

THE ELEMENTS

of

Artificial Intelligence

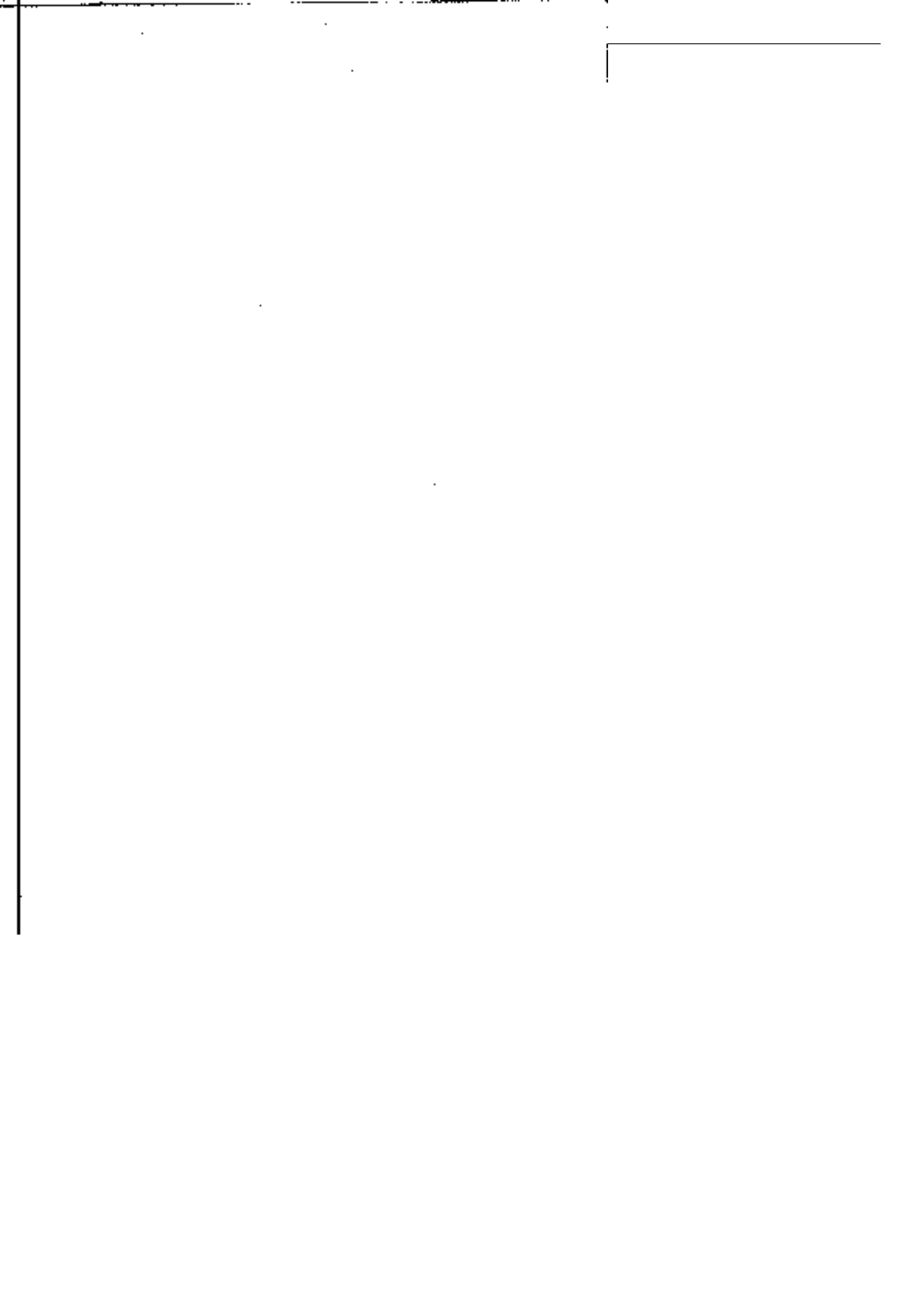
A atoms									E Expertise
Sx S-expression	R Production rules	I ISA hierarchy	K State space	Pc Predicate calculus	Pr Probability	C Concept formation	L Lexicon	Ir Image representation	Sh Shell
Cs CONS	Dn Distributed network	F Frames	D Depth-first search	Rn Resolution	B Bayes' rule	V Version space	Sy Syntax	Sg Segmentation	Ie Interference
Cd COND	Pa Pattern matching	Sn Semantic net	As A* algorithm	U Unification	Fz Fuzzy logic	H Heuristic	S Semantics	Q Relaxation	N Neural model
Df DRIFL	X Formal manipulation	Cn Constraint	Pl Planning	Ht Heuristic theorem	Sf Sufficiency factor	Cr Circular overlap paradigm	Pg Pragmatics	M Microscopy	In Inception
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 Zeta	Pp Faceted 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 0883-2096

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 Structures 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**
Nathan Gehani
6. *C: An Advanced Introduction**
Nathan Gehani
7. *C for Personal Computers: IBM PC, AT&T PC family, and compatibles**
Nathan 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 Eusimoto

* 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, FC-35
University of Washington
Seattle, Washington 98195*

Computer Science Press

Copyright © 1987 Computer Science Press, Inc.

Printed in the United States of America

All rights reserved. No part of this book may be reproduced in any form including photostat, microfilm, xerography, and not in information storage and retrieval systems, without permission in writing from the publisher, except by a reviewer who may quote brief passages in a review or as provided in the Copyright Act of 1976.

Computer Science Press, Inc.
1803 Research Boulevard
Rockville, Maryland 20856

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.

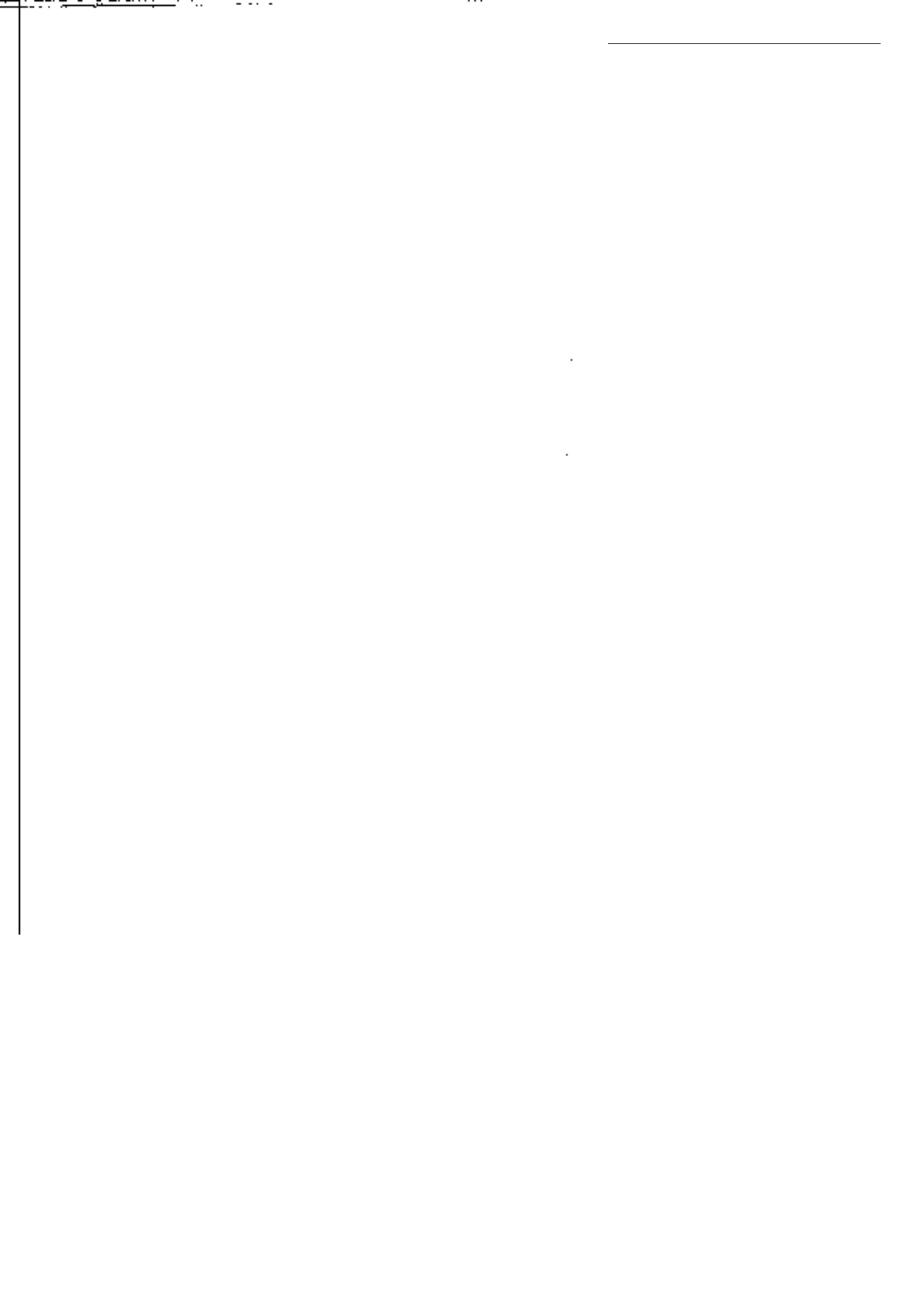
1. Artificial intelligence—Data processing.
2. LISP (Computer program language) I. Title.

Q336.L36 1987 006.3 86-31044

ISBN 0-88175-113-8

ISSN 0885-2095

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-processed 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

inferred 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 of the version-space method. Automatic theory formation is described, and a program PYTHAGORIS 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. F.

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 Androssian, Wayne Backman, John Ballard, Ron Bradford, Philip Harrison, Tom Holman, Shu-Yuen Hwang, Brian Kozlekiz, Kuo-Lung Ku, Debra Lilly, Clifford Kenner, Douglas Pan, Joseph Pfeuffer, Harry Plate, Scott Rose, Robert Shields, and Rocky Young.

Some of my friends and professional colleagues who made helpful suggestions or provided encouragement are Jean-Loup Baer, Gerard Barber, Alan Borning, Y. T. Chen, Nohbert Cox, Per-Erik Danielsson, Gösta Granlund, Alisgar Holden, Ramaiah Jain, Simon Kasif, Robert Matthews, Krzysztof Michalski, Yohji Moriguni, Charles Osborne, John Palmer, Theo Pavlidis, Ira Pohl, Richard Rice, Alan Shaw, Robert Simmons, Jean-Claude Simon, George Stockman, Ingvars Svathé, 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 arrive in a second edition, at which point the additional improvements can be made.

I am indebted to Jeffrey Ullman for numerous constructive comments, and to Ann 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 T_EX-pertise and patient instruction on the use of an aging Alphatype phototypesetter were invaluable. Finally I would like to thank my wife Gannel and daughter Elise for their patience and love that made it possible for me to undertake and complete this project.

Contents

1	Introduction	1
1.1	An Intellectual and Technical Challenge	1
1.1.1	They Said It Couldn't Be Done	1
1.1.2	Artificial Intelligence In Action	2
1.2	What Intelligence Is	4
1.2.1	Aspects of Human Intelligence	4
1.2.2	Communication	4
1.2.3	Learning	5
1.3	What Artificial Intelligence Is	6
1.3.1	A Field of Study	6
1.3.2	AI: Art or Science?	6
1.3.3	A Purpose	7
1.4	Artificial Intelligence Comes of Age	7
1.4.1	Growth of the AI Research Community	7
1.4.2	The Industrialization of AI	8
1.4.3	What An AI Practitioner Does	8
1.5	Philosophical Challenges	8
1.5.1	Turning's Test	9
1.5.2	AI and Human Beliefs	10
1.6	The Reference Literature	10
	References	12
2	Programming in LISP	15
2.1	Introduction	15
2.2	S-Expressions	16
2.2.1	Machine Representation of S-expressions	17
2.2.2	Correspondence Between Lists and Dotted Pairs	18
2.3	Functional Forms	19
2.3.1	Simple Forms and Their Evaluation	19
2.3.2	Nested Forms	20
2.4	CONS, CAR and CDR	20
2.5	QUOTE and SETQ	23

2.5.1	QUOTE	23
2.5.2	SETQ	24
2.6	Special Forms	24
2.7	Predicates	25
2.8	COND	25
2.9	Defining Ordinary Functions	26
2.9.1	DEFUN	27
2.9.2	Recursive Definitions of Functions	27
2.9.3	Appending Lists Recursively	31
2.9.4	Commenting LISP Programs	31
2.10	PROC	31
2.11	EVAL and APPLY	33
2.11.1	EVAL	33
2.11.2	APPLY	34
2.11.3	LAMBDA Expressions	34
2.11.4	Closures of Functions	35
2.12	Defining Special Forms	36
2.13	MAPCAR	37
2.14	READ and PRINT	38
2.15	The Property List of an Atom	39
2.16	Replacing Pointers in LISP Memory	40
2.17	Debugging	41
2.18	Programming Methodology	43
2.18.1	Benefits of Programming in LISP	43
2.18.2	Stepwise Refinement	44
2.18.3	Ways to Keep Complexity Manageable	44
2.18.4	Pragmatics	45
2.19	Implementation of LISP	45
2.19.1	Representations of Literal Atoms, Numbers and CONS Cells	45
2.19.2	Garbage Collection	46
2.20	Bibliographical Information	47
	References	47
	Exercises	48
3	Productions and Matching	53
3.1	Overview	53
3.2	Production System Methodology	53
3.2.1	Modularity Revisited	53
3.2.2	Ordered Production Systems	56
3.2.3	Limitations of Linear Production Systems	57
3.2.4	Discrimination Nets	57
3.3	Pattern Matching	56
3.3.1	Pattern Matching in Production Rules	56
3.3.2	The MATCH Function	60

3.4	The "SHRINK"	65
3.4.1	Implementation of SHRINK	66
3.4.2	A Session with the SHRINK	69
3.5	Mathematics' Formula Manipulation	70
3.5.1	Motivation	70
3.5.2	Overall Structure of LEIBNIZ	71
3.5.3	Production Rules for LEIBNIZ	71
3.5.4	Control Scheme for LEIBNIZ	78
3.5.5	Database of State Information in LEIBNIZ	80
3.5.6	Performance of LEIBNIZ	80
3.6	The Notion of Unification	81
3.7	Bibliographical Information	82
	References	82
	Exercises	83
4	Knowledge Representation	89
4.1	Characteristics of Knowledge	89
4.2	Overview of Knowledge Representation Methods	89
4.3	Knowledge in Production Rules	91
4.4	Concept Hierarchies	91
4.4.1	Inclusion, Membership and "ISA"	92
4.4.2	Partial Orders and Their Representation	93
4.5	An ISA Hierarchy in LISP	95
4.5.1	Preliminary Remarks	95
4.5.2	The Use of Property Lists to Store Relations	95
4.5.3	Searching a Base of Facts	97
4.6	A Conversational Front End	98
4.7	Inheritance	103
4.7.1	Inheritance from Supersets	103
4.7.2	HAS Links	104
4.7.3	Multiple Inheritance	105
4.7.4	Default Inheritance	106
4.8	Propositional and Predicate Logic	107
4.8.1	Remarks	107
4.8.2	Propositional Calculus	108
4.8.3	Predicate Calculus	110
4.9	Frames of Context	112
4.9.1	A "Kitchen" Frame	112
4.9.2	Slots and Fillers	113
4.9.3	Schemata	113
4.9.4	Attachments to Slots	114
4.10	Semantic Networks	115
4.10.1	Motivation	115
4.10.2	Representing Sentence Semantics	116

4.10.3	Representing Non-Binary Relations	119
4.10.4	Semantic Primitives and Combining Forms	119
4.11	Constraints	120
4.11.1	Constraint Schemata	120
4.11.2	Using Constraints	122
4.11.3	Satisfying Constraints	124
4.12	Relational Databases	124
4.12.1	Remarks	124
4.12.2	n -ary Relations	124
4.12.3	Selection	125
4.12.4	Projection	126
4.12.5	Joins	126
4.13	Problems of Knowledge Representation	127
4.13.1	The Closed-World Assumption	128
4.13.2	Knowledge Acquisition	129
4.14	Summary of Knowledge Representation Schemes	129
4.15	Bibliographical Information	130
	References	131
	Exercises	132
5	Search	139
5.1	The Notion of Searching in a Space of States	139
5.2	The Painted Squares Puzzle	141
5.2.1	A Kind of Geometrical Puzzle	141
5.2.2	Solution Procedure	142
5.2.3	States and Operators	142
5.2.4	Representation of the Puzzle and Its States in LISP	144
5.2.5	A Backtracking Search Procedure in LISP	146
5.2.6	Remarks on Efficiency	148
5.3	Elementary Search Techniques	148
5.3.1	A Search Problem	148
5.3.2	Hypothesize and Test	150
5.3.3	Depth-First Search	150
5.3.4	Breadth-First Search	152
5.4	Heuristic Search Methods	153
5.4.1	Evaluation Functions	153
5.4.2	Best-First (Ordered) Search	154
5.4.3	Searching Graphs with Real Distances	157
5.4.4	The A* Algorithm	160
5.5	Planning	164
5.5.1	Problem Solving Before Action	164
5.5.2	A Robotics Example	165
5.5.3	Hierarchical Planning	167
5.6	Two-Person, Zero-Sum Games	168

5.6.1	Minimizing	169
5.6.2	AND/OR Graphs	170
5.6.3	Alpha Beta Search	172
5.7	Bibliographical Information	176
	References	177
	Exercises	178
6 Logical Reasoning		187
6.1	Motivation	187
6.2	Proofs in the Propositional Calculus	187
6.2.1	Perfect Induction	187
6.2.2	Wang's Algorithm	189
6.2.3	Wang's Algorithm in LISP: "PROVER"	190
6.2.4	The "Logic Theory Machine"	193
6.2.5	The Resolution Principle in the Propositional Calculus	197
6.3	Predicate Calculus Resolution	198
6.3.1	Preliminary Remarks	198
6.3.2	An Example	198
6.3.3	Putting a Formula into Clause Form	200
6.3.4	Unification	202
6.3.5	A Unification Algorithm	202
6.3.6	A Unifier in LISP	204
6.3.7	Factors and Resolvents	207
6.4	The Logical Completeness of Resolution	207
6.4.1	Semantic Trees	208
6.4.2	The Herbrand Universe and Herbrand Base	208
6.4.3	Herbrand's Theorem	210
6.4.4	The Completeness of Resolution	211
6.5	Resolution Strategies	213
6.5.1	Set of Support	214
6.5.2	Linear Format	214
6.5.3	Unit Preference	215
6.6	Solving Problems With Resolution	215
6.7	Logic Programming and PROLOG	217
6.7.1	Intercalation	217
6.7.2	Horn Clauses	217
6.7.3	A Simple Logic Program	218
6.7.4	Another PROLOG Example	219
6.7.5	A Mock PROLOG Interpreter in LISP	220
6.7.6	PROLOG's List-Handling Facilities	225
6.7.7	Cut and Other PROLOG Features	226
6.7.8	LISP versus PROLOG	228
6.8	Non Monotonic Reasoning	228
6.8.1	Motivation	228

6.3.2	Circumscription	229
6.9	Bibliographical Information	230
	References	231
	Exercises	233
7	Probabilistic Reasoning	239
7.1	Introduction	239
7.1.1	The Need to Represent Uncertain Information	239
7.1.2	The Nature of Probabilistic Information	239
7.2	Probability	240
7.2.1	The Notion of Certainty	240
7.2.2	Axioms of Probability	241
7.2.3	Bayes' Rule	242
7.3	Probabilistic Inference Networks	244
7.3.1	Appropriate Domains	245
7.3.2	Heuristical Components of Inference Networks	246
7.3.3	Fuzzy Inference Rules	246
7.3.4	Steps in the Design of Inference Networks	250
7.4	Updating in Inference Networks	252
7.4.1	Odds and Bayes' Rule	253
7.4.2	Handling Uncertain Evidence	255
7.4.3	The Bayesian Dilemma for Inference Networks	257
7.4.4	Updating the Probabilities	258
7.4.5	Combining Independent Evidence at a Node	261
7.5	An Inference Network in LISP	262
7.5.1	The Traveller's Restaurant Selection Problem	262
7.5.2	Heuristics for Restaurant Evaluation	263
7.5.3	Implementation	264
7.6	The Dempster-Shafer Calculus	271
7.6.1	Motivation	271
7.6.2	Definitions	272
7.6.3	Dempster's Rule of Combination	273
7.6.4	Simple Evidence Functions	275
7.7	Bibliographical Information	276
	References	277
	Exercises	278
8	Learning	283
8.1	Introduction	283
8.1.1	Overview	283
8.1.2	What Learning Is	284
8.1.3	Inductive vs. Deductive Methods	285
8.1.4	Knowledge Acquisition	286
8.2	Learning Classification Rules	287

5.2.1	General Rules from Fixed Examples	386
5.2.2	Incremental Learning	391
5.2.3	Version Spaces	393
5.3	Self-Directed Conceptualization Systems	216
5.3.1	Piaget's Circular Reaction Paradigm	206
5.3.2	Automatic Theory Formation	207
5.4	PYTHAGORUS	208
5.4.1	General Remarks	208
5.4.2	Task Creation	298
5.4.3	Implementation	299
5.4.4	Behavior and Limitations of PYTHAGORUS	311
5.4.5	Discussion	313
5.5	Concluding Remarks	315
5.6	Bibliographical Information	315
	References	316
	Exercises	318
9	Natural-Language Understanding	328
9.1	Introduction	333
9.1.1	Intelligence in Understanding Language	323
9.1.2	Applications of Language Understanding	334
9.1.3	Machine Translation	324
9.1.4	A Definition of Understanding	325
9.1.5	Representing Events with Case Frames	326
9.1.6	Components of an Understanding System	327
9.2	Syntax	328
9.2.1	Formal Languages	329
9.2.2	Context-Free Grammars and Languages	331
9.2.3	A Backtracking Parser in LISP	332
9.2.4	Grammar for English	336
9.3	Semantics and Representation	337
9.3.1	Lexicons	337
9.3.2	Representing Events in Frames	338
9.3.3	Scripts: Schemata for Chains of Events	340
9.3.4	Semantic Primitives	342
9.3.5	Logical Forms	343
9.3.6	The Semantics of Quantity	344
9.3.7	The Semantics of Time	345
9.3.8	The Semantics of Space	346
9.3.9	The Semantics of Knowing and Belief	347
9.4	Computing Interpretations	347
9.4.1	Semantic Grammars	348
9.4.2	Augmented Transition Networks	349
9.4.3	Detecting and Resolving References	350

9.5	Dialog Management	352
9.5.1	Modes of Conversation	352
9.5.2	The Phases of a Conversation	353
9.5.3	Monitoring Communication Effectiveness	354
9.5.4	Schemata for Controlling Conversation	351
9.6	An English Interface to a Microworld	355
9.6.1	System Description	355
9.6.2	Network Description	357
9.6.3	Implementation	356
9.6.4	Sample Session	371
9.7	Bibliographical Information	372
	References	373
	Exercises	376
10	Vision	379
10.1	Introduction	379
10.1.1	The Richest Sense	379
10.1.2	A Key Area of AI	379
10.1.3	The Challenge of Computer Vision	380
10.1.4	Overview	380
10.1.5	The Physiology of Human Vision	381
10.1.6	Visual Illusions	384
10.2	Image Formation and Acquisition	386
10.2.1	The Variety of Image Sources	386
10.2.2	Scenes of Three Dimensional Surfaces	387
10.2.3	Image Acquisition	388
10.2.4	The Vidicon	389
10.2.5	Solid-State Image Devices	390
10.2.6	Range-Finders	391
10.2.7	Sampling	392
10.2.8	Aliasing	392
10.2.9	Non-Rectangular Pixels	394
10.2.10	Quantization	394
10.2.11	Pyramids	394
10.3	Preprocessing	395
10.3.1	Low-Pass Filtering	395
10.3.2	The Two-Dimensional Discrete Fourier Transform	395
10.3.3	Moving Average and Median Filtering	397
10.3.4	Histograms and Thresholding	397
10.4	Connectedness and Cellular Logic	398
10.4.1	Pixel Adjacency	398
10.4.2	Connected Sets of Pixels	399
10.4.3	An Algorithm	399
10.4.4	A Connected Components Program in LISP	400

10.4.5	Cellular Logic and Parallel Operations	404
10.4.6	Erosion and Dilation	406
10.5	Edges and Lines	408
10.5.1	Local Differencing Methods	408
10.5.2	Compass Gradients	409
10.5.3	A Vector-Space Approach	409
10.5.4	Heuristic Search for Object Boundaries	409
10.5.5	Gaussians, Laplacians and Zero Crossings	411
10.5.6	Edge Linking	415
10.5.7	Relaxation Labeling	416
10.5.8	The Hough Transform	417
10.6	Region Growing	419
10.6.1	Pixel-by-Pixel Region Growing	421
10.6.2	Split-and-Merge Algorithm	422
10.6.3	The Phagocyte Heuristic	422
10.6.4	Relational Descriptions	424
10.7	Shape Analysis	424
10.7.1	Circumscribing and Inscribing	426
10.7.2	Integral Projections	428
10.7.3	Topology	428
10.7.4	The Medial-Axis Transform	429
10.7.5	Digital Skeletons	431
10.7.6	Polygonal Approximation	431
10.7.7	Ramer's Recursive Algorithm	432
10.7.8	Ramer's Algorithm in LISP	433
10.8	Three Dimensions and Motion	437
10.8.1	Representations for 3-D Objects	438
10.8.2	3-D Arrays	438
10.8.3	Octrees	438
10.8.4	Generalized Cylinders	439
10.8.5	3-D Structure from 2-D Images	439
10.8.6	Shape from Shading	440
10.8.7	Stereo	440
10.8.8	Motion	442
10.9	Block-World Heuristics	443
10.9.1	Guzman Labeling for Line Drawings	444
10.9.2	Huffman-Crews Labeling	445
10.10	Bibliographical Information	446
	References	449
	Exercises	474

11 Expert Systems	461
11.1 Introduction	461
11.2 Integration of AI Techniques	462
11.2.1 Design Methodology	462
11.2.2 Ways to Combine AI Techniques	463
11.2.3 Incorporating Numerical Models	464
11.3 Tools	465
11.3.1 Shells for Rule-Based Systems	465
11.3.2 Knowledge-Base Construction	465
11.3.3 Graphical Representations	465
11.4 Hardware	466
11.4.1 LISP Machines	467
11.4.2 PROLOG Machines	467
11.4.3 Parallel Architectures	468
11.5 Examples of Expert Systems	469
11.5.1 XCON: A Configurer for VAX Computers	469
11.5.2 CADUCEUS: An Internal Medicine Consultant	470
11.5.3 TAXMAN: A Corporate Tax Law Advisor	470
11.6 Limits of Expert Systems	470
11.6.1 Quality of Knowledge	471
11.6.2 Speed of Inference	471
11.6.3 Proper and Improper Use	471
11.7 Bibliographical Information	472
References	472
Exercises	473
12 The Future	475
12.1 Where is AI Going?	475
12.1.1 Deeper Integration	475
12.1.2 More Mature Application Methodologies	476
12.2 Economic Benefits	476
12.2.1 Advantages of Expert Systems	477
12.2.2 AI Pushing Computer Technology	477
12.3 Social Growing Pains	477
12.3.1 Labor	478
12.3.2 The Possibility of Greater Confusion	478
12.3.3 User Responsibility and Alienation	478
12.3.4 A Human Identity Crisis	479
12.3.5 The Sorcerer's Apprentice	480
12.4 Cultural Benefits	480
12.4.1 Refinement of Human Knowledge	480
12.4.2 Improvements to Human Language	481
12.4.3 New Ways of Seeing	481
12.4.4 Revitalization of Intellectual Pursuits	482

12.5 Bibliographical Information	489
References	489
Appendix A: Summary of LISP Functions	485
Author Index	493
Subject Index	497
Software Ordering Information	530

sample content of Elements of Artificial Intelligence: Introduction Using LISP (Principles of computer science series)

- [click Renaissance Truths: Humanism, Scholasticism and the Search for the Perfect Language](#)
- [The Big Book of One Pot Recipes: More Than 500 One Pot Recipes for Easy, Flavorful Meals for free](#)
- [read online O Homem como Invenç o de Si Mesmo: Mon logo em Um Ato](#)
- [read online The School Bus Doesn't Stop Here Anymore: Life and Times on a Rural Route](#)
- [read online Off-Screen Cinema: Isidore Isou and the Lettist Avant-Garde](#)
- [read online St phane Mallarm  \(Reaktion Books - Critical Lives\)](#)

- <http://wind-in-herleshausen.de/?freebooks/Renaissance-Truths--Humanism--Scholasticism-and-the-Search-for-the-Perfect-Language.pdf>
- <http://flog.co.id/library/The-Big-Book-of-One-Pot-Recipes--More-Than-500-One-Pot-Recipes-for-Easy--Flavorful-Meals.pdf>
- <http://bestarthritiscare.com/library/The-Long-Road-Turns-to-Joy--A-Guide-to-Walking-Meditation.pdf>
- <http://betsy.wesleychapelcomputerrepair.com/library/150-Best-Minimalist-House-Ideas.pdf>
- <http://growingsomeroots.com/ebooks/Off-Screen-Cinema--Isidore-Isou-and-the-Lettist-Avant-Garde.pdf>
- <http://patrickvincitore.com/?ebooks/St--phane-Mallarm---Reaktion-Books---Critical-Lives-.pdf>