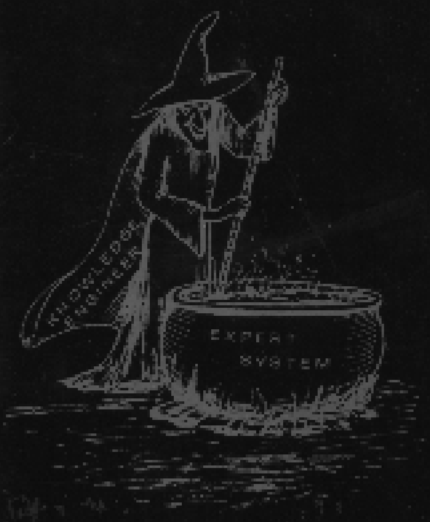
THE ELEMENTS
of
Artificial Intelligence

A atom									E Expertise
Sx S-expression	R Production rule	I ISA hierarchy	K State space	Pc Predicate calculus	Pr Probability	C Concept formation	L Lexicon	Ir Image representation	Sh Shell
Cs CONS	Dn Discrimination network	F Frame	D Depth-first search	Rn Resolution	B Bayes' rule	V Version space	Sy Syntax	Sg Segmentation	Ie Inference engine
Cd COND	Pa Pattern matching	Sn Semantic net	As A* algorithm	U Unification	Fz Fuzzy logic	H Heuristic	S Semantics	Q Relaxation	N Numerical model
Df DEFUN	X Formula manipulation	Cn Constraint	Pl Planning	Ht Herbrand's theorem	Sf Sufficiency factor	Cr Circular reaction paradigm	Pg Pragmatics	M Morphology	In Integration
Mp MAPCAR	Fc Forward chaining	Rd Relational database	Ab Alpha-beta search	P Prolog	Ds Dempster-Shafer calculus	Di Discovery	At Augmented transition net	Z Depth	Pp Parallel processing



An Introduction Using LISP

Steven L. Tanimoto

Computer Science Press

THE ELEMENTS OF ARTIFICIAL INTELLIGENCE

An Introduction Using LISP

PRINCIPLES OF COMPUTER SCIENCE SERIES

ISSN 0888-2096

Series Editors

Alfred V. Aho, *Bell Telephone Laboratories, Murray Hill, New Jersey*

Jeffrey D. Ullman, *Stanford University, Stanford, California*

1. *Algorithms for Graphics and Image Processing**
   Theo Pavlidis
2. *Algorithmic Studies in Mass Storage Systems**
   C. K. Wong
3. *Theory of Relational Databases**
   Jeffrey D. Ullman
4. *Computational Aspects of VLSI**
   Jeffrey D. Ullman
5. *Advanced C: Food for the Educated Palate**
   Narain Gehani
6. *C: An Advanced Introduction**
   Narain Gehani
7. *C for Personal Computers: IBM PC, AT&T PC 6300, and compatibles**
   Narain Gehani
8. *Principles of Computer Design**
   Leonard R. Marino
9. *The Theory of Database Concurrency Control**
   Christos Papadimitriou
10. *Computer Organization**
    Michael Andrews
11. *Elements of Artificial Intelligence Using LISP*
    Steven Tanimoto

*These previously published books are in the *Principles of Computer Science Series* but they are not numbered within the volume itself. All future volumes in the *Principles of Computer Science Series* will be numbered.

OTHER BOOKS OF INTEREST

Jewels of Formal Language Theory
Arto Salomaa

Principles of Database Systems
Jeffrey D. Ullman

Fuzzy Sets, Natural Language Computations, and Risk Analysis
Kurt J. Schmucker

LISP: An Interactive Approach
Stuart C. Shapiro

THE ELEMENTS OF ARTIFICIAL INTELLIGENCE

An Introduction Using LISP

Steven L. Tanimoto

Department of Computer Science, FR-35
University of Washington
Seattle, Washington 98195

Computer Science Press

Printed in the United States of America

Computer Science Press, Inc.
1803 Research Boulevard
Rockville, Maryland 20850

3 4 5 6 92 91 90 89 88 87

Library of Congress Cataloging-in-Publication Data

Tanimoto, S. (Steven)
The elements of artificial intelligence.

Includes bibliographies and indexes.
1. Artificial intelligence— Data processing.
2. LISP (Computer program language) I. Title.
Q336.T36 1987 006.3 86-31044
ISBN 0-88175-113-8
ISSN 0888-2096

To my parents

Taffee and Mary-Mae Tanimoto

PREFACE

Today there is a *growing* recognition of computer science as a *laboratory* science. In addition to the mathematical theory that supports techniques in subareas such as artificial intelligence, the student needs to work with actual programs and problems to get a feel for the technology. This book grew out of the perception that hands-on experimentation coordinated with textbook explanations of principles and of actual programs can provide an ideal learning combination for students of artificial intelligence.

The purpose of this book is to provide an up-to-date and didactically coherent introduction to the principles and programming methods of artificial intelligence. It is appropriate for an undergraduate or first-year graduate course. While it is possible for the student to get acquainted with artificial intelligence in a single quarter or semester, a sequence of two to three quarters or semesters is preferable. The author covers most of the material in two academic quarters at the University of Washington. During the first quarter, Chapters 1 through 6 or 7 are tackled, laying a foundation of symbol manipulation, knowledge representation and inference. The second quarter takes on the more advanced topics: learning, natural language understanding, vision and the integration of AI technology into expert systems.

If programming is to be given a heavy emphasis, the material can be spread over more than two quarters; more of the problems may be assigned, and the instructor may wish to spend some time discussing various aspects of the assignments. In the final term of a two- or three-course sequence, a term project by each student, which can grow out of one of the programs provided in the text, can be very successful.

Unlike other AI texts, *The Elements of Artificial Intelligence* integrates the presentation of principles with actual runnable LISP illustrations. I have attempted to implement a large enough fraction of these ideas in fully-presented LISP programs to allow the student to gain enough intuition through experiment to support his/her understanding of all the principles covered.

While the LISP examples encourage an experimental study of the subject, theory is not avoided. The student needs to gain an appreciation for the interplay between theory and practice. Logical reasoning plays a key role in much of AI today, and other formalisms such as various probabilistic reasoning methods are also important. Various mathematical ideas come up in practically all areas of AI, and a study of AI can serve as an invitation to the student to investigate some of these formalisms further.

The prerequisites for a course based on this book are: (a) an intuitive understanding of how a computer works; this is normally the result of programming experience, (b) an exposure to mathematical logic, at least at the level of the propositional calculus, and preferably some experience with the predicate calculus, (c) high-school algebra, and (d) some familiarity with data structures such as strings, trees, arrays and graphs. Some of the techniques and examples in this book may require an understanding of essential aspects of other subjects: an understanding of what it means to take a derivative of a function (something normally taught in freshman calculus) is needed to appreciate the LEIBNIZ program in Chapter 3; some exposure to mathematical logic would facilitate an understanding of Chapter 6; an exposure to elementary concepts of probability is recommended for students embarking on Chapter 7; and Chapter 10 makes occasional use of several kinds of mathematics, including the integral calculus and computational geometry. However, most of the examples do not require more than common knowledge (e.g., the rules of chess) to understand.

The Elements of Artificial Intelligence is designed to be a self-contained text. However, if a separate, deeper treatment of LISP is desired, there are several books on LISP that could be used in a supplementary fashion. One of these is *LISP* by Winston and Horn; another is by D. Touretsky, and a book particularly suited to students using the Franz Lisp implementation was written by R. Wilensky.

The use of programs to illustrate elements of artificial intelligence seems essential if students are to get a practical view of the field. Courses in AI today can more and more easily have access to sufficient computational facilities, and in the opinion of the author, it is inadvisable to neglect the experience of interaction with computers in introducing AI.

At the same time, a course on artificial intelligence should be an enjoyable one. A primary source of students' pleasure is the chance to write, play with, and modify programs that seem to be clever, and to understand what makes them work or not work. To this end, many of the exercises in the book consist of experimentation with or modification of the programs presented in the text, or explaining aspects of their behavior.

Various implementations of LISP may be used to run the examples, including several excellent microcomputer LISP's. One implementation has been developed by the author specifically to support the examples used in this text; it is the intention of the author and publisher to make this software available at a cost much less than what commercial systems typically cost.

The chapters are intended to be treated in the order given. However, the instructor may choose to omit or supplement material to his or her own taste, as artificial intelligence is a subject of broad scope.

Chapter 1 provides a general introduction addressing the popular question of what intelligence is and the question of how AI is related to other fields. Chapter 2 is a brief but self-contained introduction to interactive programming with the LISP language. This chapter can be skipped by students already familiar with the language. Programming tools and methodology are further developed in Chapter 3. There, a pattern-matching function, MATCH, is described that facilitates several subsequent programs. The chapter illustrates the application of LISP to simple AI problems: carrying on a dialog, and manipulating mathematical formulas according to rules of the differential calculus. The emphasis is on programming techniques.

In Chapter 4 (Knowledge Representation), we begin to explore possibilities for structuring simple factual knowledge to support subsequent inference, using concrete LISP data structures. The example program LINNEUS, described at length, builds upon the MATCH function of the previous chapter to illustrate both the representation of knowledge in an ISA hierarchy, and elementary inference based on that knowledge. The program includes a simple conversational interface. Several issues are raised here which are discussed further in subsequent chapters: search, theorem proving and natural language understanding.

The notion of search, introduced briefly in the previous chapter, is elaborated in Chapter 5 with concepts of state space, evaluation functions, etc. The importance of pruning to fight the combinatorial explosion is explained. Alternative algorithms for searching are presented and compared. Planning is presented as direct application for search algorithms. The chapter closes with a discussion of minimax search and its application in programs to play games such as checkers and chess.

The subject of Chapter 6 is reasoning with the propositional and predicate logics. This is taken to include the more general issue of mathematical logic as a means for representation and inference in AI. To show how search applies to deduction, automatic techniques are presented based on both the propositional calculus and the predicate calculus. The "Logic-Theory Machine" is presented to show a more "human" way to find proofs: to search using subgoals. Presenting unification, we elaborate on the notion of pattern matching (from Chapter 3) and introduce the PROLOG language. A "mock-PROLOG" interpreter written in LISP is presented, and several of the chapter's exercises require the student to use it or modify it. The subject of non-monotonic reasoning wraps up the chapter.

Chapter 7, in contrast to 6, deals with knowledge in which probabilities or certainty values play a crucial role. Bayes' rule is presented, as are some of the epistemological considerations for applying it. We illustrate probabilistic inference networks in the style of PROSPECTOR, and give some guidelines for constructing them. A complete example program is presented which computes probabilities for various hypotheses about the quality of a restaurant, given the values of some observable variables. Finally, the Dempster-Shafer calculus is described.

In Chapter 8 (Learning) we change our perspective. In preceding chapters the concern was with using general knowledge to prove specific theorems, diagnose particular symptoms and solve particular puzzles and problems. Not treated was the question of where the general knowledge comes from. Here the problem of going from specific facts to general knowledge is treated. Starting with empirical data, one can derive hypotheses, rules of inference and classification rules using automatic means. A logical approach to single-concept learning is described, and this leads into a presentation the version-space method. Automatic theory formation is described, and a program PYTHAGORUS is presented which explores a space of concepts about geometry using a heuristic search algorithm.

Chapter 9 addresses the subject of natural-language understanding. Beginning with design criteria for language understanding systems, the notions of syntax, semantics and pragmatics are discussed. Augmented transition networks and semantic grammars are presented as two powerful techniques for building useful systems. An interactive program "Stone World" that allows the user to communicate with a simulated character to achieve action through a subset of natural English demonstrates the power of these methods as well as their limitations.

Machine vision is the subject of Chapter 10. The chapter covers the underlying image representation problems as well as high-level vision techniques. The complexities of interpreting scenes in the midst of ambiguities and incomplete information require that vision call upon many other areas of artificial intelligence to help solve its problems. Computer-vision research has pursued two related but fundamentally different approaches. One of these is the development of algorithmic or architectural models to explain how human vision works; this approach has been labelled "computational vision" by some of its proponents. The other approach is the inventing of techniques for performing useful tasks; this approach includes image processing and robotic vision. While this chapter presents ideas from both approaches, the emphasis is distinctly on the machine, rather than the human, side of vision. This is consistent with the theme of the book that artificial intelligence is in large part a design and programming activity. Two LISP programs are included in Chapter 10, one for connected-components analysis of binary images, and another for polygonal approximation of two-dimensional shapes.

While Chapters 2 through 10 present "elements" of artificial intelligence, Chapter 11 (Expert Systems) discusses the problem of combining the elements into useful compounds. This chapter touches upon such issues as tools and shells for building expert systems, special hardware, and limitations of expert systems.

A closing chapter suggests directions in which artificial intelligence may move in the future, and it mentions some of the technical and social challenges that artificial intelligence raises or may help solve.

S. L. T.

Seattle, Washington

Acknowledgements

I received kind assistance in various aspects of preparing this text from a number of people, especially my students, some of whom had to suffer through very rough early drafts. I am grateful to them for their patience. Particularly helpful were Agnes Andreassian, Wayne Backman, John Ballard, Ron Blanford, Philip Harrison, Jon Holman, Shu-Yuen Hwang, Brian Koblenz, Kuo-Liang Ku, Debora Lilly, Clifford Neuman, Douglas Pan, Joseph Pfeiffer, Harry Plate, Scott Rose, Robert Shields, and Ricky Yeung.

Some of my friends and professional colleagues who made helpful suggestions or provided encouragement are Jean-Loup Baer, Gerard Barber, Alan Borning, Y. T. Chien, Norbert Cot, Per-Erik Danielsson, Gosta Granlund, Alistair Holden, Ramesh Jain, Simon Kasif, Robert Matthews, Ryszard Michalski, Michael Morgan, Charles Osborne, John Palmer, Theo Pavlidis, Ira Pohl, Richard Rice, Alan Shaw, Robert Simmons, Jean-Claude Simon, George Stockman, Imants Svalbe, Leonard Uhr, and Walter Wolnik. I must apologize in advance for failing to incorporate all of the good suggestions I have received. My hope is that the book will survive to a second edition, at which point the additional improvements can be made.

I am indebted to Jeffrey Ullman for numerous constructive comments, and to Art and Barbara Friedman, and Elizabeth Mergner of Computer Science Press for their encouragement during the final stages. For help with the cover design, I would like to thank Jessie Phillips Pearson. Pierre MacKay's TEX-pertise and patient instruction on the use of an aging Alphatype phototypesetter were invaluable. Finally I would like to thank my wife Carmel and daughter Elise for their patience and love that made it possible for me to undertake and complete this project.

Contents