First Edition The computer programming language Prolog is quickly gaining popularity throughout the world. Since its beginnings around 1970, Prolog has been chosen by many programmers for applications of symbolic computation, including: • relational databases • mathematical logic • abstract problem solving • understanding naturallangur~ge • design automation • symbolic equation solving • biochemical structure analysis • many areas of artificial intelligence Until now, there has been no textbook with the aim of teaching Prolog as a practical programming language. It is perhaps a tribute to Prolog that so many people have been motivated to learn it by referring to the necessarily concise reference manuals, a few published papers, and by the orally transmitted 'folklore' of the modern computing community.

Programming in Prolog Springer Science & Business Media

An introduction to PROLOG; PROLOG and logic; Metamorphosis grammars: a powerful extension; Simple programming techniques; Summary of syntax and built-in procedures; Principles of PROLOG implementation; TOY: an exercice in implementation; Two case studies; Prolog dialects.

Prolog Techniques MIT Press

Originally published in 1981, this was the first textbook on programming in the Prolog language and is still the definitive introductory text on Prolog. Though many Prolog textbooks have been published since, this one has withstood the test of time because of its comprehensiveness, tutorial approach, and emphasis on general programming applications. Prolog has continued to attract a great deal of interest in the computer science community, and has turned out to be a basis for an important new generation of programming languages and systems for Artificial Intelligence. Since the previous edition of Programming in Prolog, the language has been standardised by the International Organization for Standardization (ISO) and this book has been updated accordingly. The authors have also introduced some new material, clarified some explanations, corrected a number of minor errors, and removed appendices about Prolog systems that are now obsolete.