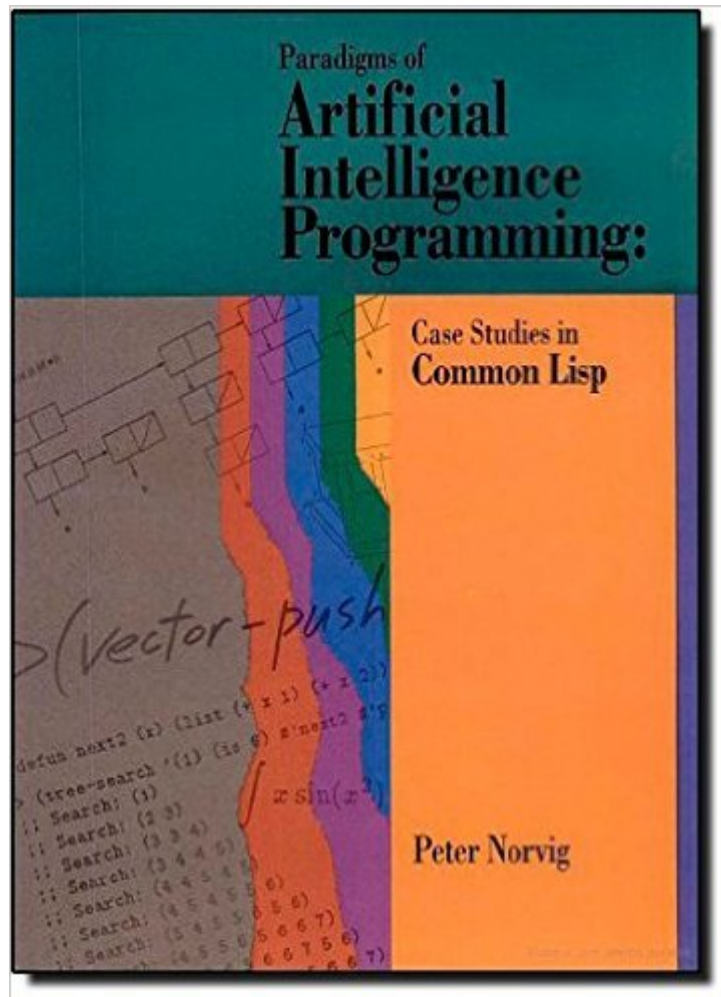


The book was found

Paradigms Of Artificial Intelligence Programming: Case Studies In Common Lisp



Synopsis

Paradigms of AI Programming is the first text to teach advanced Common Lisp techniques in the context of building major AI systems. By reconstructing authentic, complex AI programs using state-of-the-art Common Lisp, the book teaches students and professionals how to build and debug robust practical programs, while demonstrating superior programming style and important AI concepts. The author strongly emphasizes the practical performance issues involved in writing real working programs of significant size. Chapters on troubleshooting and efficiency are included, along with a discussion of the fundamentals of object-oriented programming and a description of the main CLOS functions. This volume is an excellent text for a course on AI programming, a useful supplement for general AI courses and an indispensable reference for the professional programmer.

Book Information

Paperback: 946 pages

Publisher: Morgan Kaufmann; 1 edition (October 15, 1991)

Language: English

ISBN-10: 1558601910

ISBN-13: 978-1558601918

Product Dimensions: 7.5 x 1.9 x 9.2 inches

Shipping Weight: 3.7 pounds (View shipping rates and policies)

Average Customer Review: 4.8 out of 5 starsÂ Â See all reviewsÂ (20 customer reviews)

Best Sellers Rank: #140,884 in Books (See Top 100 in Books) #7 inÂ Books > Computers & Technology > Programming > Languages & Tools > Lisp #34 inÂ Books > Textbooks > Computer Science > Artificial Intelligence #105 inÂ Books > Computers & Technology > Computer Science > AI & Machine Learning > Intelligence & Semantics

Customer Reviews

This book has been called "The best book on programming ever written". I'd have to agree--it is certainly the best that I've ever read. William Zinsser said, "The essence of writing is rewriting" and the same can be said for writing computer programs. Norvig's book presents this process--how the limitations of a program are overcome by revision and rewriting. What sets Norvig apart as a writer is that, amazingly enough, he can write about debugging (the most dreaded part of computer programming) and make it a fascinating read! Lisp has been getting a higher profile lately because of essayists like Paul Graham and Philip Greenspun; in particular, Greenspun's Tenth Rule of Programming which states: "Any sufficiently complicated C or Fortran program contains an ad hoc,

informally-specified, bug-ridden, slow implementation of half of Common Lisp." So, should this book be read as an exhortation to return to Lisp as the preferred programming language? Paradoxically, I think not. One third of the way through the book, Norvig shows us how to implement Prolog in Lisp. From then on out, most of the AI techniques he presents either directly use Prolog instead of Lisp (such as his excellent discussion of natural language processing using Prolog) or use Prolog as a base to build on (such as his discussions on knowledge representation). From this we can abstract what I'd like to call Norvig's Corollary to Greenspun's Tenth Law of Programming: "Any sufficiently complicated LISP program is going to contain a slow implementation of half of Prolog". I'm leaving out the "ad hoc", "bug-ridden" part of Greenspun's law, because Norvig's programs are neither. But it is quite remarkable the degree to which, once having absorbed Prolog, Norvig uses Prolog as the basis for further development, rather than Lisp. Is this a book about Prolog then? Again, no. What is the take-away message? It is this: as our world becomes more and more complex, and as the problems which programmers are facing become more and more complex, we have to program at a higher and higher level. Norvig does not stop at just embedding Prolog in Lisp. He also shows us how to embed scheme as well. Excellent discussion on the mysterious call/cc function and on continuations. In a capsule review, it is impossible to really give an overview of a 1,000 page book like this one. But the scope and heft of the volume really needs to be commented on: the programs presented in this book are like basis vectors, the totality of which nearly span the space of programming itself. In no way should this be considered "just an AI book" or "just a LISP book". This book transcends language, time, and subject matter. It is a programmer's book for the ages.

This book is equally excellent regardless of whether you wish to regard it as: a) A historical study of Artificial Intelligence, with **USABLE** examples of code, or b) A book presenting techniques for programming in Common Lisp. As a reference about Common Lisp, it is certainly lacking, but this is no great problem when both the Common Lisp HyperSpec and Steele's book are readily available in electronic form. It provides something more important: **SIGNIFICANT** examples, and significant discussions on **WHY** you would use various Lisp idioms, and, fairly often, discussions on **HOW** pieces of Common Lisp are likely to be implemented. Its discussion of an implementation of the LOOP macro, for instance, provides a very different point of view than the "references" to LOOP. (Contrast too with Graham's books, which largely deprecate the use of LOOP.) From an AI perspective, it is also very good, providing **WORKING SAMPLES** for a whole lot of the historically significant AI problems, including Search, PLANNER, symbolic computation, and the likes. It would be interesting to see parallel works from the following sorts of perspectives:- The same sorts of AI

problems solved using functional languages (e.g. - ML, Haskell), to allow contrasting the use of those more modern languages. Being more "purely functional" has merits; such languages commonly lack macros, which is something of a disadvantage.- The use of CL to grapple with some other sorts of applications, notably random access to data [e.g. - databases] and rendition of output in HTML/SGML/XML [e.g. - web server].

I have no background in computer science or AI, but found myself needing to use Lisp for various creative and artistic purposes. I've spent a lot of money on books relating to Common Lisp, but I wish I had just gotten this one and Touretzky's "Gentle Introduction to Symbolic Computation." The particular strengths of this book are its detailed discussion of advanced topics, especially optimization, and the practical overview of current and historical AI topics through programming examples. Very clearly written.

This is an extremely advanced book on AI techniques. The examples are the best that I have seen in thirty years of Lisp involvement. The author treats many of the classical AI programs and implements them with Common Lisp. Sophisticated search strategies are discussed with example code being given for all of them. A very strong chapter on program optimization is included, a subject often omitted by Lisp writers. Included is complete code for Eliza as well as a treatment of EMYCIN. The author also treats the implementation of embedded languages and gives PROLOG and SCHEME as examples. The book is addressed to production level programmers. This book is definately not for beginners in the Lisp language.

[Download to continue reading...](#)

Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp
Java: Artificial Intelligence; Made Easy, w/ Java Programming; Learn to Create your * Problem Solving * Algorithms! TODAY! w/ Machine Learning & Data Structures (Artificial Intelligence Series)
Javascript Artificial Intelligence: Made Easy, w/ Essential Programming; Create your * Problem Solving * Algorithms! TODAY! w/ Machine Learning & Data Structures (Artificial Intelligence Series)
The Elements of Artificial Intelligence Using Common LISP
Artificial Intelligence with Common Lisp: Fundamentals of Symbolic and Numeric Processing
Common Lisp and Artificial Intelligence
Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence (Studies in Computational Intelligence)
Swift Programming
Artificial Intelligence: Made Easy, w/ Essential Programming
Learn to Create your * Problem Solving * Algorithms! TODAY! w/ Machine ... engineering, r programming, iOS development)
On Lisp: Advanced Techniques for Common Lisp

Successful Lisp: How to Understand and Use Common Lisp LISP, Lore, and Logic: An Algebraic View of LISP Programming, Foundations, and Applications Java Artificial Intelligence: Made Easy, w/ Java Programming; Learn to Create your * Problem Solving * Algorithms! TODAY! w/ Machine Learning & Data ... engineering, r programming, iOS development) Javascript Artificial Intelligence: Made Easy, w/ Essential Programming; Create your * Problem Solving * Algorithms! TODAY! w/ Machine Learning & Data ... engineering, r programming, iOS development) Artificial Intelligence: Made Easy w/ Ruby Programming; Learn to Create your * Problem Solving * Algorithms! TODAY! w/ Machine Learning & Data ... engineering, r programming, iOS development) Social Intelligence: A Practical Guide to Social Intelligence: Communication Skills - Social Skills - Communication Theory - Emotional Intelligence - Practical Common Lisp (Expert's Voice in Programming Languages) Programming in Common Lisp Object-Oriented Programming in COMMON LISP: A Programmer's Guide to CLOS Prolog Programming for Artificial Intelligence Prolog Programming for Students: With Expert Systems and Artificial Intelligence Topics

[Dmca](#)