Clause And Effect: Prolog Programming For The Working Programmer

Batch chemical processing has in the past decade enjoyed a return to respectability as a valuable, effective, and often preferred mode of process operation. This book provides the first comprehensive and authoritative coverage that reviews the state of the art development in the field of batch chemical systems engineering, applications in various chemical industries, current practice in different parts of the world, and future technical challenges. Developments in enabling computing technologies such as simulation, mathematical programming, knowledge based systems, and prognosis of how these developments would impact future progress in the batch domain are covered. Design issues for complex unit processes and batch plants as well as operational issues such as control and scheduling are also addressed.
This book constitutes the thoroughly refereed post-conference proceedings of the 24th International Conference on Inductive Logic Programming, ILP 2014, held in Nancy, France, in September 2014. The 14 revised papers presented were carefully reviewed and selected from 41 submissions. The papers focus on topics such as the inducing of logic programs, learning from data represented with logic, multi-relational machine learning, learning from graphs, and applications of these techniques to important problems in fields like bioinformatics, medicine, and text mining.
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.
Clause and EffectProlog Programming for the Working ProgrammerSpringer Science & Business Media
Learn Prolog Now!
First International Conference, ICTL '94, Bonn, Germany, July 11 - 14, 1994. Proceedings
The Art of Prolog
Prolog Techniques
1987 Symposium on Logic Programming
History and Cultural Theory

History and Cultural Theory provides an introduction to the relationship between contemporary cultural theory and the study of history. Reflecting the growing influence on history of theorists such as Pierre Bourdieu, Michel Foucault and Gayatri Spivak, it provides a clear and accessible guide to their thought and explains the implications of their ideas for historical studies. It offers specific examples of how historians apply the insights of cultural theory in their own work. Provides a guide to cutting-edge ideas in historical thought.
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.
Logic programming synthesis and transformation are methods of deriving logic programs from their specifications and, where necessary, producing alternative but equivalent forms of a given program. The techniques involved in synthesis and transformation are extremely important as they allow the systematic construction of correct and efficient programs and have the potential to enhance current methods of software production. Transformation strategies are also being widely used in the field of logic program development. LOPSTR 91 was the first workshop to deal exclusively with both logic program synthesis and transformation and, as such, filled an obvious gap in the existing range of logic programming workshops. In attempting to cover the subject as comprehensively as possible, the workshop brought together researchers with an interest in all aspects of logic (including Horn Clause and first order logic) and all approaches to program synthesis and transformation. Logic Program Synthesis and Transformation provides a complete record of the workshop, with all the papers reproduced either in full or as extended abstracts. They cover a wide range of aspects, both practical and theoretical, including the use of mode input-output in program transformation, program specification and synthesis in constructive formal systems and a case study in formal program development in modular Prolog. This volume provides a comprehensive overview of current research and will be invaluable to researchers and postgraduate students who wish to enhance their understanding of logic programming techniques.
Programming Language Pragmatics, Third Edition, is the most comprehensive programming language book available today. Taking the perspective that language design and implementation are tightly interconnected and that neither can be fully understood in isolation, this critically acclaimed and bestselling book has been thoroughly updated to cover the most recent developments in

programming language design, inclouding Java 6 and 7, C++0X, C# 3.0, F#, Fortran 2003 and 2008, Ada 2005, and Scheme R6RS. A new chapter on run-time program management covers virtual machines, managed code, just-in-time and dynamic compilation, reflection, binary translation and rewriting, mobile code, sandboxing, and debugging and program analysis tools. Over 800 numbered examples are provided to help the reader quickly cross-reference and access content. This text is designed for undergraduate Computer Science students, programmers, and systems and software engineers. Classic programming foundations text now updated to familiarize students with the languages they are most likely to encounter in the workforce, including including Java 7, C++, C# 3.0, F#, Fortran 2008, Ada 2005, Scheme R6RS, and Perl 6. New and expanded coverage of concurrency and run-time systems ensures students and professionals understand the most important advances driving software today. Includes over 800 numbered examples to help the reader quickly cross-reference and access content.
A High Performance Architecture for Prolog
Proceedings, August 31-September 4, 1987, Hyatt on Union Square, San Francisco, California
The Craft of Prolog
The Art of Prolog, second edition

Symposium on Logic Programming
The contributions to this volume cover all aspects of the assessment and management of hepatobiliary disease. The focal points of the book consist of three state-of-the-art summaries. The first of these deals with the highly topical problem of liver transplants from the point of view of patient selection. The second considers drug-induced liver injury in view of the fact that the liver is the main metabolic site for a number of drugs. The final summary deals with liver and aging: it asks whether the liver follows the aging process of the host organisms and whether the liver of aged liver transplant candidate donors could be suitable for grafting. Aside from these topics, the volume presents basic research on hepatic transport mechanisms, intrahepatic cholestasis and gall-stone disease, which serves as a background for the topics more specifically concerning the assessment of liver function. Much of the book is then devoted to the management of the commonest forms of liver diseases and their complications, such as chronic active hepatitis, liver cirrhosis, portal hypertension, hepatic encephalopathy, hepatorenal syndrome, and ascites.
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.
SICStus Prolog is the de-facto standard industrial Prolog programming environment. With more than 25 years in fielded applications, it has a proven track record of a robust, scalable and efficient system. It is widely used for commercial applications as well as in research and education. This book edition contains the core reference documentation of SICStus Prolog release 4.3.0. SICStus Prolog complies with the ISO Prolog standard, IPv4, IPv6, and Unicode 5.0. It is interoperable with C, C++, .NET, Java, Tcl/Tk, Berkeley DB, ODBC, XML, MiniZinc, and more. It ships with a comprehensive library of modules for abstract data types, program development, operating system and file system access, processes, sockets, constraint solvers, and more. SICStus Prolog compiles to a virtual machine (WAM), emulated by efficient C code and compiled just-in-time to native code for x86-based platforms. Tools provide deployment to stand-alone, all-in-one-file, and embedded applications. The Eclipse-based development environment SPIDER provides semantics-aware editing support, static analysis tools, source-linked debugging, tracking variable bindings, profiling, code coverage, backtraces, call hierarchies, and more.
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.
Logic Program Synthesis and Transformation
Programming Language Pragmatics
Inductive Logic Programming
Programming in Prolog
Third Edition
Logic Programming '87
Summary 'Topics in Programming Languages' explores the arch from the formation of alphabet and classical philosophy to artificial programming languages in the structure of one argumentative topics list: as if it were philosophy interpreted and programmed. One such endeavour is taken to tend toward phonetics and sounds of speech analysis with -calculus, and, ultimately, Prolog - the programming language of choice in artificial intelligence - born of the natural language processing reverie and delusion. The well-ordered list of arguments targets the conceptual tree behind both the functional and the logical, the procedural and the declarative paradigms in programming languages by studying close the ascendum (convolution) of the Aristotelian efficient cause into the notions of function (Leibniz), rule (Kant) and algorithm as effective procedures in computation (Church-Turing). The Author Luis Manuel Cabrita Pais Homem graduated in Philosophy in the Faculty of Letters of the University of Lisbon in 2005. He concluded the Master in the same He is currently completing his doctoral thesis. the Post-Graduate Program holds a Quality Grant, taking in automatic passage to Doctorate, the author is currently preparing the PhD thesis subordinated to the same theme. The author is an integrated member of the Centre for Philosophy of Science of the University of Lisbon since the summer of 2011. Readership Scholars, students, programmers, computer scientists Contents Section I - Arguments;) The phonetics and philosophical argument;) The symbolic or rational argument;) The difficulty argument;) The content-and-form artificial intelligence argument;) The efficient cause argument;) The model theory argument; Notes Section II - Arguments; The endogenous to exogenous language argument;) The

efficient cause continuance argument;) The reviewing incommensurability argument;) The functional and declarative programming languages argument; Notes Section III - Arguments;) The -calculus argument;) The Prolog argument Notes Section IV - Topics in programming languages: a philosophical analysis through the case of prolog; Summary; State of the art; Goal; Detailed description Bibliography"
This volume contains most of the papers presented at the 6th Logic Programming Conference held in Tokyo, June 22-24, 1987. It is the successor of Lecture Notes in Computer Science volumes 221 and 264. The contents cover foundations, programming, architecture and applications. Topics of particular interest are constraint logic programming and parallelism. The effort to apply logic programming to large-scale realistic problems is another important subject of these proceedings.
With more substantial funding from research organizations and industry, numerous large-scale applications, and recently developed technologies, the Semantic Web is quickly emerging as a well-recognized and important area of computer science. While Semantic Web technologies are still rapidly evolving, Foundations of Semantic Web Technologies focuses
This second edition contains revised chapters taking into account recent research advances. More advanced exercises have been included, and "Part II The Prolog Language" has been modified to be compatible with the new Prolog standard. This is a graduate level text that can be used for self-study.
Proceedings of the 6th Conference Tokyo, Japan, June 22-24, 1987
Advanced Programming Techniques
The Logic Programming Tutor
Proceedings of LOPSTR 91, International Workshop on Logic Program Synthesis and Transformation, University of Manchester, 4–5 July 1991
Adventure in Prolog
Using the ISO Standard

This volume constitutes the proceedings of the First International Conference on Temporal Logic (ICTL '94), held at Bonn, Germany in July 1994. Since its conception as a discipline thirty years ago, temporal logic is studied by many researchers of numerous backgrounds; presently it is in a stage of accelerated dynamic growth. This book, as the proceedings of the first international conference particularly dedicated to temporal logic, gives a thorough state-of-the-art report on all aspects of temporal logic research relevant for computer science and AI. It contains 27 technical contributions carefully selected for presentation at ICTL '94 as well as three surveys and position papers.
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.
knowledgewrappedinrules,databases,ortheWeballowsonetoexploreintere- ing hidden knowledge.Declarativetechniques for the transformation,deduction, induction, visualization, or querying of knowledge, or data mining techniques for exploring knowledge have the advantage of high transparency and better maintainability compared to procedural approaches.
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.
Artificial Intelligence and Business Management
Negation and Control in Prolog
Fifth Generation Computer Systems 1992
Concurrent Constraint Programming
Batch Processing Systems Engineering
Clause and Effect

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.
Software -- Programming Languages.
After introducing the concept of artificial intelligence (AI), the authors of this text discuss the scope and limitations of AI technology in the various subfields that are expected to be relevant to business management systems - natural language processing, voice processing, image processing, and intelligent robots.
Can computers think? Updated edition, ideal for lay readers and students of computer science, offers well-illustrated, easy-to-read discussions of problem-solving methods and representations, game playing, neural networks, more. 2019 edition.
Concepts in Programming Languages
Core reference documentation
Applications of Prolog
Answer Set Programming
Foundations of Semantic Web Technologies
Prolog Programming in Depth

This text covers natural language processing in Prolog and presumes knowledge of Prolog, but not of linguistics. It includes simple but practical database query systems; covers syntax, formal semantics, and morphology; emphasizes working computer programs that implement subsystems of a natural language processor; features programs that are clearly designed and compatible with any Edinburgh-compatible prolog implementation (Quintas, ESL, Arity, ALS etc.); and contains nearly 100 hands-on Prolog programming exercises and problem sets.
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.
Originally published in 1981, this was the first textbook on programming in the Prolog language. Today it remains the definitive introductory text on the subject. 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. 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 new material, clarified some explanations, and have removed appendices about Prolog systems that are now obsolete.
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: D relational databases D mathematical logic D abstract problem solving D understanding natural language D architectural design D symbolic equation solving D biochemical structure analysis D 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. However. as Prolog is beginning to be Introduced to large numbers of undergraduate and postgraduate students. many of our colleagues have expressed a great need for a tutorial guide to learning Prolog. We hope this little book will go some way towards meeting this need. Many newcomers to Prolog find that the task of writing a Prolog program Is not like specifying an algorithm in the same way as In a conventional programming language. Instead. the Prolog programmer asks more what formal relationships and objects occur In his problem.
FGCS '92
Logic Programming
19th International Conference, ICLP 2003, Mumbai, India, December 9-13, 2003, Proceedings
Logic Programming with Prolog
17th International Conference, INAP 2007, and 21st Workshop on Logic Programming, WLP 2007, Würzburg, Germany, October 4-6, 2007, Revised Selected Papers
24th International Conference, ILP 2014, Nancy, France, September 14-16, 2014, Revised Selected Papers

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.
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.
A comprehensive undergraduate textbook covering both theory and practical design issues, with an emphasis on object-oriented languages.
Concurrent Constraint Programming introduces a new and rich class of programminglanguages based on the notion of computing with partial information, or constraints, that synthesizeand extend work on concurrent logic programming and that offer a promising approach for treatingthorny issues in the semantics of concurrent, nondeterministic programming languages.Saraswatdevelops an elegant and semantically tractable framework for computing with constraints, emphasizingtheir importance for communication and control in concurrent, programming languages. He describesthe basic paradigm, illustrates its structure, discusses various augmentations, gives a simpleimplementation of a concrete language, and specifies its connections with other formalisms.In thisframework, concurrently executing agents communicate by placing and checking constraints on sharedvariables in a common store. The major form of concurrency control in the system is through theoperations of Atomic Tell - an agent may instantaneously place constraints only if they areconsistent with constraints that have already been placed - and Blocking Ask - an agent must blockwhen it checks a constraint that is not yet known to hold. Other operations at a finer granularityof atomicity are also presented.Saraswat introduces and develops the concurrent constraint family ofprogramming languages based on these ideas, shows how various constraint systems can naturallyrealize data structures common in computer science, and presents a formal operational semantics formany languages in the concurrent constraint family. In addition, he provides a concrete realizationof the paradigm on a sequential machine by presenting a compiler for the concurrent constraintlanguage Herbrand and demonstrates a number of constraint-based concurrent programming techniquesthat lead to novel presentations of algorithms for many concurrent programming problems.Vijay A.Saraswat is Member of the Research Staff at Xerox Palo Alto Research Center.
Temporal Logic
Logic Programming '85
Proceedings of LOPSTR 92, International Workshop on Logic Program Synthesis and

Transformation, University of Manchester, 2–3 July 1992
Introduction to Artificial Intelligence
Applications of Declarative Programming and Knowledge Management
Proceedings of the 4th Conference Tokyo, Japan, July 1-3, 1985